Proceedings of the Seminar in Computer Science (CS-E4000), Fall 2020

Antti Ylä-Jääski and Verónica Toro-Betancur

Tutors for seminar topics

Alex Jung, Amel Bourdoucen, Anirudh Jain, Anton Debner, Aqdas Malik, Blerta Lindqvist, Esa Vikberg, Hiroshi Doyu, Matti Siekkinen, Sanna Suoranta, Thaha Mohammed, Thanh-Phuong Pham, Thomas Nyman, Vesa Hirvisalo and Wencan Mao

Preface

The *Seminar on Network Security*, *Seminar on Internetworking* and *Seminar on Software Technology and Systems Research* were previously separate Master's level courses in computer science at Aalto University. These seminar courses have now merged into one seminar course. These seminar series have been running continuously since 1995. From the beginning, the principle has been that the students take one semester to perform individual research on an advanced technical or scientific topic, write an article on it, and present it on the seminar day at the end of the semester. The articles are printed as a technical report. The topics are provided by researchers, doctoral students, and experienced IT professionals, usually alumni of the university. The tutors take the main responsibility of guiding each student individually through the research and writing process.

The seminar course gives the students an opportunity to learn deeply about one specific topic. Most of the articles are overviews of the latest research or technology. The students can make their own contributions in the form of a synthesis, analysis, experiments, implementation, or even novel research results. The course gives the participants personal contacts in the research groups at the university. Another goal is that the students will form a habit of looking up the latest literature in any area of technology that they may be working on. Every year, some of the seminar articles lead to Master's thesis projects or joint research publications with the tutors.

Starting from the Fall 2015 semester, we have merged the three courses into one seminar that runs on both semesters. Therefore, the theme of the seminar is broader than before. All the articles address timely issues in security and privacy, networking technologies and software technology.

These seminar courses have been a key part of the Master's studies in several computer-science major subjects at Aalto, and a formative experience for many students. We will try to do our best for this to continue. Above all, we hope that you enjoy this semester's seminar and find the proceedings interesting.

Seminar papers

Attacks in adversarial machine learning

Alexander Colb
alexander.colb@aalto.fi

Tutor: Blerta Lindqvist

Abstract

Deep Neural Networks are increasingly incorporated into applications in our daily lives, from virtual assistants to automatic fraud detection. With widespread adoption comes an increased threat to human life and other high-value assets. Deep learning designers are now expected to understand the inherent shortcomings of deep neural nets, as a prerequisite to comprehending what attacks they are up against. This paper aims to serve as a springboard for a basic understanding of security within the context of deep learning.

We venture into some of the currently known attacks that rely on slight perturbation of a legitimate sample with the goal of eliciting a misclassification from the deep learning model. More specifically, we explain both white-box attacks and black-box attacks.

KEYWORDS: adversarial, machine learning, attacks

1 Introduction

The current state of Machine Learning (ML) has allowed us to automate complex decision-making problems through what is referred to as Deep Learning (DL). As DL methodologies are increasingly incorporated into applications in our daily lives, from virtual assistants to automatic fraud detection, whole new fields of threats begin to introduce themselves. Applications that impact the security of human lives or other high-value systems are especially crucial. For example, self-driving cars rely on ML models to decide how to behave in different situations. Consider an autonomous vehicle that needs to detect a stop sign in order to yield to oncoming traffic. If this decision can somehow be maliciously influenced, the repercussions would prove devastatingly severe. [3]

A lot of work has gone into identifying various attacks and into introducing their respective defences. However, different modes of attack apply in different situations, and there exists no approach that will fit them all. Therefore it is paramount for the field that a general overview be understood by a wide range of DL designers. The aim of this paper is to consolidate the current state of adversarial attacks on machine learning, so as conceptualize some of the risks that designers of DL applications should be aware of. We only introduce countermeasures to those attacks on a superficial level, and leave the consolidation of those measures to further work.

Throughout most of this paper we will use visual recognition to illustrate the various attacks on ML. It should however be noted that these attacks may also be applicable to other DL paradigms such as Natural Language Processing (NLP) or fraud pattern detection.

In the first section, we underline some of the currently known attacks against Deep Neural Networks (DNNs). We go through the most notable white-box attacks, including the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD). Furthermore, we underline some known black-box attacks such as the Zeroth Order Optimization (ZOO) attack, and conclude with a brief discussion of noise-resistant attacks in a 3D context. Finally, we discuss some of the common defensive countermeasures against these known attacks.

$\longmapsto$
classification

label	confidence
zero	0.01
one	0.10
two	**0.20**
three	0.02
four	0.02
five	0.01
six	0.01
seven	**0.60**
eight	0.01
nine	0.02

Figure 1. A handwritten digit is fed to a classifier trained on the MNIST dataset. The model returns a list of probability scores for each known label. In this case, the model correctly guesses the digit to be a seven (the *top-1* class).

2 Attacks

The main attack vector we will inspect in this document targets the way a DNN model classifies a given input. We recall that a DL model accepts an input vector, and gives out a vector of confidence scores for a given set of labels. For example, a model trained on the MNIST database will accept a bitmap of a handwritten digit. The output of that model, respectively, would be a vector of each of the ten digits, each value denoting the probability of the input belonging to that label.

Consider an MNIST classifier that gets a query of a handwritten digit "7". The classifier would then output a list of all digits, from zero to nine, paired with a confidence score as illustrated in Figure 1. A successful classifier will then have a high score for the label "seven", and lower scores for all the other digits. Because of the inherent entropy in handwriting, the other scores are almost always non-zero. Edge cases, such as ones where even a human might be unsure about the label, will produce several high-value scores. For example, an unclearly written "7", might sometimes resemble a "2" instead. But as long as the score for for the label "seven" is the highest of them all, the classifier has done its job correctly.

We consider an attacker who wants to take a legitimate input, and have the model return some illegitimate output. To achieve this, the attacker slightly modifies the legitimate input according to some heuristic. Crucially, that modification needs to be small enough so that the *adversarial sample* still somewhat resembles the legitimate one, eg. to the human

eye. A possible threat scenario against our MNIST model could then be a case where a handwritten cheque is slightly modified to make the recognition software report a larger figure on the cheque. Upon inspection, a human might not be able tell the difference between the two. This unintended phenomenon, where an almost un-noticeable modification results in a misclassification, serves as the main attack scenario throughout this document. Figure 2 illustrates this attack by modification, or *perturbation*, as it is generally referred to as in the literature. [12]

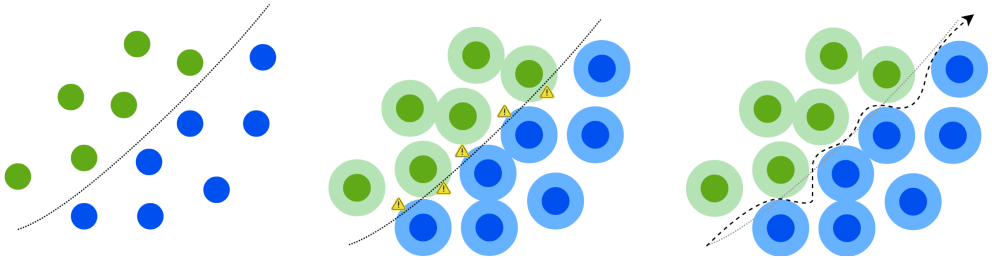


Figure 2. A classifier is able to correctly decide between two labels when presented with natural samples. However, when those samples are adversarially perturbed, the model will misclassify some of those perturbed samples - resulting in a successful attack (depicted as yellow triangles). A more complex decision boundary is required to defend against those perturbations.

2.1 White-box attacks

White-box attacks assume that we have access to the model itself, including its internal layers. [12]

The simplest of adversarial attacks against DNNs target the confidence values for the output of a deep neural network. In our MNIST example, an attacker would want to lower the classifier's confidence of the label "seven", and increase the confidence of one or several other labels. In effect, a successful attacker will have made the classifier a little less certain of its classification. [12]

While this scenario can have real-world impacts, the more interesting attack is one where we change the confidence scores so much, that the highest score (the *top-1* label) in fact shifts to another label. In our MNIST example, the top-1 label would be awarded to any other label but "seven", even though the modified input still resembled a seven. This type of attack is called a source-target misclassification, or a chosen target attack. It can be formalized as follows: [12]

$$arg \min_{\delta X} ||\delta X|| \; such\; that \; F(X + \delta X) = Y^*$$

where δX is the perturbation applied to the original vector X, F is the

DNN model and Y^* the misclassification. In other words, the attacker hopes to minimize the perturbation applied on top of a legitimate input X, while still achieving the desired misclassification Y^*. Since the DNN is inherently non-linear, this becomes complex to optimize [12].

Szegedy et al. [15] demonstrated that adversaries can be consistently crafted using an algorithm called the *Box-constrained Limited-memory Broyden–Fletcher–Goldfarb–Shanno* algorithm. Interestingly, they found that an adversarial sample will often be misclassified on a DNN trained on a different subset of training data or one with a completely different architecture. This suggests that models in use today fail to fully capture the underlying semantics of their labels, eg. what truly makes a "7" a "seven". Instead, these models simply break down when introduced to a sample that exists in the training set with only low probability [5].

Goodfellow et al. [5] go on to introduce the Fast Gradient Sign Method for creating adversarial samples. The perturbation is found using the function:

$$\delta X = \epsilon * sign(\nabla_x J(\theta, X, Y)$$

where ϵ is the amount of perturbation applied, J is the loss function used to train the model, and θ are the model's parameters. The loss function's gradient at the original image is then followed in order to reach a misclassification. For a illustration of this, refer to Figure 3.

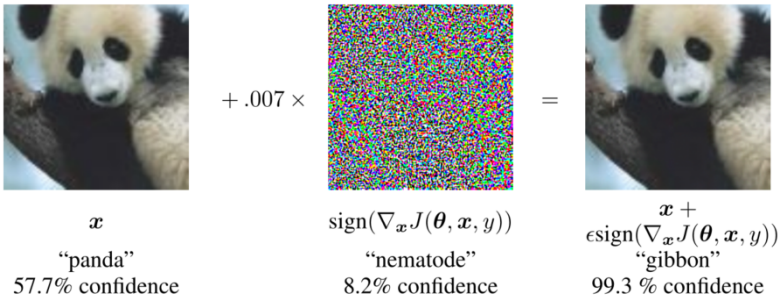


Figure 3. Adversarial noise is generated by following the loss function's gradient at the sample image. It is then applied onto the sample image ("panda") with $\epsilon = 0.007$, resulting in a misclassification ("gibbon"), ie. a successful attack. [5]

Projected Gradient Descent takes this methodology even further. While FGSM relies on a single call to the loss function's gradient, PGD iterates through the nearby topology of the loss function. Additionally, instead of starting at the sample point, the algorithm chooses a random point within the accepted perturbation space. Furthermore, to avoid getting stuck in local maxima, the algorithm does random resets within the aforemen-

tioned space. For a simplified illustration of PGD, refer to Figure 4.

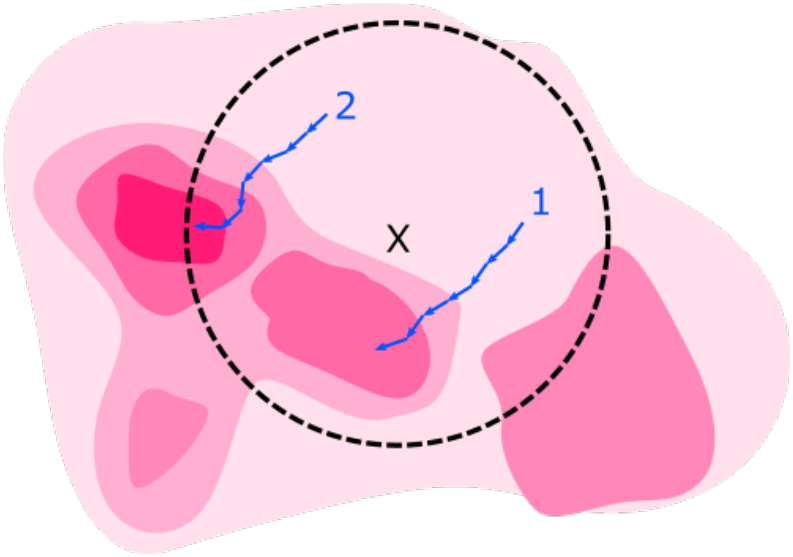


Figure 4. Visualization of the loss function, whose gradient is traversed iteratively around the sample X. In PGD, the traversal begins from a random point (here, *Point 1*) inside the allowed perturbation space (black circle). A new random point is selected (here, *Point 2*) in order to get out of local maxima.

Carlini and Wagner [3] introduce an other method to optimize this attack. They show that for $F(X + \delta X) = Y^*$ to be satisfied, the following is satisfied:

$$f(X + \delta X) \leq 0$$

where f is a function better suited for the optimization problem. They underline different candidates for f, the more successful of which use an internal, un-normalized layer of the model [3]. This method, referred to as the CW attack, seems to currently stand as one of the strongest white-box attacks against DNN classifiers [9].

2.2 Black-box attacks

The reality is that many DNN applications tend to hide their internal structure from an outside observer. For these cases, a white-box attack would not be feasible, since it requires insight into the internal configuration. Leveraging computation on the model's gradients becomes troublesome.

For these cases, we recognize black-box attacks. Black-box attacks are allowed to feed the DNN any input, and see the output vector of confidence scores. However, it has no visibility into the internal structure of the network, ie. the layers and configurations. Instead, the attacker is

assumed to have unrestricted access to an oracle which takes in the input vector, and gives out the classification.

Some attacks use this unrestricted oracle to actually train a new, substitute model. That substitute model acts as a viable approximation of the target network. This effectively provides the attacker with a new neural network, where they are free to apply a white-box attack. [7, 11, 13]

While a substitute model can approximate the original model, some loss will inevitably be introduced. One black-box attack, called the Zeroth Order Optimization attack, negates this loss in transferability. Instead of training a new model, this attack emulates back propagation by using the oracle for the DNN. The way that the ZOO attack is carried out is by selecting a random pixel coordinate, and then emulating a gradient on that coordinate according to some approximation heuristic that calls the oracle. This is repeated until the oracle returns the desired change in the output vector. [4]

Furthermore, there exist black-box attacks whose DNN oracle only returns the top-1 label, instead of the confidence scores for all of the labels. This is called a decision-based, as opposed to a score-based, attack. It addresses real-world cases where the network doesn't expose the confidence scores. One such attack is called a Boundary Attack. In this attack, we straddle the boundary between a known adversarial sample and the original sample, and stochastically optimize for a minimum distance between the two. [2]

2.3 Noise-resistant attacks

In physical applications, the front line for attacks may be subject to natural noise or transformations such as camera distance or angles. In these cases, precise perturbations of the input may not reliably succeed in yielding the intended malicious result, as the random noise may well counteract the adversarial perturbation. [1]

Athalye et al. [1] make their white-box attack take into account a set of transformations of the original image. For 2-dimensional objects such as printouts, they simulate transformations that might take place in a real-world scenario. Recognizing that the camera positioning and object setup may vary, they programmatically apply scaling, translation, rotation, random noise, and a change in lightness.

However, 3-dimensional objects cannot be reliably transformed from a 2-dimensional representation. For this case, they resort to using 3D models

of everyday objects, and render them in different poses. They then apply a texture onto the model, and perturb that texture until it fools the classifier in a large number of different poses. To preserve generality in the real world, they also apply some of the transformations mentioned in the 2D-setup. [1]

Athalye et al. [1] show that a classifier will be fooled by a 3D-printed object that has been covered with the optimized attack texture. They take pictures of the real-world object from a variety of different angles, and find that the attack in fact does generalize well to the real, physical world. They point out that this finding was achieved with a low-cost commercially available 3D printer, for the object body, and laser printer for the texture. With access to higher-fidelity printers, they predict achieving even less noticeable perturbations.

They also find that the magnitude of the required perturbation depends on the amount and range of transformations that need to be catered to. For example, to have a sample remain adversarial over a $360°$ rotation will introduce a more noticeable perturbation than if the required rotation were only, say, $90°$. This also explains why perturbations for 3D samples resemble their respective adversarial labels more than their 2-dimensional counterparts. Figure 5 illustrates this resemblance of the adversary, as the texture has been crafted to work in a variety of camera transformations. Had we only been interested in a 2D adversary, a far less noticeable perturbation would likely have managed to produce the desired, adversarial classification of an espresso cup.

Figure 5. Camera images of a 3D-printed baseball with an adversarial texture applied to make the ball register as a cup of espresso. The classifier has been fooled in the two cases outlined in red. The texture has humanly noticeable coffee-like features. [1]

In cases where the adversarial label was not achieved, interestingly the classifier was still found to have guessed semantically similar labels [1]. For example, a fifth image of the baseball in Figure 5 might not have been classified as "espresso", but instead "coffee" or "tea."

Sharif et al. [14] present an other interesting application of 3-dimensional, perturbed adversaries. They show that a printed set of eyeglasses can be

used to fool a state-of-the-art facial recognition system, consequently allowing an attacker to impersonate another person.

3 Common defenses

Hinton et al. [6] first introduced a method called distillation, which uses the output scores of one or more DNNs to train a new neural network. In other words, the final model will have been trained with probabilistic "soft" labels, instead of the conventional use of binary "hard" labels. The method's original motivation was to reduce the size of the DNN architecture without compromising the model's accuracy, so that models could more readily be deployed to devices with limited computational capability. Papernot et al. [12] take distillation a step further, and argue that it can be used to defend against adversarial attacks.

Defensive distillation causes the final model to learn a structural similarity between the different predictions [12]. In our illustration in Figure 1, for example, the new model will learn that there's some structural similarity between the number two and the number seven. By using the output labels' probabilities for training, Papernot et al. [12] argue that the new model will be less prone to decisions based off of irrelevant, adversarial noise.

Madry et al. [10] suggest that defending against Projected Gradient Descent attacks (Figure 4) will secure the network against a wide range of other attacks. They point out that the capacity of the network needs to be larger in order to accommodate the more complex decision boundary. The intuition behind this requirement is presented in Figure 2.

Another valid way to defend against perturbed adversaries is to add correctly-labeled, perturbed samples to the training data [8]. However, this defence does not generalize well, since annotating one type of perturbation is not guaranteed to help defend against an other kind [9].

Other defences tend to either detect adversarial input or hide the gradient from attacks that leverage gradient descent. Most of these methods, however, seem to have since been broken. [9]

4 Conclusion

In this paper we presented different ways to mount attacks on a trained DL model, so that it fails to correctly classify an adversarially crafted sample. Namely, we directed our attention to cases where an attacker is able to perturb a legitimate input ever so slightly, and elicit an illegitimate output.

We noted that the magnitude of that perturbation needs to be constrained by some metric, as an unrestricted perturbation would be the same as allowing for the whole image to simply be replaced by another. We mentioned that one such informal restriction is human perception; a human should not be able to notice the perturbation.

After formalizing the perturbation attack, we introduced the following noteworthy attacks:

- Box-constrained Limited-memory Broyden–Fletcher–Goldfarb–Shann (Box-constrained L-BFGS) attack
- Fast Gradient Sign Method (FGSM) attack
- Projected Gradient Descent (PGD) attack
- CW attack
- Zeroth Order Optimization (ZOO) attack
- Boundary attack
- Synthesis of noise-resistant adversaries

Most of the known attacks leverage the model's internal gradients, and therefore require visibility into the internal workings of the DNN. However, we recognized that attackers might not always have access to this level of information, as models will often be deployed out of public reach.

For these cases, we underlined some black-box attacks. Most of them use the model's oracle to emulate a white-box model, even if the approximation results in some loss. Furthermore, we discussed the ZOO attack, which circumvented that loss in transferability.

We ventured into some of the more noteworthy defenses against these attacks, but only at a superficial level. We leave this to be considered by future work.

Our contribution is meant to help designers of DL models understand the inherent imperfections within DNNs, and how those imperfections may be leveraged by attackers.

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[4] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[7] Weiwei Hu and Ying Tan. Black-box attacks against rnn based malware detection algorithms. *arXiv preprint arXiv:1705.08131*, 2017.

[8] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[9] Blerta Lindqvist. Tricking adversarial attacks to fail. *arXiv preprint arXiv:2006.04504*, 2020.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[11] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[12] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[13] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[14] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

[15] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Usability analysis of IoT devices in a healthcare ecosystem

Oksana Baranova
oksana.baranova@aalto.fi

Tutor: Amel Bourdoucen

Abstract

KEYWORDS: IoT, security, usability, privacy, authentication

We are currently living in the era of the "Fourth Industrial Revolution", in which cyber-physical systems are massively introduced into people's everyday life to serve basic human needs. One of the key technologies in this trend is the Internet of Things, which encounters the problem of ensuring security and usability simultaneously.

IoT manufacturers release non-secure products in pursuit of profit and users are often unaware of the risks associated with such devices. The smart devices simplify some labor-intensive work and increase the comfort and convenience of use. However, the problem of ensuring the proper level of security and reliability still remains a serious obstacle to the development of IoT.

The paper gives an overview of the main vulnerabilities and attacks in IoT, and analyze security and usability paradox with a special focus on the healthcare system.

1 Introduction

With the advent of innovative technologies, the Internet of Things (IoT) has become an integral part of daily life. The IoT is a system of interconnected computer networks and objects connected to them with embedded sensors and software for collecting and exchanging data, with the ability to remotely monitor and control in an automated mode, without human intervention.

The Internet of Things is presented in many applications, such as smart homes, transportation, military, healthcare, and others. The National Institute of Standards and Technology (NIST) proposes dividing the IoT into five functional areas [14], such as connected devices, usual consumer devices, medical equipment and devices used in health care, "smart" buildings and "smart" production, including automated process control system.

The number of devices connected to the Internet grows each year and, as a consequence, has become a concern of cybersecurity. However, the implementation of security in IoT infrastructure is complicated because of the complex infrastructure [16]. Hence, software developers should take into consideration different hardware and software standards.

To develop devices securely, potential threats during the entire lifecycle of IoT device should be considered. The lifecycle includes phases as follows: manufacturing, operation and decommissioning. At the beginning of life, devices must be configured, by establishing a stable and secure communication channel with the control platform using authentication mechanisms. In addition to the main functionality, developers should guarantee the reliability and maintainability of the future device during its operation. At the end of the device's life cycle, it is necessary to ensure that the device is easily and safely replaced or removed if it becomes outdated or broken. Therefore, devices must remain reliable, efficient, sustainable, and safe at every stage.

As practice has shown, software developers are primarily concerned about functionality, design and practicality of the future smart devices. These devices constitute small computers connected to the Internet, which are not sufficiently secure in comparison to conventional computers. In this regard, there are many potential threats caused by existing vulnerabilities.

In order to address the issue of security and usability for IoT, this paper discusses the main vulnerabilities and potential threats of devices and

attacks on them with special focus on the medical field. In addition, this paper summarizes recent approaches in security protection methods for IoT and proposes solutions to solve the aforementioned problem.

2 IoT security threats and attacks

2.1 General threats and attacks

IoT devices produce large amounts of data. Different cloud platforms, such as Amazon Web Services, Google Cloud Platform are used to process this data. In this connection, IoT integration with cloud systems has become a serious security challenge to ensure privacy, integrity, availability and authenticity. Rajendran et al. [16] noted that presence of malware, viruses, and zero-day attacks stands for security problems in the IoT infrastructure. They evolve fast and make it difficult to apply preventive measures.

According to research [20], smart devices are usually deployed in limited spaces, which constraints the possibility of installing software updates and fixing security vulnerabilities. The author of the research highlighted the fact that smart devices cannot stay online for a long period of time and support intensive protocols because they need to sleep in order to save energy. Therefore, these devices have severe limitations, such as implementing firewalls or using strong cryptosystems to make secure connections with other devices [11].

Most security concerns in IoT include weak authentication, default credentials that remained after buying the device, lack of message encryption between devices, SQL-injections and poor handling of security updates.

Main threats for IoT devices are as follows:

- Some errors in the software of the network equipment and modules are persistent, which allows an attacker to exploit the vulnerability.
- The use of old equipment or protocols. Due to the explosive growth of Internet technologies, specialists are forced to create new devices with the use of old infrastructure, which is vulnerable by its nature in the 21st century.
- Threats that come from users. Many users have an indifferent attitude to computer security and they continue trying new features and services even if they have been warned about risks.

- Natural and technical threats.

According to the Dyn analysis investigation [8], an enormous Distributed Denial of Service (DDoS) attack was carried out on the company in October 2016 by botnets with a large number of IoT-infected devices. This incident became a signal about the insufficient security of IoT devices and systems in general, resulting in a deeper study to ensure security of IoT.

Another DDoS attack on the IoT devices was the Mirai Botnet Attack in September 2016. The attackers targeted IP cameras and routers with weak passwords in the login page by exploiting smart devices with ARC processors and using them as botnets. The created botnet launched a massive DDoS attack on the Internet.

Based on the above examples, it can be clearly seen that security plays a vital role in the IoT design and implementation.

According to research [19], during the initializing stage of a smart device an attacker may intercept and obtain security parameters or initial settings. In addition, an adversary can force device authentication since smart objects usually do not have prior knowledge of each other and cannot always distinguish between malicious and non-malicious networks.

Main attacks on the IoT are as follows:

- Network attacks both on the network itself and its elements: network communication devices, end terminals and on modules.
- Spoofing the network: listening to traffic, influencing information and conduction activities destabilizing production processes.
- Spoofing, analysis and changes in traffic, manipulation of confidential information, code injection and unauthorized access, certificate substitution and firmware update on devices.
- Use of social engineering to influence personnel in order to obtain confidential information that will expand the capabilities of the attacker.

2.2 Threats and risks in healthcare

According to statistics in [9], the nature of cyber security violations in the medical field has changed in recent years. From 2009 to 2015, a majority of the violations were presented by theft and loss of patient data and electronic health information. In early 2016, the growth of cybercrime with the help of ransomware increased to 320% compared to 2015 [17]. In recent years, the largest number of violations were related to hacker actions

and incidents in computer systems. From 2017 to 2020, the number of attacks on medical devices using ransomware has quadrupled. According to forecasts, by 2021 their number will increase five-fold compared with previous years, which will lead to an increasing demand from the medical institution for information security products. Based on the statistics, it can be clearly seen that there is an increasing demand for information security products in the healthcare system.

One of the main problems in the healthcare system relates to ensuring data privacy. Medical records contain all personal data of the patients and their health conditions, thus, represents valuable data for attackers. Another issue is the shutdown of vital and necessary systems for the patients through the control of the captured device, which can lead to unpleasant consequences for patients and doctors. For example, insulin pumps and implantable devices can be remotely controlled, therefore, they pose huge security and privacy risks.

Inadequate information security methods, outdated systems and programs, weak passwords and vulnerabilities open up a path for attackers to manipulate data from healthcare institutions. As stated by FDA, many vulnerabilities exist in drug vending machines and drug infusion pumps. In addition, with the use of cloud technologies and the Internet of things in healthcare, the number of vulnerabilities is growing, as shown in (Table 1). According to [18], the smart healthcare was a leading way for the cybercriminals in 2019 year.

Type of breach	**PHI breaches**	**records affected**	%
Hacking/IT incidents	109	13,383,846	80,6
Unauthorized access	130	1,641,006	9,9
Theft/loss	78	1,462,403	8,8
Improper disposal	7	125,730	0,7
Not categorized	1	unknown	N/A
Totals	325	16,612,985	100%

Table 1. Healthcare data breaches in 2016

Many institutions world wide have paid attention to the protection of medical data and medical equipment. The cost of one medical card on the black market can be 10-20 times more than credit card information. The high value of the data stored and used by medical institutions explains why in 2019, health care was subjected to more cyber attacks than other

industries [12].

Statistics and research have shown that the existing protection measures for medical institutions and medical devices are not enough to protect them properly from cybercriminals. Hence, a change of strategies for ensuring information security in the healthcare sector is required. With the implementation of "smart medicine", it is necessary to foresee all possible risks in advance, using the experience of countries who successfully implemented it.

Healthcare is an element of critical infrastructure that is attractive to attackers. With the transition to the format of Industry 4.0 and Medicine 4.0, with the development of the "Internet of medical things", the industry's vulnerabilities to cyber attacks grows and requires new standards, approaches and methods for ensuring the information security of devices throughout the entire lifecycle, in order to protect any person in a country.

2.3 Consequences of security attacks on the healthcare

There are growing number of advertisements appearing on the darknet for the sale of medical records and insurance policies, as such information is considered a valuable resource for cybercriminals. This data can be used for social engineering to deceive patients or their relatives and sell them non-existent products. Hackers can also modify data to carry out targeted attacks and deliberately complicate diagnostics.

Device problems, such as buggy applications, poor interaction of applications, and periodical device crashes can cause dangerous and unsafe states of the system. Currently, there are a considerable number of devices for changing motor activity, pressure, pulse that transmit data online. Karie et al. [10] consider that attacks lead to devastating effects on the system and its users. Intruders could bring significant security threats to a network with viruses or malware and cause patient's death. For example, life-sustaining devices can be easily disrupted by exploiting the connectivity capabilities that link together medicines delivery systems and medical records. Obviously, some of the patients may mistakenly take the wrong pills, or not receive them at all, if an attacker successfully compromised the device.

In October 2018, the FDA informed patients and their doctors about the cybersecurity vulnerabilities connected to Medtronic's cardiac implantable device programmers. Potential attackers could connect the wireless device to a nearby MiniMed insulin pump and change its settings that could

lead to over-deliver insulin to a patient, resulting in hypoglycemia, or to stop insulin delivery, leading to diabetes [7]. The issue was solved with the software update released by Medtronic, however, it was not the first vulnerability that was discovered by FDA. There were another four safety communications from FDA regarding vulnerabilities in cardiac devices.

As a rule, IoT medical devices are mostly used to diagnose a patient's condition on a regular basis, which forces users to take devices with them. For example, fitness trackers are usually geo-tagged and keep track of the user's location. If the device is not well secured, then the attacker can use the location data of the target group and abuse it afterward. In November 2017, there was an incident [2] with the fitness trackers recording the movements of runners and cyclists by mobile application Strava, which revealed the location of American military bases in the area of military conflicts in the Middle East.

3 Usability and security of IoT

The IoT technologies provide convenience in daily activities, safety and comfort. IoT solutions increase production efficiency several times, and the payback period of such projects in most cases does not exceed several months. For instance, the equipment of a Philips razor factory in Holland operates in an unlit room with 128 robots while the entire staff of the plant consists of nine employees [13].

To increase efficiency many smart devices typically implement specialized protocols and data formats. Although it reduces the load on resource-constrained devices, it limits the usability for users [22]. N. Karie et al. [10] consider some requirements that improve user interactions of constrained devices: the impact on the constrained devices should be at minimum value, interfaces should be easy in use, devices should interact with other devices, and devices should contain minimum configuration.

In the context of medicine, the usability of smart devices has become highly important. With IoT networks storing equipment data, healthcare professionals can quickly find available devices. Therefore, doctors will be able to direct patients to the right clinics to avoid queues and to receive medical care efficiently. In addition, monitoring sensors can be installed on the equipment, which will monitor the proper operation of the devices. In the event of a breakdown or poor-quality operation, the devices will be able to inform the staff about the problems themselves, which will help to

quickly eliminate them.

GE Healthcare [4] developed an AutoBed System, which helps nurses quickly and efficiently assign patients to beds, controls the number of patients and tracks their movements. The solution reduces waiting times in emergency situations that allows patients to obtain help and treatment faster.

Harsh V Thakkar et al. [21] proposed a usable system MED-IoT (Medical Confirmation System with Internet of Things) that analyzes the weight of the dosage that should be taken by the patient for a particular illness based on the prescription given by the user from the web page. The system is able to control consumption of medicines and to send notifications to the user when the weight is changed.

Littman and Kortchmar [1] adhere to an opinion that using IoT should be user-friendly both for user interface and for technology itself. By that authors meant that interfaces of the devices should be not only user-friendly but also better integrated. “If users need to learn different interfaces for their vacuums, their locks, their sprinkles, their lights, and their coffeemakers, it is tough to say that their lives have been made any easier”.

While IoT devices improve the quality of life, by enabling access to data anytime and anywhere, the potential threat level when using these devices is still remaining. Obviously, designing information security solutions that could be user-friendly at the same time is extremely challenging. A secure but inconvenient device will not provide security if not used. Software developers suggest approaches to ensure the security of user’s devices and whether or not to follow this process is up to the user. Therefore, it is essential to enforce IoT production to provide secure solutions regardless of the endorsements from users who often have no understanding of possible risks and threats caused by poor security solutions.

Information security and usability is an ever-growing problem, due to the small amount of research in this area, and the small number of implemented standards that could solve the urgent problems of device security. The lack of standards that would set up rules for secure device development open up opportunities for sophisticated attacks to be undetected in IoT networks [10].

4 Mitigation techniques

Potential threats in the domain have grown dramatically, and, at the same time, a great number of threats mitigation measures have also been researched on significantly [10]. Currently, many IoT frameworks and standards are being developed to support the developers to design the products for different consumer needs.

Chung et al. [5] proposed a new method for configuring security on-demand, with which the old security modules can be replaced with new ones without regenerating the device image. Another study [3] developed a Consumer Safety Index (CSI) with consumers and security experts to help consumers make decisions and drive better security measures for IoT development.

Producers of smart devices must provide support for software updates or security certifications, even after development and sale phases. This can be done by encrypting data, creating different levels of access to information, and controlling access from the particular device.

Local and external network connections should be tested for a Man-In-The-Middle Attack. For mobile and stationary devices, the communication protocol should be analyzed and after that the scheme should be supplemented. The website should be checked for forced encryption (https) and for common web vulnerabilities proposed in OWASP Top-10 [15].

At the legislative level, it is necessary to establish a standard between the type of data collected and the security of the Internet connection. For example, if the device collects only data on the patient's pulse and heart rate and has no technologies such as Bluetooth and Wi-Fi, then the device may release to the market without security standardization. However, if the device collects information about the user's location, then it must have additional security, such as the SSL certificate. Thus, the law ensures information security by allowing manufacturers to produce secure devices and prohibiting unsafe ones.

Another, not less important aspect for reducing potential information security risks, is that developers must adhere to the separation of hardware and computing part or an interface. The device should be divided into two, independent from each other, parts, one of which saves the settings and performs basic functions, and the other collects data and displays it on the Internet. The Internet-related part must have a proper security mechanism.

The research on striking a balance between usability and security of IoT [6] proposed a System Security Guidance for the IoT development on how to mitigate some of the security concerns, which are presented in the Figure 1.

System Security Question	Affects (C,I,A)	Improvement Recommendation	Security Attributes
1. Is it impossible for the feature to affect the health and safety of people or property?	availability	Provide safety guarantees for failure conditions	physical security
2. Does the feature require a local, physical interface to access it?	availability	Lock down all control and data input interfaces	remote control
3. Can authorized users or devices patch or update the feature in the future?	integrity	Build and maintain a patch / update service	Maintenance
4. Can only authenticated, authorized users or devices access the feature?	availability confidentiality	Construct and enforce authentication and authorization policies	authentication authorization
5. Is all received data automatically inspected and validated?	availability integrity	Validate all input	cleaning input validation
6. Are data transmissions encrypted and mutually authenticated?	confidentiality integrity	Use secure transport techniques	transport security
7. Does the feature avoid storing personally identifying information, tokens, or passwords?	confidentiality integrity	Be deliberate and careful with secure storage of credentials	sensitive data
8. Is any stored data only accessible after authentication by an authorized user or device?	availability confidentiality	Consider encrypting data at rest	data storage encryption authorization
9. Does the feature routinely log use and errors in a way that authorized users can inspect the logs?	logging integrity	Store log data securely	auditing error investigation
10. Is the source code available for inspection by a third party?	integrity	Adopt open source principals where appropriate, and accepted vulnerability disclosure practices	transparency

Figure 1. System Security Guidance

5 Conclusion

IoT has tremendous potential for widespread usage that will improve everyday life only in the case if the smart user-friendly devices fully meet the requirements of security and privacy. Therefore, it is essential to implement proper security frameworks and standards in order to ensure the resistance of IoT things to various cyber attacks.

The paper states that by setting required practices, patterns, and principles, security and usability could be improved synergistically.

References

[1] Michael Littman, Samuel Kortchmar. The Path To A Programmable World, June 2014. Available at https://footnote.co/the-path-to-a-programmable-world/, last accessed on 12/11/2020.

[2] BBC. Fitness App Strava Lights Up Staff At Military Bases, January 2018. Available at https://www.bbc.com/news/technology-42853072, last accessed on 25/11/2020.

[3] J. M. Blythe and S. D. Johnson. The consumer security index for IoT: A protocol for developing an index to improve consumer decision making and to incentivize greater security provision in IoT devices. In *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, pages 1–7, 2018.

[4] Yash Chetan Chitalia. Autobed: A Web-Controlled Robotic Bed, April 2016. Available at https://sites.gatech.edu/hrl/autobed-a-web-controlled-robotic-bed/, last accessed on 25/11/2020.

[5] B. Chung, J. Kim, and Y. Jeon. On-demand security configuration for IoT devices. In *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1082–1084, 2016.

[6] Saurabh Dutta. *Striking a balance between usability and cyber-security in IoT devices*. PhD thesis, 01 2017.

[7] FDA News Release. FDA warns patients and health care providers about potential cybersecurity concerns with certain medtronic insulin pumps, 2019. Available at https:www.fda.govnews-eventspress-announcementsfda-warns-patients-and-health-care-providers-about-potential-cybersecurity-concerns-certain, last accessed on 22/11/2020.

[8] Scott Hilton. Dyn Analysis Summary Of Friday October 21 Attack, 2016. Available at http://dyn.com/blog/dyn-analysissummary-of-friday-october-21-attack/, last accessed on 12/11/2020.

[9] HIPPA Journal. Healthcare Data Breach Statistics. Available at https://www.hipaajournal.com/healthcare-data-breach-statistics, last accessed on 10/11/2020.

[10] N. M. Karie, N. M. Sahri, and P. Haskell-Dowland. IoT Threat Detection Advances, Challenges and Future Directions. In *2020 Workshop on Emerging Technologies for Security in IoT (ETSecIoT)*, pages 22–29, 2020.

[11] Ximeng Liu, Yang Yang, Kim-Kwang Raymond Choo, and Huaqun Wang. Security and privacy challenges for internet-of-things and fog computing. *Wireless Communications and Mobile Computing*, 2018:1–3, 2018.

[12] Steve Morgan. Cybersecurity Almanac: 100 Facts, Figures, Predictions And Statistics, 2019. Available at https://cybersecurityventures.com/cybersecurity-almanac-2019/, last accessed on 02/11/2020.

[13] Jim Nash. Robots Hone Shaver Line at Philips, June 2015. Available at https://www.roboticsbusinessreview.com/manufacturing/robots_hone_shaver-_line_at_philips/, last accessed on 23/11/2020.

[14] NIST. Internet Of Things. Available at https://www.nist.gov/topics/internet-things-iot, last accessed on 02/11/2020.

[15] OWASP. Top 10 Web Application Security Risks, 2017. Available at https://owasp.org/www-project-top-ten/, last accessed on 27/11/2020.

[16] Gowthamaraj Rajendran, R S Ragul Nivash, Purushotham Parthiban Parthy, and S. Balamurugan. Modern security threats in the Internet of Things (IoT): Attacks and Countermeasures. In *International Carnahan Conference on Security Technology (ICCST)*, pages 1–6. IEEE, 2019.

[17] Omnibus HIPPA Rulemaking. The final rule in a federal register. Available at https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/combined-regulation-text/omnibus-hipaa-rulemaking/index.html, last accessed on 10/11/2020.

[18] Health IT Security. 82 % IoT Devices of Health Providers, Vendors Targeted by Cyberattacks. Available at https://healthitsecurity.com/news/82-iot-devices-of-health-providers-vendors-targeted-by-cyberattacks, last accessed on 22/11/2020.

[19] M. Sethi, J. Arkko, and A. Keränen. End-to-end security for sleepy smart object networks. In *37th Annual IEEE Conference on Local Computer Networks - Workshops*, pages 964–972, 2012.

[20] Mohit Sethi. Security for Ubiquitous Internet-Connected Smart Objects. Aalto University Publication Series Doctoral Disserations; 278/2016, pages 113 + app. 69. Aalto University; Aalto-yliopisto, 2016.

[21] Harsh V Thakkar, Jigar Chauhan, Viral Trivedi, and Urvi Jolapara. MED-IoT: A Medicine Confirmation System. *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, pages 1–5, 2018.

[22] F. Van den Abeele, E. Dalipi, I. Moerman, P. Demeester, and J. Hoebeke. Improving user interactions with constrained devices in the web of things. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 153–158, 2016.

Orchestration Techniques for Fog Computing

Leo Kivikunnas
leo.kivikunnas@aalto.fi

Tutor: Vesa Hirvisalo

Abstract

Orchestration is a important topic in fog computing research. This paper provides a review of the special requirements fog computing places on orchestration tools and the current state of fog computing orchestration.

KEYWORDS: IoT, Fog, Edge, Kubernetes, Orchestration

1 Introduction

The internet of things (IoT) has caused massive growth in the number of connected devices. The cloud computing paradigm pairs well with IoT, providing the horsepower to process the data gathered from IoT devices [1]. However, the shortcomings of cloud computing in demanding IoT applications with e.g. real-time requirements have given rise to new computing paradigms like edge computing and fog computing, that are more suitable for these demanding scenarios [2].

One of the hot topics for research in fog computing is orchestration. We would like to enjoy the ease and flexibility of orchestration techniques and tools used in the cloud, but without taking too big of a hit in performance. This paper is a survey of orchestration techniques fog computing.

We see what are the special requirements and limitations of orchestrating fog computing for IoT applications. Additionally, we review research that evaluates how well do currently popular orchestration tools like Kubernetes fit into this use case.

The remainder of this paper is structured as follows. In section 2, we introduce fog computing and compare it to cloud computing. Section 3 introduces orchestration by the example of Kubernetes. Next in section 4, we take a look at issues and special requirements that arise from orchestrating fog computing. In section 5 we see how well does Kubernetes fare for this use case. Finally, section 6 concludes this paper.

2 Fog Computing

The OpenFog Consortium defines fog computing as “A horizontal, system-level architecture that distributes computing, storage, control, and networking functions closer to the users along a cloud-to-thing continuum." [3] Fog computing sits in the gap between IoT devices and the cloud. Like cloud computing, it aims to provide computing, storage, networking, and data management for applications. [4]. Fog computing has developed from the inability of cloud computing to satisfy the needs of some demanding IoT applications [5]. The term was initially coined in 2012 by Bonomi et al [6].

It is often technologically challenging to get data from IoT devices to the cloud. IoT devices are often on unreliable internet connections. A massive amount of devices can generate data at volumes that would require ludicrous bandwidth. In some applications, the latency requirements are so strict that the additional latency from going to the cloud and back is unacceptable. There might also be privacy concerns related to sending data to the cloud. [2]

On the other hand, processing the data on the IoT devices themselves is also often unfeasible. They simply do not have the required processing power. To overcome these limitations a different paradigm for computing is required. [5]

In fog computing, the nodes can be located anywhere between the IoT devices and the cloud, providing a more flexible platform for applications compared to the cloud. It tackles the above-mentioned issues with cloud computing and IoT [7]. There are 2 major differences between cloud and fog computing:

- The geographical spread of the nodes. Fog computing is decentralized, nodes can be spread geographically and are available in large numbers. Cloud computing resources are centralized in large data centers. This means that fog computing resources can be accessed through the network from the edge to the core whereas cloud computing must be accessed through the network core. This means that continuous internet connectivity is often not essential for fog computing applications. [2]

- The scale of available computing resources. In practice, fog computing nodes could be e.g. small servers, routers, or switches. Fog computing offers less computing power than the cloud. On the other hand, fog computing nodes do not require lots of power and have a small form factor. This enables the previous point, which is the main advantage of fog computing. [7]

Fog computing is intended to complement computation in the cloud [3]. Fog nodes can be used to implement functions that require less computing power and possibly benefit from reduced latency. Figure 1 shows the fog layer positioned between the edge and the cloud in the context of an IoT data processing application. Fog nodes can be used i.e. for preprocessing data before sending it on to the cloud for more processing.

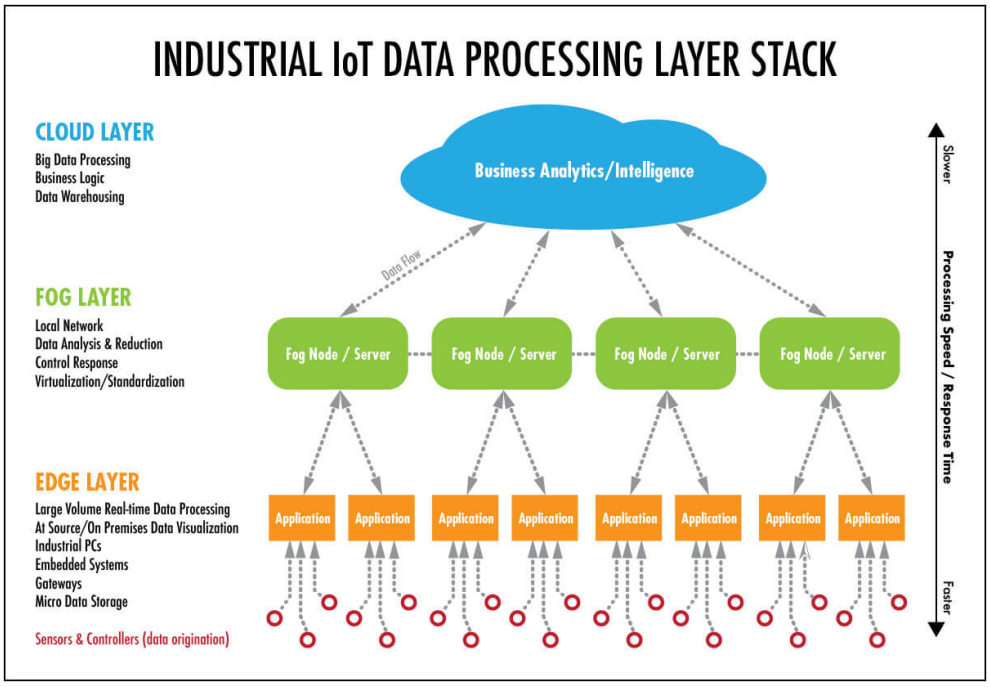


Figure 1. IoT data processing stack [8]

It is important to note that there is no universally accepted definition

for fog computing and the term is sometimes used interchangeably with similar concepts such as edge computing or cloudlets [2].

3 Orchestration and Kubernetes

Redhat defines orchestration as "the automated configuration, management, and coordination of computer systems, applications, and services" [9]. There are several orchestration tools out there based on different virtualization technologies. Traditional virtual machines are too resource hungry for some fog nodes, so lightweight virtualization is preferred. Unikernels are a promising new lightweight virtualization method that has also been proposed for use in fog applications [2]. But in this paper, we will concentrate on container-based virtualization, since it is the most widely used and researched method. Specifically from now on, we will focus mostly on Kubernetes, again as it is the most popular container orchestration tool currently, and there have already been some efforts to use it for fog orchestration [2].

With the mass adoption of container-based virtualization, Kubernetes has become the most widely used orchestration tool out there [10]. Kubernetes was developed at Google and was initially released in 2014 [11]. Kubernetes provides developers with [12]

- Service discovery and load balancing.
- Storage orchestration.
- Automated rollouts and rollbacks.
- Automatic bin packing.
- Self-healing.
- Secret and configuration management.

A Kubernetes deployment is called a cluster. A Kubernetes cluster is made up of worker machines, called nodes. The nodes run containerized applications in pods. A Pod represents a set of running containers [12]. A service in Kubernetes is an abstraction of an application running on a set

of pods as a network service. [13] Fig. 2 represents a Kubernetes cluster.

The part in Kubernetes which manages pods and nodes is called the control plane. The control plane is responsible for example, for scheduling, as well as detecting and responding to cluster events like starting up a new pod when there are not enough replicas of a deployment. [12]

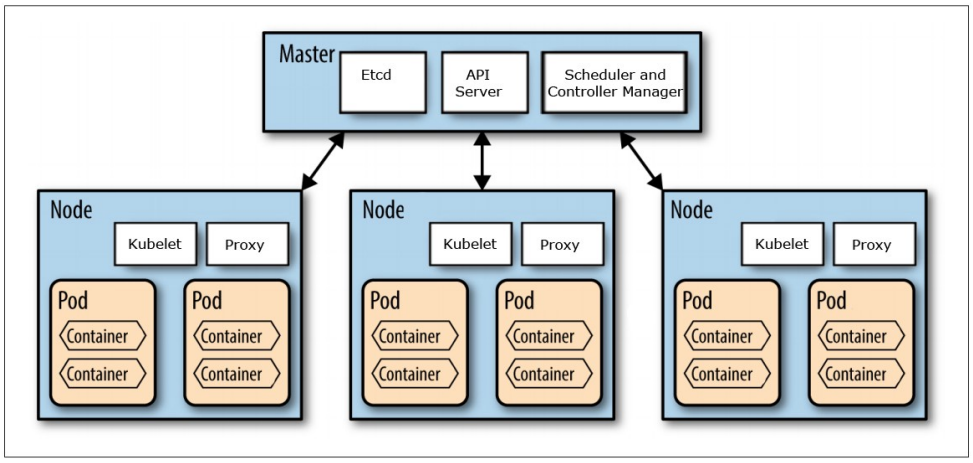


Figure 2. A Kubernetes cluster [14]

Even though Kubernetes was originally developed for orchestration in pure cloud environments it has also been adopted for fog and edge computing.

4 Fog orchestration

IoT applications have diverse requirements in terms of scalability, security, and reliability. The key issue in fog orchestration is the diversity among the fog nodes. Diversity in location, configuration, and served functionalities increase dramatically compared to the cloud. The issue is then to configure the system so that it fulfills all the requirements and preferably in an efficient manner. This is especially challenging since in fog computing applications one has to often consider non-functional requirements like security, quality of service (QoS), and latency. [15]

Fog nodes can be very low power devices, and the orchestrator needs to run the worker nodes on these devices. It might even be required to run the master node on a low power device. Thus, the orchestrator application needs to be lightweight and preferably modular and configurable, so unnecessary features can be disabled to improve performance.

In IoT systems that comprise of heterogeneous devices, selecting optimal components becomes increasingly complicated. Some applications

can only operate on a specific architecture. In some cases, the application might even require specific hardware accelerators to function. The orchestrator needs not only to cater to such functional requirements, it must be able to perform adequately for increasingly complex systems that change dynamically. The orchestrator assembles systems from cloud resources, fog nodes, and sensors. It needs to take into account geographic distributions and constraints to provision complex services correctly and efficiently. [15]

Particularly challenging in the fog environments is predicting, detecting, and resolving issues related to the dynamicity of the available resources. Fog nodes can frequently join or leave the network changing the available resources from which the orchestrator builds the application flows. This means that application performance can change very dynamically owing to this transient behavior of the system. This leads to a strong requirement for automatic and intelligent reconfiguration of the topological structure and assigned resources within the workflow. [15]

5 Kubernetes for fog orchestration

As mentioned earlier, Kubernetes originates from the cloud orchestration world, but due to its popularity and attractive feature set, it has also been adopted for fog orchestration. However, since it has been developed with the cloud in mind, Kubernetes is not perfectly suitable for fog applications.

Even though Kubernetes system requirements are not too steep, more lightweight distributions of Kubernetes have emerged. They allow one to run Kubernetes on low power devices. This is beneficial and sometimes even necessary to utilize some fog nodes. The most prevalent of these distributions is K3s. K3s is a lightweight fully compliant Kubernetes distribution by Rancher. K3s has a small memory footprint and a binary size of less than 100MB [16]. This is achieved by removing some legacy and cloud-centric features from the main distribution. Instead, they are provided as addons if needed [17].

The authors of [18] define 3 basic requirements for a fog orchestrator. Their 3 requirements are:

- The orchestrator needs to support low power nodes. When a new node is added to the cluster it needs to join the cluster seamlessly. Only minimal required software should be installed on the node.

- As the fog nodes can be physically widely spread, the orchestrator should support scheduling containers to specific nodes. E.g. in an industrial setting, for a manufacturing machine that is connected to a fog node, the node is ideally the closest one to the machine to minimize latency.

- The orchestrator needs to be aware of the capabilities of the nodes when making scheduling decisions. This can mean e.g. the orchestrator needs to consider factors like CPU power, available memory, or even what CPU architecture the node has. Additionally, sometimes the containers might need to access IO devices i.e. serial ports. Thus, the containers need to support mapping other resources than just the CPU and networking.

They also evaluate Kubernetes based on their 3 criteria. They state that Kubernetes meets requirements 1 and 2, but fails on 3. According to them, in Kubernetes, it is impossible to access the resources of the node since the containers are running inside the sandbox of the pod. However, it seems that they are not entirely correct there. According to the discussion in [19] it is possible to give containers running inside pod access to the node's resources, but that requires running the containers in a privileged mode which has security implications.

5.1 Scalability of the control layer

In fog computing applications, the control layer has to coordinate a massive amount of nodes. The authors of [20] argue that a conventional central controller in an orchestrator does not scale well to massive fog applications.

They take methods discussed in cloud computing research as a starting point for tackling the problem in fog computing. The main two approaches in the cloud world can be classified as hierarchical controllers and flat controllers. They both employ a distributed controller infrastructure and a divide and conquer method, where the nodes are divided into multiple domains and each domain has its own internal main controller that manages the domain. The controllers use periodic synchronizations to communicate the global view of the system to all the local controllers. Compared to the flat controller approach, the hierarchical controller employs additional higher-level controllers in addition to the local domain controllers to take advantage of multi-level processing to reduce the workload on the lower layers.

They argue that the two above mentioned methods could be also applied to fog computing. The hierarchical system is usable when the fog is a single operator system e.g. an LTE-based mobile network. However, in a larger scale fog network the domains can be owned by different parties e.g. homeowners, universities, factories, etc. It may then be impossible to find a jointly trusted higher level authority for the higher-level controllers. In these cases, the flat option is more suitable.

Still, they state that it is not practical for a massive fog network to be treated as completely flat. Since each controller maintains the view of the system via periodic updates from other controllers the volume of the inter-controller messaging of the network can be huge and adding new domains leads to exponential growth in this volume.

The design of inter-controller communications becomes a critical issue that arises from fog computing orchestration. They conclude that a good trade-off between sharing necessary information to conduct fog computing and the privacy of domains needs to be achieved for a solution to be viable.

Kubernetes currently supports 5000 nodes per cluster and multiple clusters so on paper the control layer should at least scale to meet the needs of current fog applications [21]. However, the author was unable to find research evaluating the suitability of the Kubernetes control layer for fog applications.

5.2 Orchestrating latency sensitive applications

Kubernetes has been developed with the assumption of orchestrating cloud systems, where the nodes are heterogeneous e.g. in terms of processing power [22]. Kubernetes is designed to balance the load evenly across the nodes, but it does not take into account the implications this might have to e.g. latency in a fog environment. Kubernetes is not aware of network topology or the latencies between nodes [13]. Even though it is possible to force Kubernetes to schedule containers to specific nodes, it is not ideal. As stated earlier one key challenge is the dynamicity of the fog environment and hardcoded scheduling to specific nodes does not sit well with that. In complex scenarios, this method might be too cumbersome or unreliable [18]. Thus using plain Kubernetes is not feasible in some real-time systems. Several papers have proposed modified versions of Kubernetes that complement its feature set to make it suitable for systems with latency requirements.

The authors of [22] research three approaches to using Kubernetes for

fog orchestration. First, they introduce the fog application allocation problem (FAA) which formalizes the job of a scheduler that takes into account latencies. They also introduce the greedy border allocation (GBA) algorithm originally defined in [23] by the same group. The GBA algorithm was shown to perform well compared to other algorithms [23]. Then, they compare three different approaches to implementing the GBA algorithm in a Kubernetes based orchestration system.

They first introduce a decoupled solution. It features a separate scheduler module that implements the GBA algorithm and communicates with Kubernetes to make resource allocation decisions. They state that the major advantage of this approach is the decoupling from Kubernetes. This prevents a lock into Kubernetes. This approach could be relatively easily adapted to some other orchestrator e.g. Docker Swarm. The major downside is also due to decoupling. The decoupling causes situations where features available in the orchestrator need to be re-implemented in the FAA module. This requires extra development and maintenance effort. They give node resource capacity and fault tolerance as examples of this. Interestingly, the authors of [24] don't mention this downside in a similar implementation where they add an external scheduler to Kubernetes. They first use Kubernetes to pre-schedule and finally, the suitable nodes are sent for the scheduler module for the final scheduling decision. They seem to be able to utilize all the features of Kubernetes and yet have a separate custom scheduling module.

The second option is called native. It relies on pre-existing Kubernetes features for resource allocation. Since Kubernetes is not aware of the network topology an adapted version of the GBA algorithm, that is usable in Kubernetes, K8S-GBA is introduced. Compared to the decoupled option, native avoids the need to implement redundant functionality. The main drawback is also the opposite of the decoupled option. Native means complete lock into Kubernetes. Also if in the future a better alternative to the GBA algorithm emerges, it may or may not be possible to implement a similar performing alternative to it in Kubernetes. However, regardless of the negatives, they consider this to be the best option since it offers a good balance between development effort and features.

They call the third option modified. It relies on a modified version of the Kubernetes scheduler. The authors did not try this version in practice, since according to them it would require significant modifications to the Kubernetes source code to make it aware of network topology. Com-

pared to the two other options, modified has the best potential to support advanced features. However, as stated earlier would require significant development efforts to realize.

6 Conclusion

This paper covered orchestration techniques for fog computing applications. The objective was to conduct a review of recent research and offer a look into the current state of fog orchestration.

The background sections 2 and 3 briefly covered fog computing and orchestration. Fog computing was compared to cloud computing and it was shown how fog computing complements cloud computing. Orchestration was covered from the point of view of Kubernetes. The features of Kubernetes and the structure of a Kubernetes cluster were discussed.

The special requirements that fog computing applications place on orchestrators were discussed in section 4. The main challenge that fog computing applications introduce is their dynamic nature. Many orchestration tools are designed for cloud applications where the nodes are often homogenous. In fog computing applications, nodes are often very heterogeneous in terms of performance, location, and functionality. In addition, nodes often dynamically leave and join the network. It was also noted that in some fog computing applications the orchestrator needs to also consider non-functional requirements like QoS and latency.

In section 5, Kubernetes was analyzed in terms of suitability for fog orchestration. Kubernetes was deemed suitable for a basic setup, however, it might not perform well in more demanding applications. Two possible issues for using Kubernetes as a fog orchestration tool were introduced: scalability of the control layer and the Kubernetes scheduler being unaware of network topology. Papers discussing these topics were briefly covered.

From the review, we can conclude that fog computing orchestration and the whole fog computing field are still taking baby steps. There are several papers discussing the topic on a theoretical level covering formal definitions, requirements, and specifications for fog computing and fog orchestration tools. However, actual implementations of said tools are rare and still in a very experimental state. That being said, so are many of the fog applications. There is still a long way to go to achieve the grandiose visions of some researchers, where autonomous vehicle cruise around cities

lending their computing capabilities to different fog networks as they go. To realize this vision the development of orchestration tools is crucial and will continue to be a major focus of fog computing research and development.

References

[1] Ben Zhang, Nitesh Mor, Jack Kolb, Douglas Chan, Nikhil Goyal, Ken Lutz, Eric Allman, John Wawrzynek, Edward A. Lee, and John D. Kubiatowicz. The cloud is not enough: Saving iot from the cloud. In *HotCloud15*, July 2015.

[2] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P. Jue. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, 98:289 – 330, 2019.

[3] OpenFog Consortium Architecture Working Group. Openfog reference architecture for fog computing. *OPFRA001.020817*, 2017.

[4] IEEE. Ieee standard for adoption of openfog reference architecture for fog computing. *IEEE Std 1934-2018*, pages 1–176, 2018.

[5] Cisco. Fog computing and the internet of things: Extend the cloud to where the things are. *Cisco White Paper*, 2015.

[6] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, page 13–16, New York, NY, USA, 2012. Association for Computing Machinery.

[7] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan. Fog computing: Survey of trends, architectures, requirements, and research directions. *IEEE Access*, 6:47980–48009, 2018.

[8] Cloud, fog and edge computing – what's the difference? https://www.winsystems.com/cloud-fog-and-edge-computing-whats-the-difference/. Accessed: 2020-01-10.

[9] What is orchestration? https://www.redhat.com/en/topics/automation/what-is-orchestration. Accessed: 2020-01-10.

[10] Asaf Yigal. The rise of kubernetes in 2017. https://logz.io/blog/rise-kubernetes-2017/. Accessed: 2020-01-10.

[11] Brendan Burns. The history of kubernetes & the community behind it. https://kubernetes.io/blog/2018/07/20/the-history-of-kubernetes-the-community-behind-it/. Accessed: 2020-01-10.

[12] What is kubernetes? https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/. Accessed: 2020-01-10.

[13] Service. https://kubernetes.io/docs/concepts/services-networking/service/. Accessed: 2020-12-10.

[14] How to install kubernetes(k8) in rhel or centos in just 7 steps. https://miro.medium.com/max/1177/0*Vu9u8x8M-dYKjkO0.jpg. Accessed: 2020-01-10.

[15] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos. Fog orchestration for internet of things services. *IEEE Internet Computing*, 21(2):16–24, 2017.

[16] K3s - lighweight kubernetes. https://rancher.com/docs/k3s/latest/en/. Accessed: 2020-16-11.

[17] Arseni Leskinen. Applicability of kubernetes to industrial iot edge computing system; kuberneteksen soveltuvuus teolliseen esineiden internet reunalaskentajärjestelmään. G2 pro gradu, diplomityö, 2020-10-19.

[18] S. Hoque, M. S. De Brito, A. Willner, O. Keil, and T. Magedanz. Towards container orchestration in fog computing infrastructures. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 294–299, 2017.

[19] add support for host devices. https://github.com/kubernetes/kubernetes/issues/5607. Accessed: 2020-14-10.

[20] Y. Jiang, Z. Huang, and D. H. K. Tsang. Challenges and solutions in fog computing orchestration. *IEEE Network*, 32(3):122–129, 2018.

[21] Building large clusters. https://kubernetes.io/docs/setup/best-practices/cluster-large/. Accessed: 2020-29-10.

[22] R. Eidenbenz, Y. Pignolet, and A. Ryser. Latency-aware industrial fog application orchestration with kubernetes. In *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 164–171, 2020.

[23] M. Suter, R. Eidenbenz, Y. Pignolet, and A. Singla. Fog application allocation for automation systems. In *2019 IEEE International Conference on Fog Computing (ICFC)*, pages 97–106, 2019.

[24] J. Santos, T. Wauters, B. Volckaert, and F. De Turck. Towards network-aware resource provisioning in kubernetes for fog computing applications. In *2019 IEEE Conference on Network Softwarization (NetSoft)*, pages 351–359, 2019.

Normalizing flows for graph structured data generation

Vlada Strazdina
vlada.strazdina@aalto.fi

Tutor: Anirudh Jain

Abstract

Molecular graph generation with the desired chemical properties based on deep structured data generation provides a promising way for the drug discovery process accelerating. State-of-the-art generating molecules methods contain normalizing flows that allow to achieve precise high validity in molecular designing. However, these methods require an oracle to verify at each subsequent stage whether the proposed action in the molecular graph will lead to a real molecule. Since the process of verifying the oracle is a rather slow process, in this paper we want to explore the importance of the oracle and normalizing flows in the molecular generation methods. Specifically, we compare state-of-the-art flow-based models and not flow-based models and their metrics with and without the oracle. This paper presents that the oracle helps to generate molecules with higher validity, and flow-based models show better metrics in molecular graph generation.

KEYWORDS: normalizing flow, oracle, validity, uniqueness, molecular graph generation.

1 Introduction

Creation of new molecules with desired properties is a fundamental challenge in various fields such as drug discovery or material science. Finding molecules with the essential chemical properties demands several years of development [1], since the entire chemical search space is huge. Currently, deep learning in drug discovery has gained significant attention. Generative deep graph models automate molecular development. State-of-the-art generating molecules methods contain flows in the models. Flow-based methods allow to achieve precise likelihood maximization in molecular designing.

However, state-of-the-art methods of generating molecules [17, 19] require an oracle for graph structured generation. These methods are usually slow since the oracle verifies at each subsequent stage whether the proposed action on the molecular graph will lead to a real molecule. The state-of-the-art methods discard invalid actions by checking with cluttered if-else blocks in the code. Slow execution of cluttered code is a major problem of these methods.

In this paper, we explore the importance of normalizing flows and the oracle in the state-of-the-art graph generating methods. We compare metrics of different flow-based and not flow-based models with and without the oracle. This paper presents that flow-based models show better performance metrics in molecular graph generation, and the oracle helps to generate molecules with higher validity.

This paper is organized as follows. Section 2 reviews the main principles for molecular generation and flow-based models. Section 3 discusses the concept for the oracle, autoregressive flows, graph neural networks and state-of-the-art graph generation methods. Section 4 shows the importance of normalizing flows and the oracle in graph generation. Section 5 visualises molecule generation by implementation of property optimization experiment. Finally, section 6 presents some concluding remarks.

2 Related work

Molecular generation. Early works for molecular generation are based on a simplified molecular-input line-entry system (SMILES), which represents molecular graphs in human-readable strings. Recently, recurrent neural networks (RNNs) have evolved into impressive generative models

in chemistry. Segler [15] combined SMILES and RNNs to demonstrate a molecular graph generation model based on retraining small sets of already known active components.

Variational autoencoder (VAE) models generate molecular graphs instead of linear SMILES strings. Junction Tree VAE (JT-VAE) [5] forms a framework over chemical substructures and integrates them into a molecular graph. GraphVAE [17] predetermines the maximum graph size and outputs the result as a fully connected graph.

Generative adversarial network (GAN) [1] models achieved great progress in chemical property scores and reduced training time regarding to the SMILES. De Cao [2] proposed the MolGAN model, which predicts the structure of the molecular graph at once, rather than sequentially.

Autoregressive (AR) models generate molecules sequentially and additionally check the accuracy at each stage of generation. This algorithm is represented in the Graph Convolutional Policy Network (GCPN) [19].

Flow-based models.

Unlike GANs ans VAEs, flow-based models achieve precise likelihood maximization. Due to the reversibility of flow models, new graphs are generated by feeding a latent vector into the same model in reverse order. To our best knowledge, currently, there are only four flow-based models for generating molecular graphs.

To provide precise likelihood maximization, the deep flow model GraphNVP [10] divides the creation of a graph into two steps. First, this model generates a graph structure using an adjacency tensor. Secondly, GraphNVP generates node attributes based on the graph structure.

Graph residual flow (GRF) [4] represents the generation of a molecular graph in a one-shot manner. GRF does not demand splitting the latent vector and updating all node attributes at every level.

The molecular graph generation model GraphAF [16] simultaneously incorporates the benefits of both autoregressive and flow-based approaches. GraphAF dynamically generates nodes and edges based on existent structures of subgraph.

One of the most recently developed flow-based graph generative models was presented by Zang [21] and called MoFlow. To create molecular graphs, this model first generates bonds (edges), and secondly, a Moflow model forms atoms (nodes) specified by a recent graph conditional flow. Finally, the Moflow model assembles bonds and atoms into a chemically valid molecular graph, adjusted for validity.

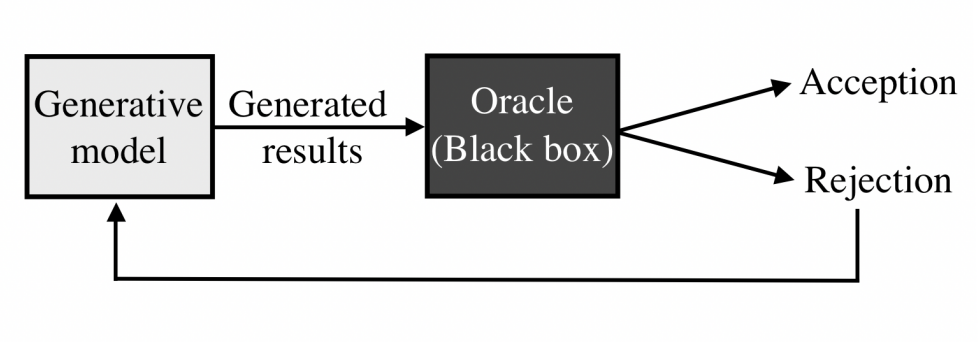


Figure 1. The oracle represents a black box system.

3 Preliminaries

3.1 Oracle

State-of-the-art methods of generating molecules require an oracle to verify at each subsequent stage whether the proposed action on the molecular graph will lead to a real molecule . The oracle represents a black box system that receives the output of a generative model and accepts or rejects a sample based on chemical rules [11]. If the oracle rejects, the generative model is re-run to draw new samples. The oracle helps to improve the molecular predictions. It checks the state of the molecular graph after each step and only accepts the correct validity. The oracle is demonstrated in Figure 1.

3.2 Autoregressive Flows

Kingma [7] represented autoregressive models as a form of a normalizing flow. The normalizing flow [9] determines a parameterized invertible deterministic transformation from a simple distribution $\mathcal{E}$ (e.g. Gaussian distribution) into real-world observational space $\mathcal{X}$ (e.g. speech) using the sequence of invertible and differentiable mappings. Let $f : \mathcal{E} \to \mathcal{X}$ be an invertible transformation where $\epsilon \sim p_{\mathcal{E}}(\epsilon)$ is the base distribution. Normalizing flows as a generative model involves two key processes: computing data likelihood and sampling. The first process calculates the exact density $p_x(x)$ in a given a datapoint x, by inverting transformation f, $\epsilon = f_\theta^{-1}(x)$. The second process chooses x from the distribution $p_X(x)$ by the first sample $\epsilon \sim p_{\mathcal{E}}(\epsilon)$ and then performs a direct transformation $x = f_\theta(\epsilon)$. The density function of real-world data x, *i.e.*, $p_X(x)$, can be calculated via the change-of-variables formula [16]:

$$p_X(x) = p_{\mathcal{E}}\left(f_\theta^{-1}(x)\right)\left|det\frac{\partial f_\theta^{-1}(x)}{\partial x}\right|. \tag{1}$$

To perform these processes more efficiently, the function f_θ needs to be invertible using a simply computable Jacobian determinant. Autoregressive flows (AF) comprise a triangular Jacobi matrix where determinant is calculated linearly [12]. Let g_μ and g_α be unconstrained and positive scalar functions of $x_{1:d-1}$ respectively for calculating the mean and deviation. The given $x \in R^D$, where D is the dimension of observation data, autoregressive conditional probabilities can be represented as Gaussian distributions:

$$p(x_d|x_{1:d-1}) = \mathcal{N}(x_d|\mu_d, (\alpha_d)^2),\ where\ \mu_d = g_\mu(x_{1:d-1};\theta), \alpha_d = g_\alpha(x_{1:d-1};\theta), \tag{2}$$

Considering that $\frac{\partial x_i}{\partial \epsilon_j}$ is non-zero only for $j \leq i$, the Jacobian matrix is triangular in AF. Consequently, we can effectively compute the determinant via $\prod_{d=1}^{D} \alpha_d$. In particular, all individual scalar affine transformations can be applied in parallel to calculate the base density, each of them depends on previous variables $x_{1:d-1}$. In practice, these transformations can be realized in the form of neural networks. Then the affine transformation of AF implemented as:

$$f_\theta(\epsilon_d) = x_d = \mu_d + \alpha_d \cdot \epsilon_d;\ f_\theta^{-1}(x_d) = \epsilon_d = \frac{x_d - \mu_d}{\alpha_d}. \tag{3}$$

3.3 Graph Neural Networks

Currently, there are many graph neural network (GNN) architectures that are used for learning graph representations [6, 14]. Kipf [8] proposed the graph convolutional network (GCN) that linearly scales according to the number of graph edges and explores hidden layers which encode the nodes features and the structure of the local graph. Velickovic [18] introduced graph attention networks (GATs). Architectures of GATs leverage masked layers and overcome the prior methods shortcomings based on graph convolutions.

Each of the architectures described above represents a molecule in the form of a graph $G = (A, X)$, in which A is an adjacency tensor and X is a matrix of node feature. Supposing n is the number of nodes in the graph, b and d are the quantity of different types of edges and nodes, respectively,

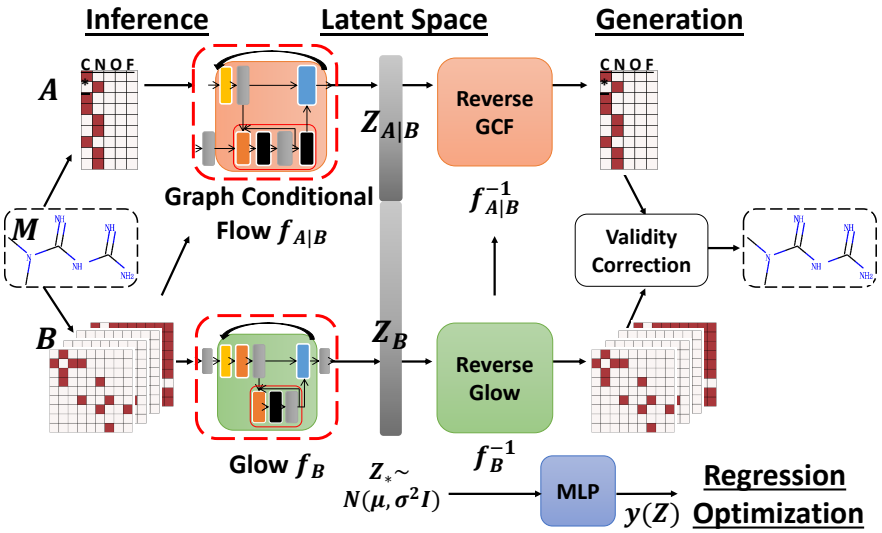


Figure 2. MoFlow model [21].

then $A \in \{0,1\}^{n\times n\times b}$ and $X \in \{0,1\}^{n\times d}$. Provided that a bond of type k exists between the nodes i^{th} and j^{th}, then $A_{ijk} = 1$ [16].

3.4 State-of-the-art graph generation

Recently, Zang [21] represented the most state-of-the-art flow model for molecular graphs generation called MoFlow (Molecular Flow). This model generates molecular graphs in three steps. First, MoFlow creates bonds (edges) using the glow based model. Secondly, atoms (nodes) are generated, using a recent conditional graph flow. Finally, MoFlow collects bonds and atoms into a chemically valid molecular graph. The proposed MoFlow scheme is represented in Figure 2 [21].

A molecular graph M (e.g. Metformin) includes a feature matrix A for the atoms and adjacency tensors B for the bonds. Inference: the conditional flow of graph $f_{\mathcal{A}|\mathcal{B}}$ for atoms converts A given B into conditional latent vector $Z_{A|B}$, and at the same time the glow $f_{\mathcal{B}}$ for bonds transform B into latent vector Z_B. The latent space shows a spherical Gaussian distribution. Generation: this process reverses previous operations, followed by a procedure of the validation correction that provides chemical confidence. Regression and optimization: the $y(Z)$ mapping between molecular properties and latent space optimizes molecular graph and predicts properties.

4 The importance of normalizing flows and oracles in graph generation

To evaluate the importance of the oracle and normalizing flows in graph generation, we compare different flow-based models (GraphNVP, GraphAF, MoFlow) and not flow-based models (MolGAN, GCPN) and their metrics with and without the oracle. Furthermore, MolGAN and GraphNVP models do not contain oracle. Three widely-used metrics were approved for method comparison, including validity, validity without (w/o) check and uniqueness. The first metric represents the percentage of chemically valid molecules for all generated molecules. The second metric, validity w/o check, shows validity of ablation models provided that not using validity correction or validity check, i.e., without the oracle. Finally, uniqueness specifies the percentage of unique valid molecules among the total number of generated molecules.

Table 1 shows that the percentage validity among all the represented above methods is more than the percentage validity without the oracle. This proves the importance of the oracle in the graph generation methods for both flow and non-flow methods. As for non-flow models, MolGAN achieves 100% validity, but only 2% uniqueness, indicating that the GAN is flawed due to the mode collapse [1]. For GCPN, the reliability is similarly reduced from 100% to 20% without the oracle. In contrast, flow-based models show better performance than non-flow methods, both for validity without the oracle metric and uniqueness metric. Specifically, the validity without the oracle metric of MoFlow and GraphAF is 4 and 3.5 times large respectively than the validity without the oracle metric of GCPN in Table 1. Therefore, the oracle increase in times the percentage of validity without check for flow-based models. Furthermore, each of flow models generate high unique molecules. In conclusion, the oracle helps to generate molecules with higher validity, and flow-based models show better metrics in molecular graph generation.

5 Molecular generation visualisation

Currently, the state-of-the-art MoFlow method shows the best metrics for generating molecules [21], therefore we decided to implement the property optimization experiment for this method. The property optimization practice directs at generating new molecules with the best Quantita-

Table 1. Comparison of different graph generation models with and without the oracle.

Method	% Validity	% Validity w/o check	% Uniqueness
MolGAN [2]	100	n/a	2
GCPN [19]	100	20	99.97
GraphNVP [10]	42.60	n/a	94.80
GraphAF [16]	100	68	99.1
MoFlow [21]	100	81.76	99.99

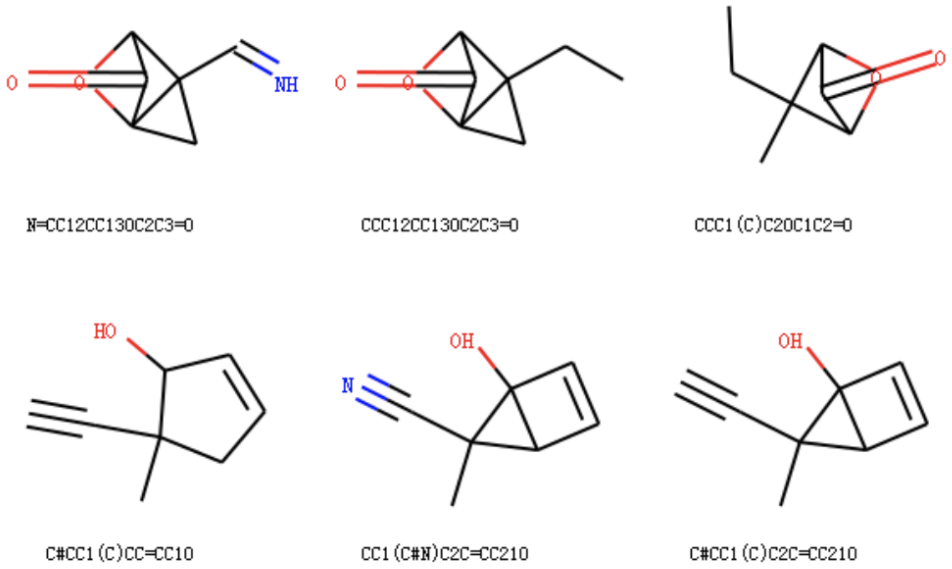


Figure 3. Molecular generation visualisation.

tive Estimate of Druglikeness (QED) metrics [13], [20]. QED determines the similarity to a drug for generated molecules. We use the pre-trained MoFlow method to generate molecules [3]. Figure 3 represents the generated result with the high QED metric.

6 Conclusion

In this paper, we have explored the importance of normalizing flows and the oracle in the molecular graph generating methods. First of all, we have considered the features of the structure of the oracle and normalizing flows. Furthermore, the principle of operation of state-of-the-art graph generation method has been considered in detail. To evaluate the importance of the oracle and normalizing flows in graph generation, we have compared three flow-based models and two not flow-based models and their metrics with and without the oracle. This paper presents that the

oracle contributes to generate molecules with higher validity, and flow-based models show better metrics in molecular graph generation. As a result, the property optimization experiment was realized to visualize the molecule generation.

References

[1] The rise of deep learning in drug discovery. *Drug Discovery Today*, 23(6):1241 – 1250, May 2018.

[2] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs, May 2018.

[3] Jérémy Besnard Sorel Muresan G. Richard Bickerton, Gaia V. Paolini and Andrew L. Hopkins1. Quantifying the chemical beauty of drugs, Jan 2012.

[4] Shion Honda, Shoi Shi, and Hiroki R. Ueda. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery, Nov 2019.

[5] Wengong Jin, Regina Barzilay, and Tommi S. Jaakkola. Junction tree variational autoencoder for molecular graph generation, 2018.

[6] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8):595–608, Aug 2016.

[7] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10215–10224. Jul 2018.

[8] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, Mar 2017.

[9] Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, Aug 2020.

[10] Kaushalya Madhawa, Katushiko Ishiguro, Kosuke Nakago, and Motoki Abe. Graphnvp: An invertible flow model for generating molecular graphs, May 2019.

[11] Maris Ozols, Martin Roetteler, and Jérémie Roland. Quantum rejection sampling. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference on - ITCS '12*, Mar 2011.

[12] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation, May 2018.

[13] Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties. May 2019.

[14] Kristof T. Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R. Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8(1), Jan 2017.

[15] Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focussed molecule libraries for drug discovery with recurrent neural networks, Jan 2017.

[16] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation, Jan 2020.

[17] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders, Feb 2018.

[18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, Oct 2017.

[19] Jiaxuan You, Bowen Liu, Rex Ying, Vijay S. Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation, Jun 2018.

[20] Jiaxuan You, Bowen Liu, Rex Ying, Vijay S. Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. Jun 2018.

[21] Chengxi Zang and Fei Wang. Moflow: An invertible flow model for generating molecular graphs, Aug 2020.

Games for elderly people

Youqie Li
Youqie.li@aalto.fi

Tutor: Sanna Suoranta

Abstract

This article introduces the psychology of the elderly at the beginning, then it explains the development of elderly games and the impact of e-games on the health of the elderly, this paper describes how electronic games improve the health of the elderly from the two parts: physiology and psychology. Finally, it summarises some design tips and principles of elderly games.

KEYWORDS: Game design, Elderly User Group, Cognitive Research, Human-Computer Interaction.

1 Introduction

Games have always been regarded as a way of entertainment for young people, and the main user groups of games are young people. In League of Legends, the world's most popular game, only 10% of players are over 40, people over 50 are less than 1%[20]. Research on the mobile game usually ignores the elderly people, and there are few studies on elderly game players. At the same time, we are facing an ageing society. People aged 60 and above had reached 185 million, 13.7 per cent of the total population in China in 2021. The situation is the same in both Europe and

America. In 2021, the elderly population will reach 248 million, 17.17% of the total population[14].With the aggravation of the ageing rate, society is faced with tremendous pressure of providing for the aged. The advent of ageing society promotes the development of the "silver hair market“. People need to accelerate the pace to adapt to a greying society and meet the needs of the elderly. As a result, it is necessary to study game design for the elderly.With age growing, people's cognitive abilities, reaction time, hearing and touch are declining when compared with younger people. Therefore, games for elderly people need a unique design. On the other hand, games may also help older adults maintain their mental health; by playing games, the elderly can improve their cognitive ability and learning ability[11]. This paper reviews the latest approaches used for gaming design and interaction for elderly people.This paper is organised as follows. Section 2 introduces previous research on the game for elderly people. Section 3 introduces the characteristics of elderly people, and Section 4 discussed some games specially designed for elderly people. Section 5 describes why older adults need to play games. Section 6 discusses some interaction methods that can be used for game design. Finally, Section 7 is the conclusion for the whole paper.

2 Existing game research for elderly

There have been many years of research history on computer game design for the elderly. In the 1980s, Weisman figured out the importance of adaptability by using Apple II games among older adults[15]. Generally, the research on older adults mainly focuses on game interaction and mental health while playing games. Sanches Lam used some experiments to confirm that the elderly can play smartphones games to fulfill their learning needs [8].Wijnand Ijsselsteijn attempted to find out the benefit that older adults get by playing Nintendo Wii[19].Jan-Henk Annema have proven that video games can help to motivate patients, develop skills and serve as a distractor in pain management[6]. Minzhu Jin proposed the design norms of games for the elderly, and summarised the following interaction norms[12]: “Multi-channel interaction optimisation for information transfer. The flat design promotes logical guidance. The consistency principle lightens the burden of memory. The natural interface improves mental model matching. Simplify gestures to improve performance.” Based on these norms, she has designed a new chess game

specifically for the elderly. To sum up, at present, Research on games has focused on its benefits to older adults, such as its positive physical or mental effects on old people. However, there is relatively little research on the games themselves. What kind of games do the elderly like and what changes can smartphones bring to the games of the elderly? Based on the characteristics of elderly people, what kind of unique design do we need to do?

3 Characteristics of elderly people

3.1 Physiological decline

Games usually start the interaction with users through text, audio, video and other display methods. With the growth of age, users' visual and auditory abilities decline, which directly leads to the loss of perception function.

Visual System: Because of the age and insufficient processing of screen brightness and small text characters[7]; The lens of the eye loses its elasticity and becomes thickened and yellowish, resulting in presbyopia which cannot see patterns and textures. With the growth of age, the static and dynamic visual acuity of the elderly will decline and contrast, colour sensitivity will decline. On the other hand, their ability to adapt to the dark will weaken, and the sensitivity to glare will increase[10]. Such visual impairments may make it harder for older people to perceive small elements on a monitor (for example, small details in a game map)[19], read tiny instructions or locate information on a complicated screen.

Auditory System: Hearing loss, which requires a higher decibel level to achieve the same hearing as in younger people; High-frequency sound can cause severe hearing impairment in the elderly. It is more challenging to maintain listening attention in noisy environments than young people. Older people may find it challenging to understand synthetic speech because it tends to be somewhat distorted. For non-verbal audio signals, older adults are more likely to hear lower frequencies (in the 500-1000 Hz range) than higher tones[23].

3.2 Psychological change

Thinking ability: The decline of older adults' thinking ability mainly reflected on the degeneration of generality, logical thinking and creative thinking ability. The decline of general thinking is reflected in the limitation of the elderly's inductive cognition of games. When playing, it is difficult for the elderly to summarise the functions and attributes of game elements and apply them to other interfaces by comparing and summarising[22]. The decline in logical thinking is reflected in difficulty in understanding and deriving the hierarchical relationship of games. Elderly users tend to hold tentative, timid intentions when the operation is not smooth; once the operation fails, the elderly will reduce the enthusiasm of exploration. The decline of creative thinking is reflected that the elderly tend to be more familiar with conventional thinking, and it is difficult to reorganise the new thinking mode of game with new knowledge and experience.

Memorising ability:According to the memory system model, the memory of information can be divided into three degrees. The first one is sensory memory, where information enters the memory system through the sensory organs in different ways. The second one is working memory, the initial perception of information and the conscious processing of information stored in the brain's memory. The third one is long-term memory, which is made up of what was learned. Short-term memory is used for information processing, and the short-term memory capacity of the elderly tends to decline with the growth of age. The short-term memory capacity of the elderly is generally 5-9 units, while the memory capacity of the elderly over 60 years old is only 3-7 units[1].

Risk aversion:It is not easy to take risks for elderly people. With the increase of age, the memory and learning ability of the elderly deteriorate. At the same time, decision-making relies on emotional and cognitive processing[17], so the degeneration of cognitive neural of the elderly will also have a particular impact on risk decision-making. When cognitive function declines, the elderly are more inclined to use simple decision-making strategies with less information processing and easier integration. As a result, older people are risk-averse, so they rarely try riskier things, and tend to be conservative[21]. Most older people prefer to go to the park rather than to travel over mountains and rivers. In the psychological experiment about expectations[3], the elderly group has a stronger

tendency of irrational decision making as opposed to the young group. The elderly are more willing to choose the low-risk stable return compared with the high-risk and unstable return.

Anaclisis and Nostalgia:Old people usually have strong dependent psychology(Anaclisis), because as cognitive ability decline, the elderly are unable to understand the new information very well. The frustration will make older adults tend to seek for help, that is why dependent psychology generates, especially on electronic products or video games for the elderly, the advanced electronic products are very unfamiliar for them[5]. Because of the fear of failure, they do not dare to experience and try these new things. Elderly people will hope that someone could teach them before using. So in the game design, we can design the teaching part more detailed.

Older people tend to be nostalgic and have a strong sense of nostalgia for places where they lived or people they loved Games for the elderly should focus more on the past rather than the future. For the nostalgia complex of the elderly, when designing games, attention should be paid to the emotional changes of the elderly.

4 Elderly game situation

In the past, the public believed that the elderly's preference for digital games might be traditional chess and card games. However, with the deepening of research, people began to realise that the elderly also have various understandings of games. Bob De Schutter made a questionnaire among 239 individuals, and 124 people completed the questionnaire[2].Bob De Schutter divided these people into three groups: "young " (33.1%, age ,45–54), "old " (50.8%, age, 55–64), and "senior" (16.1%, age, 65 and above). Over one-third of the people(34.9%) received the highest education, less than one-third people (31.5%) were trained at the middle level, while only 27.4% had primary education. In the survey, the elderly people play quite limited games in their daily life, The main types of games for the elderly are puzzle games and traditional chess and card games, which spread widely among the elderly. More than 80 per cent of elderly people played only those two types of games, while the remaining 20 per cent played a variety of games. For the remaining 20 per cent of older people, they played first-person shooters (FPS) games like Half-life;Role play games(RPG) like Devil May Cry 5; Simulation games like Minecraft;

Car racing games like Need For Speed; Sports games like FIFA. For older core gamers, Call of Duty is the most frequently mentioned game. For casual games, Tetris, Spider Solitaire and Zuma are the most mentioned games. In terms of the choice of game platforms, more than 80% of the elderly play games on desktop. Mobile phones, consoles are relatively unpopular platforms. Moreover, the ARPU(average revenue per user) of the elderly players is generally not high; over, over 76% of the players never spending money on any games. In terms of playing time, the average time of the elderly was 1.45 hours per day. Heavy gamers (16.1%) averaged 2.5 hours in the gaming playing, moderate gamers (39.5%) averaged 1 hour and 25 minutes, and light gamers (44.4%) averaged less than an hour in gaming. Bob De Schutter's survey was done in 2011, nearly ten years passed, and we have entered the mobile era, with the development of smartphones, mobile games also get rapid growth. There is a noticeable change on the game platform, in the past computers were the primary way of playing games, but now it gradually turns to the double platform: mobile phones and computers. Because of its good portability and convenience, more and more old people start playing mobile games.In a Chinese survey[9], more than 60% of the elderly people play mobile phone games every week. Angry birds(Casual game), PUBG Mobile(Battle game), Honor of King(MOBA) and candy crush(Casual game), those four games are their favourite games. The traditional chess and card game market is gradually shrinking; the elderly are gradually changing their game interest to young people.

5 What do the elderly get from playing e-game?

Bob De Schutter divided the old players into six groups based on their motivation[2],In tabel 1 The most popular motive to play digital games is the challenge, and the least popular motive is Social Interaction. From the research above, we can get the conclusion that most old people play games to get a challenge.

5.1 Physiological demands

With the growth of age, the human body will have specific changes, such as the decline of senses, memory, and other problems. Therefore, the games should make up for the physiological problems of the elderly and

Motivation	Explanation
Competition	To be the best in games
Challenge	To beat the game or to get to the next highest level
Social Interaction	To play as a social experience with friends
Diversion	To pass the time or to alleviate boredom
Fantasy	To do things that cannot do in real life
Arousal	To play because the game is exciting

Table 1. Game Motivation For Elderly People

help elderly people keep cognitive abilities. Studies have shown that games have a specific therapeutic effect on mild cognitive impairment[21]. Researchers gave ten elderly people who scored 16-24 in the MINI-mental State Examination (MMSE) a 10-week game training. The MMSE is a 30 points questionnaire to measure cognitive impairment and 16- 24 points means a mild cognitive impairment.The game included chess and card game, word games, action games, casual games and memory games. The average MMSE score increased from 21 points to 23.2 points, proving that games can indeed improve the cognitive level of the elderly. Attention and memory were significantly improved. West GL asked elderly people to play the super Mario 64 game 2 hours per day and three times a week. After 90 hours of playing, the hippocampus grey matter increased compared to control group[18]. The hippocampus is widely credited with memory and direction positioning function is also the damaged area in Alzheimer's disease. Although researchers emphasise the result does not prove that super Mario 64 can help keep the hippocampus grey matter. A growing body of research shows that the fair game would be helpful to improve the cognitive ability,in an experiment,the Portal2(an FPS and decryption game) improves players in basic cognitive test scores[16].

5.2 Psychological demands

The psychology of the elderly will be more sensitive than young people. Designers should actively integrate The elderly into modern technology through some interactive means. First, the elderly need to accompany. In modern society, there are more and more empty-nesters who are eager to communicate and want to be noticed. Therefore, there are a lot of remote interactive apps which can let their offspring know the living conditions of the elderly in time and provide a good communication platform for the

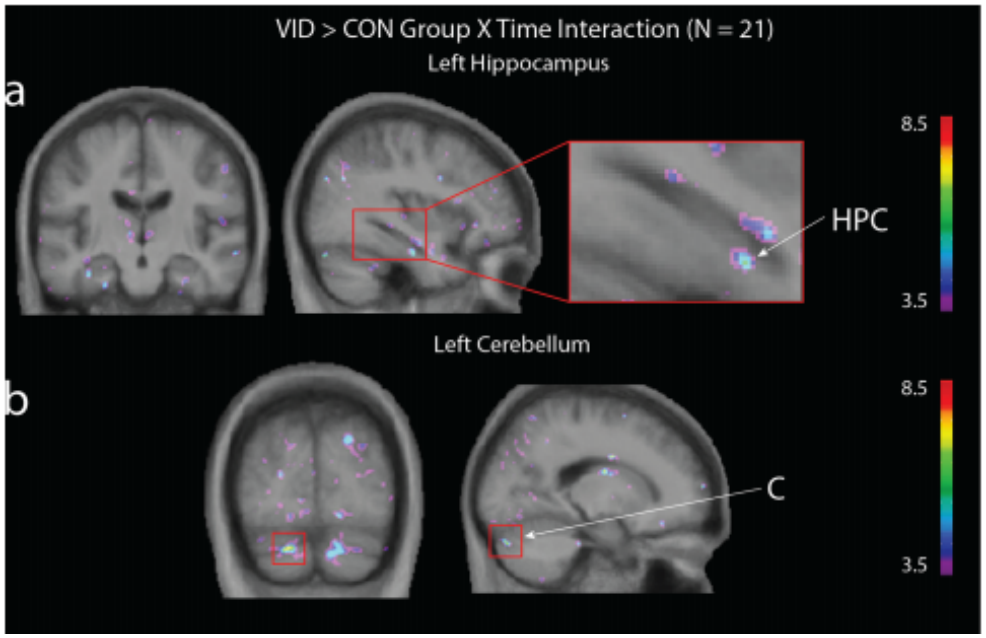


Figure 1. Brain CT of Hippocampus

elderly and their offspring. Second, the elderly desire to live a longer and healthier life, and they begin to pay attention to health care, so more and more medical apps come into people's sight. Finally, the elderly have needed to realise their self-worth, which is the highest level of Massno's demand theory. The elderly need more affirmation of their self-worth to guarantee their self-confidence. The excellent product is designed to fulfil people's demands. The elderly have much time after retirement. As a result, the real demands of the elderly must be taken into account when designing an APP.

6 Interaction Suggestion

6.1 Elderly people-centred design

User-centred design is to focus on users themselves,user is the centre. It breaks through the traditional mode that function determines the form, but by using the user's psychological feelings and behaviour to determine the product performance. The elderly-centred design concept focuses on the needs and goals of the elderly and solves the problem of the application environment for the elderly. At this time, the elderly are not only the objects to be designed, they can also guide the design, giving their advice in the game design process. The task for designers is to transform the

needs into design language and try to design a more suitable game for the elderly. In the design of smartphone games, they should first identify the needs of the elderly, locate the goals of games, analyse competing products in the market. Secondly, they should study the needs of the elderly, including their physiological characteristics, psychological characteristics and behavioural habits. Specific research methods include qualitative research (observation method, user interview method, focus group.) and quantitative research (questionnaire survey method). The functional and content needs of games are defined according to the research results. And then they should design the elderly game interaction model, and build a mobile game prototype according to the previous research. This part including low-fidelity prototypes and high-fidelity prototypes, after that they need to test the target group to evaluate the game, whether it meets the needs of the users, whether the task flow is reasonable, whether there are still some needs have not been fulfilled; After the usability testing, they can do some small changes and release the game; After the product launch, designers can collect feedback from the elderly players, maintain and improve the games in the next version.

6.2 Usability Principle

As a universal principle, the usability principle has been widely used in Internet products. It is a necessary standard to test whether a product is successful or not. In the game design of the elderly, we still need to follow this design principle. Nielsen, a well-known usability expert, has come up with a definition of usability that he believes should include the following ten things[13]:

1. Visibility of system status.Keep the status of the interface visible; every change should be visible; any content should also be visible.

2. Matching ideal system and the real: Use the user's language, rather than use system language that are hard to understand.

3. User control and redo function. For stand-alone games, designers can set the pause button and archive buttons.

4. Consistency and standards:The operation of the game should be consistent with other mainstream games.

5. Error prevention principle: Remind the elderly people when they may make mistakes, such as the possible result of deleting the game archive.

6. Recognition: Actions and options should be visible. The instructions for the system should be clear and visible, such as giving older people more game cues during the game to help them remember the game play rules.

7. Flexibility and efficiency

8. Aesthetic and minimalist design .Getting older people to focus more on the game itself

9. Help users recognise from errors: When the game cannot open due to network failure, Users can be told some possible solutions.

10. Help and documentation: It is necessary to provide help and documentation.

6.3 Emotional Design

When it comes to the emotionalisation principle of old people, Donald A.Norman put the emotional design into three layers: viscera layers, behaviour layer and reflection layer, viscera layer mainly focused on the product appearance design, behaviour layer mainly focused on the product interaction, reflection layer the combine of the first two levels, and cause more in-depth emotional experience. For elderly emotional expression in the game, we mainly concentrated on the interface of the game; Norman divided emotional factors into five aspects: the connotation, graphical interface, interface colour, interface layout and interface text[4].

1. Graphical factors of emotion in the game interface. In terms of game graphic design, the elderly are a particular group. As they are not as experienced as the young in using mobile apps, they will inevitably encounter some problems. They cannot understand something strange, so designers need to satisfy the mental model of the elderly in graphic design. Due to the general loss of vision in the elderly, graphic design should take this into account. In the interface design, designers should put more effort into solving these problems.

2. Emotional colour factors in the game interface of the elderly. In the design of the game interface, the use of colour need be mentioned. Colour plays a vital role in interface design. Colour can set the emotional tone for the whole game. Meanwhile, it can distinguish information modules, suggest functions and highlight essential information. The elderly have dropped to the resolution of the colour, and their vision decreased, and the study found that older people can distinguish the induction of lightness colour best because the colour looks more relaxed and comfortable,
3. Text factors of emotion in the game interface of the elderly. Font and size of text are two main factors influence the experience of the elderly. At present, the most commonly used fonts on mobile phones are sans serif fonts, and the right font equals a mature design. A good font is easy for users to read, but it also conveys what the designer wants to highlight. On the contrary, if the font is not appropriately chosen, it will be difficult for gamers to read, then their inner emotions will turn into negative emotions, and the whole game will be a failure. At a visual distance of 50cm, the minimum text size that the elderly can see easily is 15PT.
4 The layout of the emotional factors. The life of older adults is simple; the game in the aspect of layout should also meet the demand of the elderly simple. In the interface design of the game, we can try to delete the interface which has many the chaos elements, at the same time we can hide the non-important function to cause the user interference to a minimum level, in order to give players a clean game experience. After removing unwanted elements, we can re-layout the interface. The location of each element is arranged primarily through hierarchy, alignment, grouping, and spacing.
5. Connotation factors of emotion in-game interface for the elderly. The design with connotation is good. The involved concept is visualised and materialised using metaphor and symbol. The connotation factors of game interface mainly include symbol, metaphor and story.

7 Conclusion

This paper mainly analyses the design elements and methods of phone games for elderly people, what are the psychological and physiological characteristics of the elderly, the current situation of the elderly games and the elderly's game needs. Finally, it gives some Suggestions for the elderly game design. For the elderly, There are significant differences compared with young people in-game habits and goals, So, for game design, game, goals and rules of the game will be made to fulfil the needs of

elderly, and due to the physical and psychological conditions of the elderly, we need to simplify the game interaction as easy as possible, in order to suit the operation habit of the elderly. Among these design elements of games for the elderly, usability, emotionalisation and elderly-centred design are needed.

References

[1] Miller G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 2(81–97), 1956.

[2] De Schutter B. Never too old to play: The appeal of digital games to an older audience. *Games and Culture: A Journal of Interactive Media*, 6(155–170.), 2011.

[3] Robbins T. Deakin J, Aitken M. Risk-taking during decision-making in normal volunteers changes with age. *Journal of the International Neuropsychological Society*, 10(590-598), 2004.

[4] Donald.Norman. *Emotional design*. Beijing: Electronic Industry Press, 4th edition, 2004.

[5] Schieber F. Human factors and ageing: Identifying and compensating for age-related deficits in sensory and cognitive function, 2018.

[6] Stef Desmet. Jan-Henk Annema, Vero Vanden Abeele. Video games in therapy: a therapist's perspective. *Proceedings of the 3rd International Conference on Fun and Games*, 10(94–98), 2010.

[7] Lennart Nacke. Kathrin Gerling, Ian Livingston. Full-body motion-based game interaction for older adults. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 12(1873–1882), 2012.

[8] Shung W. Lam S. Understanding the need for mobile ict learning as an elderly learning tool. *International Journal of Emerging Technologies in Learning*, 4(35-40), 2020.

[9] Zongwei Lv. Research on the game design of urban elderly smartphones. Master's thesis, East China University of Science and Technology, June 2018.

[10] D. Maughan. Eyesight in the elderly. *https://www.physio-pedia.com /Eyesight*, 2020.

[11] Vero Vanden Abeele. Max V. Birk, Greg Wadley. Video games for mental health. *Interactions*, 4(32–36), 2019.

[12] Haibo Wang. Mingzhu Ding. Research on interactive design strategy of games based on cognitive characteristics of elderly. *Users.Design and Art*, 2(99-101), 2020.

[13] Jakob Nielsen. Usability heuristics for user interface design, 2018.

[14] National Bureau of Statistics. Statistical communiqué. *http://www.stats.gov.cn /tjsj/zxfb/201902/t201902281651265.html.*, 2018.

[15] Weisman S. Computer games for the frail elderly. *Gerontologist 23*, 4(361-363.), 1983.

[16] Ventura M. Shute V. The power of play: The effects of portal 2 and lumosity on cognitive and noncognitive skills. *Comput. Educ.*, 80(58-67), 2015.

[17] Li Tao. Research on the influence of emotion induction on age difference of risk selection preference. Master's thesis, Southwest University, June 2012.

[18] Konishi K West GL, Zendel BR. Playing super mario 64 increases hippocampal grey matter in older adults. *PLoS ONE 12*, 12(12), 2017.

[19] Yvonne de Kort Wijnand Ijsselsteijn, Henk Herman Nap. Digital game design for elderly users. *In Proceedings of the 2007 conference on Future Play*, 5(17-22.), 2007.

[20] Wukong. Age distribution of league of legends players. *https://www.wukong.com/question/6608044797775053069/*, 2020.

[21] Li Fadi Xia Weihai, Yin Nan. Comparison of risk decision-making between the elderly and the young. *Chinese Journal of Gerontology*, 22(5680-5683), 2017.

[22] MAO Xiaoou. User research and design of goal-oriented for elderly it products. Master's thesis, Dalian Maritime University, June 2007.

[23] Gong Xuechen Zhang Honglei. Nvestigation of hearing loss and speech recognition ability in the elderly and risk factor analysis. *Chinese Journal of Otolaryngology-Head and Neck Surgery*, 2(116-120), 2019.

Distributed Learning on the Edge

Alvar Wihuri
alvar.wihuri@aalto.fi

Tutor: Thaha Mohammed

Abstract

This study performs a survey on methods for training machine learning models at the network edge. The goal is to form a picture of the problems posed by edge learning when compared to cloud computing and distributed learning, and the current technologies that attempt to solve these problems. The study is performed by studying current literature, using a set of existing surveys as a guideline. The problems posed by edge learning are skewed and varying amounts of training data, communication overhead, heterogenous device capability and privacy. A set of solutions is identified to comprise federated learning (FL), accumulating or compressing sent gradients, varying the communication topology, relaxing the synchronization requirement of FL, using lower numeric precision in the network and in communication, and a secure aggregation protocol and a differential privacy mechanism. As a conclusion we note that a wide range of solutions exist that may be applicable to edge learning, many of them being improvements over FL. However, the applicability of some solutions to the edge context is left ambiguous since the solutions are studied in the context of more general distributed training.

KEYWORDS: machine learning, deep learning, edge computing, distributed learning

1 Introduction

The growth of the Internet of things (IoT) paradigm [21] has created many applications for Internet-connected smart devices, such as autonomous vehicles, smart cities and smart healthcare. In turn, this trend has brought along the need for moving computation away from the cloud closer to the devices on the network edge, a paradigm called edge computing [24].

The field of machine learning (ML), and deep learning (DL) in particular, has shown great progress in a wide range of tasks from image classification to text analysis, accelerated by the IoT trend and the increasing amount of data that IoT devices are generating [12]. DL is an active research area in both academia and the industry, as well as a highly useful technology in a wide range of applications.

However, while Internet-connected smart devices are an important platform for DL applications, the edge network paradigm poses some novel challenges for DL computation. Issues that need to be resolved include communication overhead and privacy [28]. Different approaches are needed for both learning and inference, two key components that comprise deep learning computation.

This paper examines and categorizes the existing approaches for performing the learning phase of a DL model on the network edge. Some earlier surveys exist on the topic [28, 26, 25, 18], but the taxonomies are created from different perspectives, and thus the relation of different approaches may remain ambiguous. The goal of this study is to identify a set of problems that differentiate learning on the edge from different contexts, and to analyze which of the problems are alleviated by each of the examined solutions.

This paper is organized as follows. Section 2 examines the background technologies of both deep learning and edge computing. Section 3 provides an overview of the main ideas of performing DL computation in a distributed manner. Section 4 examines and categorizes the main approaches to distributed learning in the edge computing context. Section 5 analyzes the approaches. Finally, section 6 presents a conclusion.

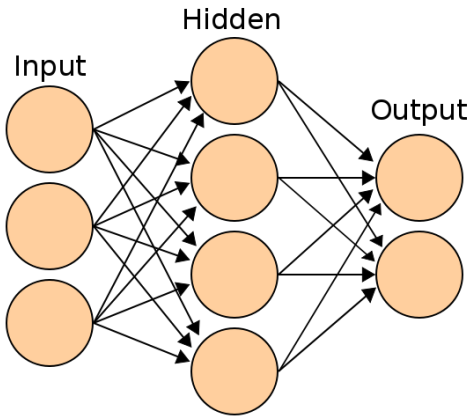


Figure 1. An example of a simple neural network. en:User:Cburnett, CC BY-SA 3.0, via Wikimedia Commons

2 Background technologies

Training ML models on the edge is achieved as a combination of two distinct technologies: machine learning and edge computing. Thus, an overview is provided on these technologies. In machine learning, a large amount of attention is focused on the branch of deep learning, which is focused on here.

2.1 Deep learning

Deep learning refers to the approach of using deep neural networks (DNN) for solving ML problems. While the widespread interest in DNNs has been a recent development, research related to neural networks has a longer history. For instance, the perceptron, a fundamental building block of neural networks, was developed in the 1950s [22]. More recently, AlexNet [16], a convolutional neural network, started the recent trend of widespread attention and rapid improvements in deep learning.

At their core, neural networks consists of several layers of interconnected nodes, or neurons, with adjustable weights. The first layer forms the inputs of the network, the last layer forms the outputs, and in between, a number of hidden layers provide a complex set of connections that map from the inputs towards the outputs. These layers have adjustable weights that can be adjusted algorithmically to yield desired functionality for the network.

Depending on the application area, different types of neural networks may be used. A simple architecture, the multilayer perceptron is comprised of a set of fully connected layers, where the value of a node is com-

puted as a linear combination of its inputs, mapped through some activation function. An example of a fully connected neural network is shown in Figure 1.

Some architectures incorporate different types of layers. A convolutional neural network (CNN) [16] includes convolutional layers that can recognize local features from e.g. image based inputs, and pooling layers, that reduce the amount of data to compute in the network. Recurrent networks compute data in a sequential manner, using recurrent units such as the long short-term memory (LSTM) [13] or the gated recurrent unit (GRU) [7] to control how to handle previous parts of a sequence in relation to latter parts. These networks are particularly suitable to language processing tasks, such as translation.

Some additional network architectures are achieved by different configurations of networks, such as autoencoders [3] which try to compress and codify given information, and generative adversarial networks (GAN) [11], which are a two-part generative model where a second network attempts to identify samples that came from a generative network instead of the training data.

Typically, the weights of the network are initialized arbitrarily. Then, in a training stage, inputs are mapped to outputs, and using some loss function and backpropagation, the weights are adjusted towards producing more correct outputs using their gradients.

The approach of using gradients to adjust weights is referred to as gradient descent. Typically, some variant of gradient descent is used, such as mini-batch stochastic gradient descent, that computes gradients only based on a small set of samples, called a mini-batch.

After learning, the network can be used for inference, where the output values of the network are interpreted as the desired result, such as a classification of an object to a set of given categories. Thus, DL computation comprises two distinct stages: learning, and inference.

2.2 Edge computing

Edge computing (EC) is defined as a model of computation where computing is performed in network-connected devices close to the edge of the network [24]. The edge is viewed in contrast to the cloud computing paradigm, where computation takes place in large, centralized servers. Edge networks are comprised of a wide variety of devices, comprising data-producing IoT devices, data-consuming end-user devices, and servers

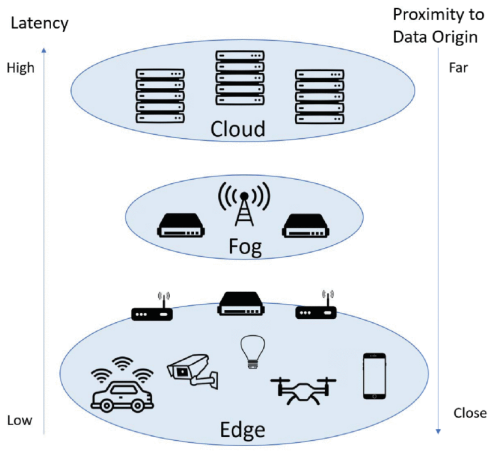


Figure 2. An illustration of the edge network, according to Caprolu et al. [5], © 2019 IEEE

providing other functionalities [5].

Edge computing is closely related to a similar term: fog computing (FC). FC describes a more network-heavy approach, where the fog may be viewed as a variant of the edge but where computation is performed more in the network itself [9]. However, in some cases, FC may be viewed as a synonym for EC [9]. A typical interpretation of the cloud, fog and edge is presented in Figure 2.

Edge computation performs computation closer to the devices in an attempt to increase efficiency from the point of view of the network-connected devices. More precisely, a shift towards edge computation is motivated by issues over latency, bandwidth, and privacy [24]. The reduced latency would be useful for real-time applications, such as autonomous vehicles. Bandwidth becomes an issue, since IoT devices produce ever larger amounts of data, making transmitting the data over the network impractical. Privacy is a concern, since IoT devices may generate private data, e.g. health monitoring, which would be preferable not to be sent over the network to the cloud.

Some key challenges of edge computing are related to communication and distribution of computing: network communication needs to be efficient enough to not produce new bottlenecks and allow for smart vehicles to quickly move around the network, and the computation needs to be orchestrated such that neither the edge devices nor edge servers get overloaded [9].

3 Distributed learning

This section examines distributed learning in general, without considering the practical issues that might arise at the network edge. First, the difference is explained between data and model parallelism. Then, the algorithms of distributed stochastic gradient descent and federated learning are explained.

3.1 Data and model parallelism

The parallelization in distributed learning can be roughly divided into two types: data parallelism and model parallelism [26, p. 9]. In data parallelism, several copies of a DNN are made, which are trained on different data in parallel. On the other hand, in model parallelism, a model is partitioned into several parts, which only handle their own communication and communicate the data coming to/from other parts of the model. Even a combination of these types of parallelization could be used in model training.

Currently, the main type of parallelism in training on the edge is data parallelism, whereas combining the two approaches is seen as an open challenge [26, pp. 27–28].

3.2 Distributed stochastic gradient descent

Since learning algorithms are based on gradient descent, implementing distributed training can be viewed as a problem in how to implement gradient descent in a distributed context. SGD by itself approximates gradient descent by replacing the gradient over the entire training set with the gradient of a single training sample. In practise, SGD can be modified to be a hybrid that computes the gradient over a mini-batch, a small subset of the training samples.

A distributed, asynchronous implementation of SGD is presented in [8]. The approach uses a combination of data and model parallelism, where the former aspect can be interpreted as a modification to SGD. The algorithm consists of a central parameter server and multiple workers. The parameter server holds the model weights, and is able to update the weights based on gradients received from a worker. The workers asynchronously fetch up-to-date weights from the parameter server, then compute new gradients based on a mini-batch, then asynchronously send the

gradients back to the parameter server. That is, the workers are not synchronized, but they fetch new weights and send back gradients without knowledge of the other workers.

Distributed SGD can also be implemented synchronously [6]. In a synchronous implementation, the central server coordinates the computation of individual workers. First, the server sends new weights to all workers. Then, the workers compute gradients based on mini-batches and send the gradients to the server. The server waits until it has received gradients from all workers. Then, it updates new weights based on all of the received gradients, and sends the updated weights to all workers for a new round of computation.

The advantage of a synchronous approach is that workers run on up-to-date weights, in contrast to an asynchronous approach where the weights on a central server are likely to have changed between a worker fetching new weights and sending back its updates [6]. However, a synchronized iteration must wait for the slowest worker to finish its computation. Therefore, modifications are required for a synchronous approach to be practical [6].

However, these data-center based approaches face problems when used in a less controlled setting, such as on edge devices connected over a network. These problems can be alleviated by the use of federated learning.

3.3 Federated learning

Federated learning (FL) [20] is a distributed synchronous variant of SGD where only a subset of available devices are used for a single round of computation. It is based on an algorithm called FederatedAveraging [20], and it is run on a central parameter server and a large number of participating devices. It is controlled by three parameters, producing different variants of the algorithm: the number of devices selected each round, the number of training passes a client performs over its local data, and the mini-batch size.

FederatedAveraging [20] works as follows. First, a central parameter server selects a subset of clients and sends to them the new weights. Then, the clients locally run iterations of SGD, i.e. instead of computing only gradients, the clients have local weights that are updated with gradients. The clients iterate over their local data a given number of times, with some mini-batch size, updating local weights on each mini-batch. Once the clients have completed their iterations, they send the updated local

weights back to the parameter server, where the global weights are updated as a weighted sum of the client updates, adjusting for the variable amount of training data per client.

4 Distributed learning on the edge

Performing various AI related tasks can be classified into a wide hierarchy of categories depending on which parts of the computation are performed on the edge and which parts are performed in the cloud [28]. Here we consider the training of ML models on the edge, thus, we require most of the training-related computation to be performed on edge devices. However, storing model data and combining results of individual devices may still be done in the cloud on a centralized parameter server.

In practice, surveys would indicate that performing learning on the edge is often studied in the context of FL [26, 18]. Thus, many related technologies are modifications or improvements of FL. However, while other studies focus on distributed learning more generally or leave their context ambiguous, their techniques may still be applicable in an edge learning context.

While learning on the edge alleviates communication and privacy issues when compared to cloud training [26, p. 18], additional challenges still remain in edge learning, some of which are not present in a distributed data center context. In particular, communication still creates high network load [25]. The devices used for training may have varying amounts and different types of data [20]. Additionally, some privacy concerns remain, and the wide distribution of device capability needs to be addressed [18].

This section gives an overview of some of the technologies that focus on these issues in distributed learning, in particular at the network edge.

4.1 Federated learning on the edge

Federated learning by itself alleviates several problems that arise in a real-world distributed context [20]. The main issue that FL solves is non-independent and identically distributed (non-i.i.d.) data. In practice, this means different devices can be skewed towards certain types of training data. For example, if smart phone messages are used for training, the data are likely to be comprised of different languages depending on geographical location of the device. McMahan et al. find that federated

learning is robust against non-i.i.d. data.

FL is suitable for a setting with a large number of devices, since only a subset is used at a given time. The selection process may also circumvent the issue of device availability if only network-connected devices can be chosen for a training round. FL improves privacy, when compared to cloud training, since the training data is not sent outside the devices, e.g. to the cloud.

4.2 Communication improvements

If model updates are communicated with a central server, such as in FL or some other distributed SGD variants, the sent data can be reduced to reduce network load.

One method to reduce the bandwidth use is to send only important updates from the clients to the central server. Lin et al. [19] implement a scheme where only sufficiently large gradient updates are sent, and smaller updates are accumulated on device and only sent once the local gradients have accumulated to a sufficient size.

Communication overhead can also be reduced using lossy compression. For example, Caldas et al. [4] use lossy compression for server-to-client communication, and in conjunction with other optimizations, achieve large reductions in communication costs.

A key observation by Lin et al. [19] is that a large number of conventional sent updates would be redundant, indicating that without specific consideration to network bandwidth, edge training could have an unnecessarily large bandwidth use.

4.3 Alternate communication topologies

In many SGD variants, including FL, the used algorithms assume the existence of a central server, which holds the current parameters and controls their updates. However, distributed communication schemes have been implemented which do not require a central server.

Gossip algorithms are algorithms where individual nodes communicate only with their neighbors, possibly chosen randomly. A set of these algorithms has been developed to compute aggregate operations in a decentralized manner, and Šajina et al. [23] show that this idea can be suitably combined with distributed SGD.

The algorithm in [23] works as follows. First, a step of learning is per-

formed. Then, nodes are able to send their weights to one other node. A node receiving weights can decide whether it uses the new weights. If the received weights are similar enough to the node's own weights, the weights are combined. The combination is weighted based on loss computations.

The communication can be extended further in an approach where all nodes communicate with each other. This is shown to be efficient in a data center context [15], and avoids problems of all nodes reaching consensus. However, the communication overhead may be too large for network edge communication.

4.4 Synchronization of model updates

Central server based models may be modified in how the updates are sent to the server. FL requires these updates to be synchronized. However, in an edge network, devices are heterogenous in terms of their processing power, and as a result, some devices may take much longer for training iterations than others. As discussed in section 3, distributed SGD can be implemented asynchronously. Similarly, an asynchronous variant for FL has been developed [27], where workers can both start training and send results to the parameter server at any time. This is more efficient in a case where some workers are slow to respond with their computations, since the algorithm may proceed without having to drop devices from the computation.

4.5 Low precision networks

Networks and their related data may be encoded with a low amount of numeric precision, while still allowing training and inference with a reasonable amount of accuracy [14]. This would make the learning more suitable for mobile devices with low processing power and battery life constraints. Li et al. [17] show that low precision gradients can be used for distributed training.

4.6 Privacy improvements

Due to the edge computing paradigm, real user data may be used for training, raising privacy and security concerns. Since FL performs training locally, it does not require sending training data to the network, thus making plain FL more secure than performing training in the cloud. However,

FL still sends the model weight updates to the server. To mediate the risk of an adversary being able to infer sensitive information, a secure aggregation protocol may be used [2] such that only the sum of several updates is sent to a server at once.

Another privacy concern is that based on the final model and possible additional knowledge, information of the training data could be inferred backwards. Differential privacy based mechanisms have been developed for both plain SGD [1] and in a FL context [10] to mitigate these threats.

5 Analysis

A key problem in learning on the edge is non i.i.d. data [20], against which FL is robust. Thus, if other approaches are to be used on the network edge, their performance, particularly training convergence, needs to be verified on skewed training data.

Devices having varying amounts of data is similarly a problem against which FL works well in practice [20]. If the varying amount of data causes some devices to be significantly slower than others, an asynchronous variant [27] might be preferable.

Another problem on the network edge is communication overhead. A large body of research focuses on this problem from a wide range of different angles. Since communication overhead is a problem in distributed learning in general, not all research explicitly considers the FL angle; however, the proposed techniques, such as only sending sufficiently large updates [19], or applying lossy compression [4], would seem generalizable to different contexts. Varying the communication topology may be experimented with, but the applicability remains ambiguous for FL or other approaches at the network edge. Low precision gradients can also be used in distributed training [17], which could entail sending less data over the network, but again the edge perspective needs to be studied. An asynchronous variant of FL [27] may reduce load if updates are more distributed over time.

The wide range of device capabilities might be alleviated by using an approach where a central parameter server, if used, needs not to wait for all updates to proceed. For example, an asynchronous variant of FL may be used. Alternatively, using a network with lower numeric precision [14] may alleviate device load, which could make training more accessible for a wide range of participating devices.

Privacy is handled effectively in plain FL when compared to cloud training, since the training data can stay on the participating devices. Some threats remain, e.g. the possibility to infer information about the training data based on weight updates, which can be mitigated by aggregating the weights in a certain way [2]. Additionally, the effects of different communication topologies, such as [23], could be further investigated from the privacy perspective.

6 Conclusion

This paper has conducted a survey on distributed learning on the edge network. Training at the network edge is separated from more generic distributed training by a number of practical concerns, such as communication overhead. We find that a wide variety of research addresses these novel problems. In particular, many of the proposed solutions are modifications of FL. However, some techniques, such as compressing sent data, may be applicable in a wider range of scenarios. More distinct alternatives, such as decentralized topologies, could be further studied in the edge network context.

Future research may attempt to bridge the gap between the paradigms of distributed learning and federated learning, since current research is partitioned into these paradigms. Determining which techniques translate from one context to another would be fruitful for both application areas.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 308–318, 2016.

[2] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 1175–1191, 2017.

[3] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294, 1988.

[4] S. Caldas, J. Koneny, H. B. McMahan, and A. Talwalkar. Expanding the reach of federated learning by reducing client resource requirements, 2019.

arXiv:1812.07210.

[5] M. Caprolu, R. Di Pietro, F. Lombardi, and S. Raponi. Edge computing perspectives: Architectures, technologies, and open security issues. In *2019 IEEE International Conference on Edge Computing (EDGE)*, pages 116–123, 2019.

[6] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz. Revisiting distributed synchronous sgd, 2017. arXiv:1604.00981.

[7] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. arXiv:1406.1078.

[8] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems 25*, pages 1223–1231. Curran Associates, Inc., 2012.

[9] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C. Lin. Edge of things: The big picture on the integration of edge, iot and the cloud in a distributed computing environment. *IEEE Access*, 6:1706–1717, 2018.

[10] R. C. Geyer, T. Klein, and M. Nabi. Differentially private federated learning: A client level perspective, 2018. arXiv:1712.07557.

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014. arXiv:1406.2661.

[12] W. G. Hatcher and W. Yu. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access*, 6:24411–24432, 2018.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[14] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Quantized neural networks: Training neural networks with low precision weights and activations, 2016. arXiv:1609.07061.

[15] X. Jia, S. Song, W. He, Y. Wang, H. Rong, F. Zhou, L. Xie, Z. Guo, Y. Yang, L. Yu, T. Chen, G. Hu, S. Shi, and X. Chu. Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes, 2018. arXiv:1807.11205.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, 2012.

[17] B. Li, W. Wen, J. Mao, S. Li, Y. Chen, and H. Li. Running sparse and low-precision neural network: When algorithm meets hardware. In *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 534–539, 2018.

[18] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. C. Liang, Q. Yang, D. Niyato, and C. Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.

[19] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training, 2020. arXiv:1712.01887.

[20] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[21] A. H. Mohd Aman, E. Yadegaridehkordi, Z. S. Attarbashi, R. Hassan, and Y. Park. A survey on trend and classification of internet of things reviews. *IEEE Access*, 8:111763–111782, 2020.

[22] F. Rosenblatt. The perceptron: A perceiving and recognizion automaton. Report 85-460-1, Cornell Aeronautical Laboratory, January 1957.

[23] R. Šajina, N. Tankovi, and D. Etinger. Decentralized trustless gossip training of deep neural networks. In *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 1080–1084, 2020.

[24] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[25] Z. Tang, S. Shi, X. Chu, W. Wang, and B. Li. Communication-efficient distributed deep learning: A comprehensive survey, 2020. arXiv:2003.06307.

[26] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(2):869–904, 2020.

[27] C. Xie, S. Koyejo, and I. Gupta. Asynchronous federated optimization, 2019. arXiv:1903.03934.

[28] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762, 2019.

Defenses in adversarial machine learning

Tuomas Väisänen
tuomas.m.vaisanen@aalto.fi

Tutor: Blerta Linqvist

Abstract

Machine learning has been adapted into various tasks in several different domains. The existence of adversarial examples, however, is a threat to machine learning models that are being used in safety-critical areas. A lot of research is being done on attacks against machine learning models but the defenses research is lacking. To understand the threat facing these models it is important to understand how they are being guarded. This paper provides a review on types of defenses used against adversarial examples.

KEYWORDS: adversarial machine learning, defenses

1 Introduction

In recent years, machine learning (ML) systems, especially deep neural networks (DNN), have been applied successfully in a range of tasks. Deep neural networks have been used in domains such as: image and speech recognition and natural language processing. DNNs have also been used for playing games, where OpenAI trained an AI in the video game Dota 2 and beat a world champion team [**?**]. Because of ML based systems achievements in domains like these, they are being increasingly used in

safety-critical tasks. Auto manufacturers like Tesla are using ML systems as part of their autonomous vehicles to recognize road signs and objects. Some other safety-critical fields where ML has been applied are facial recognition, malware detection and medicine. The critical nature of these fields would require the ML models to be resistant against attacks but while great machine learning algorithms have been designed, their security and robustness is not good.

As with any security-critical fields there are adversaries, such as hackers, organized crime and rogue states looking to cause harm to these systems. If an adversary managed to get a classifier used in autonomous vehicles to misclassify stop signs or objects on the street it could be dangerous. Scenarios like these are the reason why the security aspect of machine learning has become a concern recently. A field devoted to this subject is adversarial machine learning (AML) and a lot of research on this has been published over recent years. Szagedy at al. [17] first learned that neural networks will misclassify an image if an imperceptible perturbation is applied to the image. This means that two images can look like a cat to a human but the image with perturbation applied to it will make the classifier classify it, for example, as an airplane. This same perturbation can also work on different networks to misclassify the same input. Such adversarial examples exist in all application domains.

This paper reviews types of defenses used in adversarial machine learning and their effectiveness. Section 2 shortly presents the types of attacks, so it is easier to understand how the defenses in the following sections work. Section 3 reviews the types of defenses used. Lastly, section 4 discusses the current state of defenses in adversarial machine learning and provides concluding remarks.

2 Types of Attacks

To understand how defenses, work in adversarial attacks, you should understand what they are defending against. In this section, we give a basic overview of the current forms of attacks.

2.1 Evasion Attacks

The idea of evasion attacks is that the inputs to the model are modified by the attacker in a way as to make the model generate false predictions

and evade detection. A real-world example of evasion attacks can be for example putting stickers on different objects or road-signs to cause the model to misclassify them. In the case of the stop sign Eykholt at al. [7] attack managed to fool the classifier into believing the Stop sign is a Speed Limit 80 sign. Evasion attacks are currently the most common type of attack against machine learning systems [5]

Figure 1. A physical perturbation on a stop sign. [7]

2.2 Poisoning Attacks

In poisoning attacks, the inputs are modified during training and the model is trained on these poisoned inputs. This type of attack can happen when the attacker has access to the database the model is being trained on. The poisoning of the inputs would then be the attacker inserting or modifying samples to the database. [18] An example of a specific poisoning attack is the boiling frog attack. In a boiling frog attack the model is poisoned incrementally over a period of time. The model fails to identify the poisoning as it has been slowly built up. [6]

2.3 White-Box Attacks

White-Box refers to the setting the attack takes place in. In a white-box setting the attacker has full knowledge of all of the aspects of the machine learning model used for classification. This means that the attacker has knowledge of the algorithm used for training, training data distribution and the parameters of the trained model. With this knowledge the attacker is can create sophisticated attacks to target weaknesses in the model. [5] A goal of adversarial defenses is to secure machine learning models against white-box attacks [**?**] .

2.4 Black-Box Attacks

A Black-Box setting is the opposite of a white-box one. In black-box setting the adversary has no knowledge of the machine learning model used for classification. In a black-box attack the adversary uses inputs and outputs to the model to find out its weaknesses. Black-box attacks are more pragmatic in applications as usually models are not open source. [18]

3 Proposed defenses

Different strategies have been proposed for protecting machine learning models. Barreno at al. [11] proposed three different ways of categorizing defenses. These categories are regularization, information hiding and randomization. Defenses have also been classified into two proactive and reactive strategies [19]. Overall, three different main categories for countermeasures exist [18]. Gradient masking or obfuscation, robust optimization, and adversarial examples detection.

Adversarial examples show how modern machine learning models are easily broken. In this section we investigate the proposed countermeasures against adversarial attacks. In recent years hundreds of papers have been written about proposed techniques for securing these models but almost all these methods have been broken. Most of the defense methods in this section have been broken but there are valuable lessons to be learned from the way these methods were broken and evaluated. [2] These lessons are discussed in the section after this one. The defense strategies in this section are split into the three different main categories.

3.1 Why are machine learning models hard to secure?

OpenAI et al. [13] proposed two reasons for why adversarial examples are hard to defend against. The first reason is because it is difficult to construct theoretical models about the adversarial example crafting process. This crafting process is a complex optimization process and because there are no suitable theoretical tools to describe this process it is even harder to make theoretical arguments about possible defenses. The second reason OpenAI outlined is because adversarial examples require ML models to create good outputs for all possible inputs.

3.2 Gradient Masking

The gradient is an important part of many machine learning algortihms and it is used for finding the optimal solution.

Since most attacks are based on the classifier's gradient information, the gradient masking defense strategy tries to hide the gradient information to prevent adversaries from exploiting it. [18] The way the attackers utilize the gradient is that they look for example a picture of a cat and then test which direction in the picture space makes the probability of a different class increase. After a direction is found the attackers perturb the input in a direction that would make the model misclassify it. [13]

Defensive Distillation

Distillation is a technique used to reduce the size of DNN architectures [18]. Papernot et al. [15] first introduced a defensive distillation method based on the distillation technique. The point of defensive distillation is to produce a model with a smoother output surface and one that is less sensitive to perturbations.

Figure 2 shows an overview of the defense. How the defense works is described by Papernot et al. [15] with "We first train an initial network F on data X with a softmax temperature of T. We then use the probability vector F(X), which includes additional knowledge about classes compared to a class label, predicted by network F to train a distilled network F^d at temperature T on the same data X."

In the paper authors demonstrated that using defensive distillation the success rates of attacks drop from 95% to 0.5%. As with many defensive strategies, later works on black-box attacks demonstrated that the defensive distillation method can be easily broken [14].

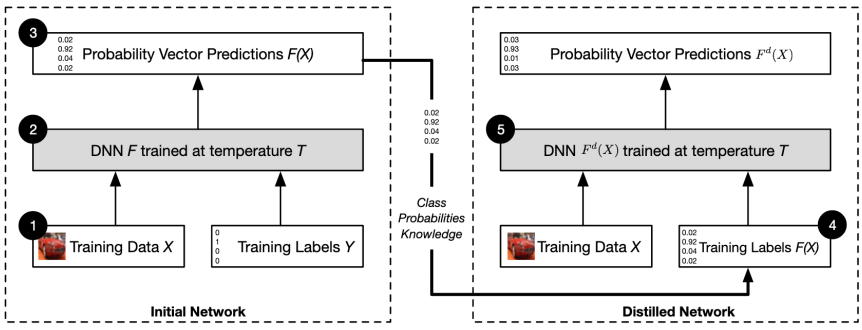


Figure 2. An overview of the defensive distillation defense. [15]

Defense-GAN

Samangouei et al. [16] proposed a method of utilizing Generative Adversarial Networks (GAN) [8] trained on legitimate training samples to "denoise" adversarial examples. The basic idea of Defense-GAN is to project a potential adversarial example onto the benign data manifold, before feeding the image to the classifier. [18] An overview of how the Defense-GAN algorithm works is shown below in the figure 3. This method works against both white-box and black-box attacks.

Samangouei et al. [16] noted that Defense-GAN is a feasible defense method against attacks, but the success of this method is heavily influenced by the GAN. If the GAN is not trained properly the performance of Defense-GAN will decline.

The reasoning as to why this method would work in defending the model is that by adding a GAN before the classifier DNN the final classification model ends up being an extremely deep neural network. This causes a cumulative effect on the gradient which makes it either extremely small or large. This would prevent the adversary from estimating the directions of adversarial examples. [18]

As with many defense methods, even though early results looked promising and paper authors said the method works, defense-gan was proven to not reliably as a defense method. Athalye et al. created an attack that evaded Defense-Gan at a 45% success rate. [1]

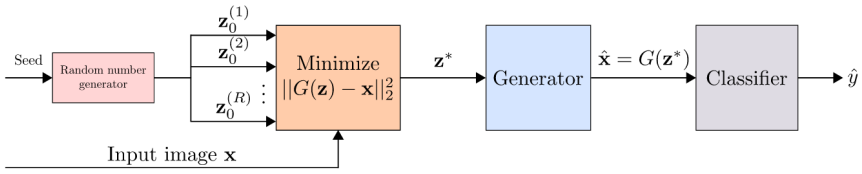


Figure 3. An overview of the Defense-GAN al. aIgorithm. [16]

Shattered Gradient

The idea of shattered gradients as a defense is to create a non-existent or incorrect gradient. This is done by non-differentiable operations. This way the connection between the models input and output is blocked so it is harder for the adversary to find the gradient to attack. Athalye et al. [1] created an attack technique called Backward Pass Differentiable Approximation that approximates the derivatives to circumvent the defenses.

3.3 Robust Optimization

The idea of robust optimization as a defense is to change how the DNN model learns to make the classifier more robust. This is done by learning model parameters that give good predictions on adversarial examples. The classifier trained this way would then classify the subsequent adversarial examples correctly. The downfall of this method is that for it to work the typically, the algorithm should know what the attack is. Then a defense is built against this specific known attack. [18]

Adversarial training
Carlini et al. [2] and OpenAI et al. [13] both described Adversarial training as one of the only strategies for securing machine learning models that seem to work. The basic idea of this strategy is to increase model robustness by generating a lot of adversarial examples and train the model so that it is not fooled by them. [13] The downfalls of using adversarial training as a defence is that it is a brute force solution. Goodfellow et al. [9] first introduced the idea of inputting the adversarial examples to the training process. The problem with adversarial training is that it only works on adversarial examples that are constructed on the original model.

As with many defense strategies adversarial training does not work against all adversarial examples. [5, 14] Narodytska et al. [12] devised a black-box attack where they treated the network as an oracle and only added perturbations to a single pixel or a small set of them. Against this attack adversarial training only improved the ability to resist the attack by 1-2%.

Zhang et al. [20] observed that the success rate of adversarial training is correlated with the attributes of the dataset. They found out that a model that is defended by adversarial training is still vulnerable because data points that are far enough from the manifold of training data are still prone to adversarial attacks. Based on this information they created an attack called the "blind-spot attack". In this attack the input images are in the "blind-spots" of the training data.

3.4 Detection

One method of safeguarding machine learning models is detection of adversarial examples. Detection works by trying to detect adversarial exam-

ples from benign ones before the model's input is predicted. If the defense detects an input as adversarial it refuses to predict its label. This method of defense however has been shown to not work well and when it does it has a high false positive rate, meaning it rejects many benign inputs. [10]

Statistical Detection

Some of the early defense methods utilizing detection used statistics to predict which inputs were adversarial and which were benign. Hendrycks and Gimpel proposed a detection method that found out that adversarial images place a higher weight on later principal components and benign images place a higher weight on the early principal components. Based on this they could split the images to benign and adversarial ones.

Prediction Consistency

One proposed way of detecting adversarial examples is by checking the consistency of the model's prediction. This defense method changes the model parameters or input examples and then checks if the outputs have changes. This method works based on the belief that adversarial examples would have unstable predictions when the parameters are changed whereas benign examples would make the classifier behave stable. So, if the model makes a prediction on X and then the parameters are randomized and the model predicts again and the result is vastly different from the previous, X would likely be an adversarial example. [18]

4 Provable Defenses

The defenses presented in the previous section are heuristic defenses. This means that they have been proven experimentally but not theoretically. While some of the defenses might work now there is no guarantee that they will not be broken in the future with more sophisticated attacks. Therefore, developing theoretically proven defenses is a key research direction and many researchers have put efforts into developing these. [19]

The basic idea of provable defenses is that the classifiers should have robustness guarantees. Carlini et al. introduced a Reluplex algorithm to verify the robustness of models. This Reluplex algorithm can be used to both verify attacks and defenses. This method was the first to be used to formally prove a defense that was at first only designed with emperical results. The method of defense proven by this method was adversarial

training. [4]

5 Conclusion

Machine learning models are vulnerable to adversarial examples. These adversarial examples can have bad consequences if they are used to attacks machine learning models used in safety-critical areas. As a result, a research field has developed to try and understand these adversarial examples and to create attacks and defenses for machine learning models. This field is adversarial machine learning.

Adversarial machine learning is rapidly developing at least on the attack research side. As for the defense research side the progress has been comparatively slow. Not only is the defense research progress slow, most of the proposed defenses are quickly broken or are shown to have been only partly evaluated by the researchers. Even the models that researchers think now might be broken when more computational power is applied. Because of this assessing progress of the field and reading papers on defenses is somewhat difficult. As a result of this a focus of the field is now to try and unify how defenses are evaluated. The purpose of this unification is to help people build better defenses and to help researchers and readers to identify downfalls of defense papers. [3] Carlini suggested in a keynote speech in 2019 that the state of adversarial defenses is the same as cryptography was in 1920s. The field has a long way to go but slow progress is being made. [2]

There have been hundreds of papers written on adversarial machine learning defenses and the subset of those papers that get re-analysed by others the only defense method that seems to work is adversarial training. Even adversarial training does not work for every possible attack, but it is currently the best defense available. Everything else that gets analysed gets broken. If you want secure machine learning today the recommendation is to use adversarial training. [2]

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, 2018.

[2] Nicholas Carlini. On evaluating adversarial robustness. CAMLIS, 2019.

[3] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness, 2019.

[4] Nicholas Carlini, Guy Katz, Clark Barrett, and David L. Dill. Provably minimally-distorted adversarial examples, 2018.

[5] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *CoRR*, abs/1810.00069, 2018.

[6] Vasisht Duddu. A survey of adversarial machine learning in cyber warfare. *Defence Science Journal*, 68:356, 06 2018.

[7] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models, 2018.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[10] Jinkyu Koo, Michael Roth, and Saurabh Bagchi. Hawkeye: Adversarial example detector for deep neural networks, 2019.

[11] Marco Barreno and Blaine Nelson and Russell Sears and Anthony D. Joseph and J.D. Tygar. Can machine learning be secure? In *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25, March 2006.

[12] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks, 2016.

[13] OpenAI, Feb 2017.

[14] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning, 2017.

[15] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks, 2016.

[16] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models, 2018.

[17] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

[18] Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil K. Jain. Adversarial attacks and defenses in images, graphs and text: A review, 2019.

[19] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning, 2018.

[20] Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S. Dhillon, and Cho-Jui Hsieh. The limitations of adversarial training and the blind-spot attack, 2019.

A review on point-based methods for 3D semantic segmentation

Peng Zheng
peng.zheng@aalto.fi

Tutor: Anton Debner

Abstract

Recently, semantic segmentation on point clouds has been attracting a wide range of attention, due to its applications in many areas, such as computer vision, 3D reconstruction, and autonomous driving. In the past few years, deep learning has shown its great performance on many vision tasks, and numerous deep learning methods on point clouds have been proposed. However, as a consequence of the irregularity and sparsity of point clouds, researchers have met a lot of challenges of how to processing the point cloud data efficiently and precisely. Therefore, different types of approaches have been proposed to achieve an accurate and efficient performance on the 3D point cloud semantic segmentation. This paper presents a comprehensive review of point-based methods on 3D semantic segmentation. Detailed ablation studies of existing methods on publicly available datasets will be given later.

KEYWORDS: deep learning, 3D point clouds, semantic segmentation

1 Introduction

With the rapid development of 3D techniques like self-driving cars and 3D reconstruction, an accurate and efficient way of 3D semantic segmentation is very much needed.

Benefitting from various types of 3D sensors, such as LiDARs, RGB-D cameras, and 3D scanners, different types of datasets have been collected and proposed, including point clouds, meshes, depth images, and volumetric grids. Among all the data formats, point cloud representation has the advantage of preserving original 3D geometric information compared with others. Therefore, 3D point clouds are very suitable for semantic tasks, and many related datasets have been proposed, including KITTI Benchmark Suite[9, 2], Semantic3D[11], ShapeNet[3], etc.

However, due to its data format is unstructured and sparse, there are challenges of applying deep learning methods on it. To overcome these problems, many methods have been proposed to try to solve them from different perspectives.

This paper is organized as follows. Section 2 describes 3D point cloud segmentation of different fine types, including semantic segmentation, instance segmentation and part segmentation. Section 3 analyzes the challenges of point-based methods for semantic segmentation. Section 4 presents a comprehensive review of point-based methods on 3D semantic segmentation and compares the pros and cons of existing methods, and ablation studies of these methods on publicly available datasets will be given later. Finally, Section 5 provides future research points and the concluding remarks.

2 Background

In the area of Computer Vision, there are three basic and important tasks, which are image classification, object detection and semantic segmentation. Each of them has its own specific applications.

2.1 Image Classification

In 2012, AlexNet won the first place of ImageNet easily, which is an image classification competition. Then more and more network architectures, like ResNet[12], DenseNet[14] have been proposed, and also algorithms

on more advanced pipelines have been proposed, such as self-supervised learning and knowledge distillation. By now, the top-1 accuracy on the validation set of ImageNet has been over 80%[30], which is a very good result.

2.2 Object Detection

In the field of object detection, traditional image processing methods often use hand-crafted features and sliding windows to detect the pattern of the local regions[21].

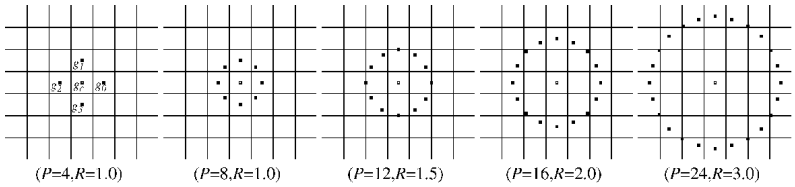


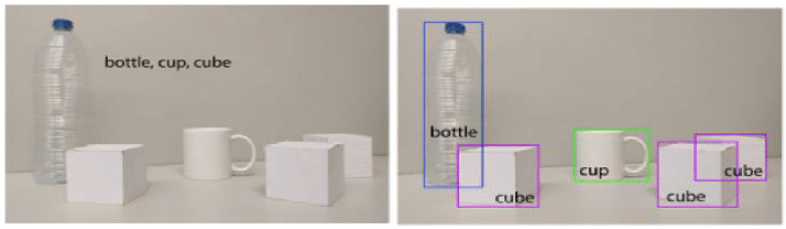
Figure 1. Local Binary Pattern

Then in the era of deep-learning, many end-to-end methods have been proposed and shown much greater performance over the hand-crafted ones. Those methods mainly consist of one-stage ones like YOLO, RetinaNe and two-stage ones like Faster-RCNN.

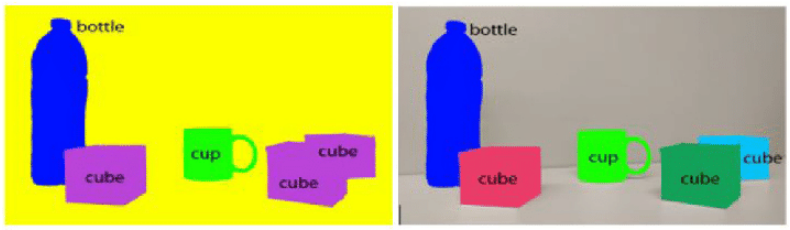
2.3 Semantic Segmentation

Different from image classification and object detection, which are scene-level task and object-level tasks, semantic segmentation is a both scene-level and pixel-level tasks, which need higher precision. Semantic segmentation is a bit different from image segmentation which is a long lasting task in the field of image processing, where many great works on it have been proposed, like SLIC Superpixels[1]. Image segmentation doesn't aim at getting the label of regions, which means that classification is not necessary here. Instead, it focuses more on the texture. The outline of the semantic segmentation is to give labels to all the pixels in an image, whether the pixel is in the region of trees, cars or the background. One thing needs mentioning is that semantic segmentation doesn't need to discriminate the different independent objects which have the same category.

In the past few years, many great deep-learning methods on the semantic segmentation have been proposed, such as FCN[19], U-Net[25], DeepLabV3[4]. Most of the semantic segmentation networks follow the



(a) Image Classification (b) Object Detection



(c) Semantic Segmentation (d) Instance Segmentation

Figure 2. image classification / object detection / semantic segmentation / instance segmentation

encoder-decoder design priciple, in order to get the high resolution results and multi-scale information, the U-Net is a typical example:

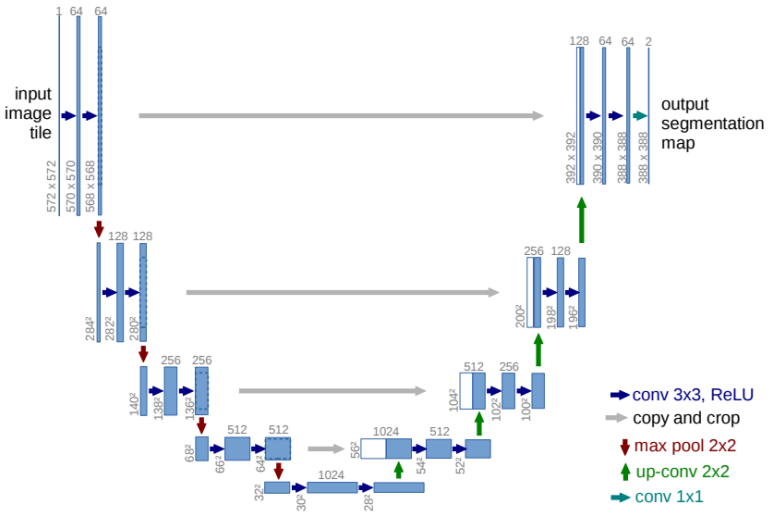


Figure 3. Architecture of U-Net

By the achievements in semantic segmentation, deep-learning have made much contribution in many fields such as autonomous driving, medical image diagnosis, as the figure below shows.

Apart from the improvement of algorithms or computation power, dataset is a key to the performance of learning-based methods. In 2D semantic segmentation tasks, there are many such dataset, such as MS-COCO[18], Cityscapes[6].

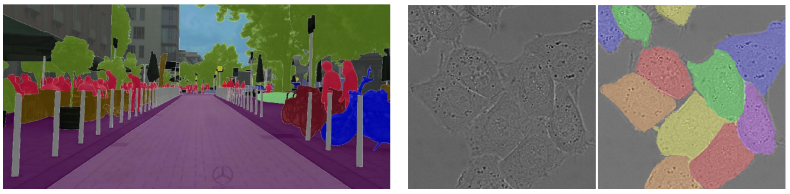


Figure 4. Applications of semantic segmentation

2.4 Point Cloud Data

Most of the computer vision tasks use images captured by 2D cameras, which are oftern in the format of 2D matrices.

Although many researchers have been using images from binocular cameras or the depth-aware cameras and made some achievements, sometimes 2D images are still not enough for many 3D tasks, such as precise autonomous driving, 3D reconstructions, and then the point cloud data shows its indipensability.

The 3D point cloud data mainly consists of 3+N elements for each unit: $(x, y, z, color, category, intensity, ...)$.

We often use LiDAR to get the point cloud data, drones, laser scanners and car MMS are the good devices to do the collection task.

However, it's difficult for previous algorithms like deep neural networks to directly use the point cloud data, so many intermediate representations of point cloud have been proposed to help make better use of the data:

There are also publicly available datasets for 3D semantic segmentation, such as KITTI Benchmark Suite[9, 2], Semantic3D[11].

3 3d Semantic Segmentation

Semantic segmentation is a basic challenge in the field of computer vision, many great methods have been proposed, in terms of precision [20, 5], and speed [26, 22]. However, they are all designed to use 2D image as the input, which cannot be directly used for point cloud data. Therefore, network architectures which are specially designed for processing point cloud data have been proposed in the past years, such as PointNet[23], PointNet++[24], SqueezeSeg[29].

Based on these networks, we can get great performance on 3D shape classification, 3D object detection and tracking, and 3D point cloud segmentation, which are also the tasks we commonly do on 2D image data.

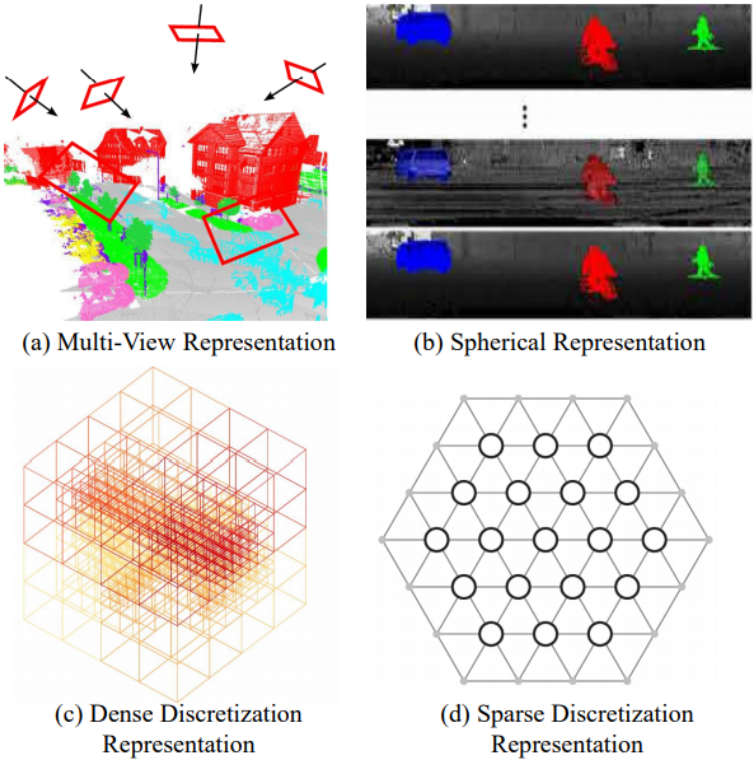


Figure 5. Intermediate representations of point cloud data

Moreover, we can divide the segmentation into three branches, semantic segmentation[23, 28], instance segmentation[13, 16] and part segmentation[27].

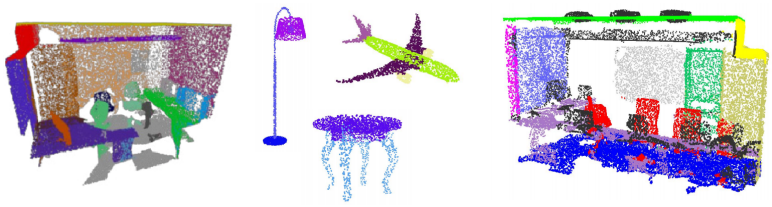


Figure 6. 3D Instance / part / semantic segmentation

In this paper, we will focus on the 3D semantic segmentation task later, where the methods can be concluded into four main categories: projection-based methods, discretization-based methods, hybrid methods, and point-based methods, which we will talk about further in Section 4.

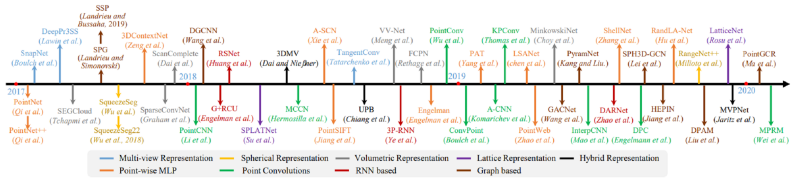


Figure 7. Overview of the deep learning-based 3D semantic segmentation methods

3.1 projection-based Methods

Since there have been much success in the area of 2D semantic segmentation, some researchers try to use adapted 2D data for representing 3D data. Among these methods, multi-view representation and spherical representation are the most commonly used two representations, which are shown in fig.6.

Multi-view Representation

By projecting the 3D point cloud data onto 2D planes from different views, Lawin et al.[17] designed a multi-stream Fully Convolutional Network to do the predictions of pixel-wise scores on 2D images. After predictions from all views, the reprojected scores of different views will be fused together to obtain the final semantic label of each point.

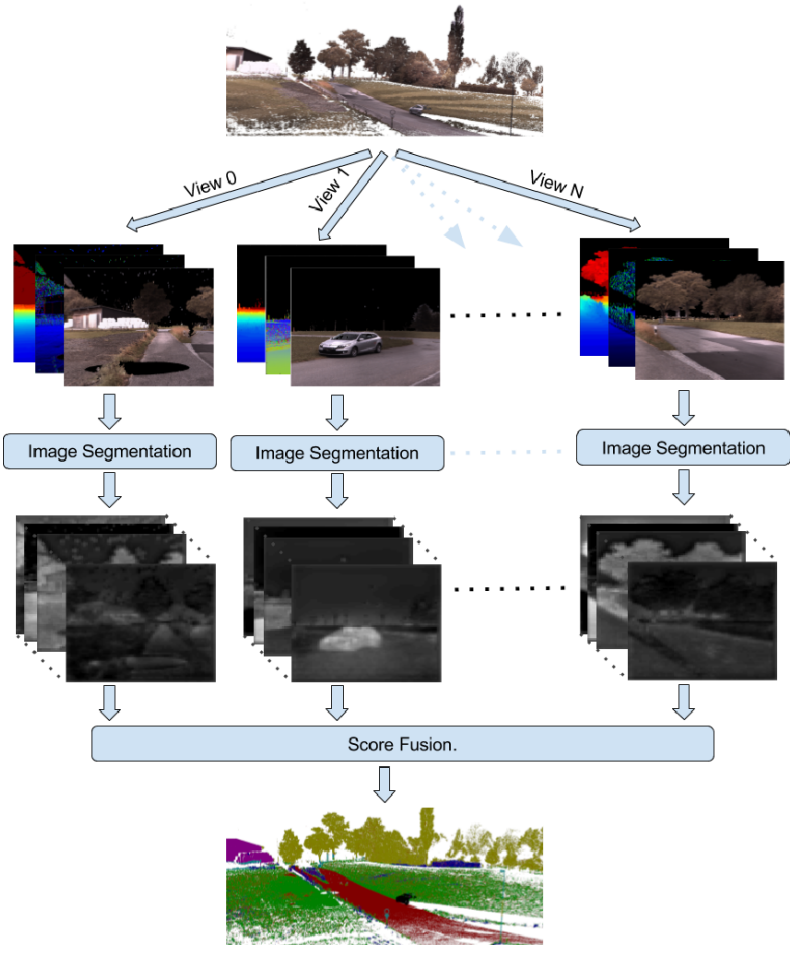


Figure 8. Multi-stream CNN for semantic segmentation with multi-view input[17].

Spherical Representation

Compared to the project from single view, spherical projection can retain more information, which is better as the labels.

However, all the intermediate representations inevitably lead to the loss of information, which would further cause the discretization and occlusions.

3.2 Discretization-based Methods

These methods make discretization on the point cloud to convert it into sparse or dense discrete representations, in the formats of volumetric or sparse permutohedral lattices, which is easier for 3D convolutions to compute.

Sparse Discretization Representation

Since the ratio of non-zero values is very small, volumetric representation is very suitable here. Due to the sparsity, dense convolutional neural networks are difficult to handle this data. Therefore, submanifold sparse convolutional network[10] was proposed. By restricting the output of convolutions to be only related occupied voxels, it managed to reduce the computation requirements a lot and solve this problem.

Dense Discretization Representation

Dense discretization methods usually divide the point clouds into dense grids, and design a 3D CNN architecture to handle the data. Huang et al.[15] divide the point cloud data into many voxels, and view all points within a voxel as a unit, where all points have the same label.

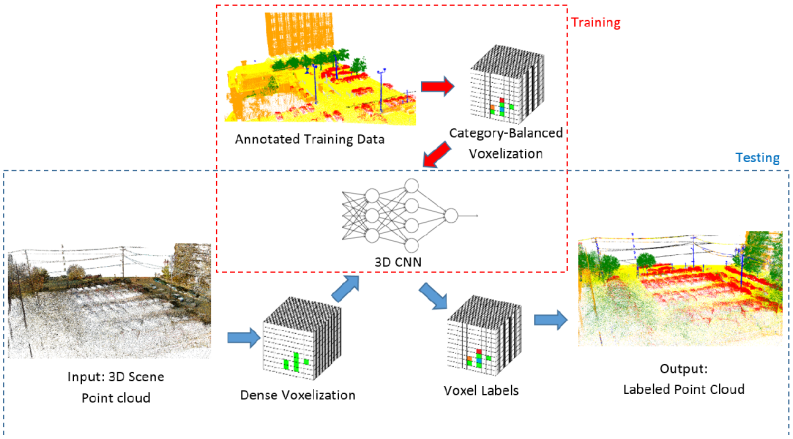


Figure 9. The labeling system pipeline of dense discretization representation[15].

3.3 Hybrid Methods

To obtain more information as much as possible to help models learn, 3D point clouds and 2D images are sometimes taken together to give the model more semantic information. These methods often apply point-based networks to do the extraction of features from sparse point sets without voxelization[7].

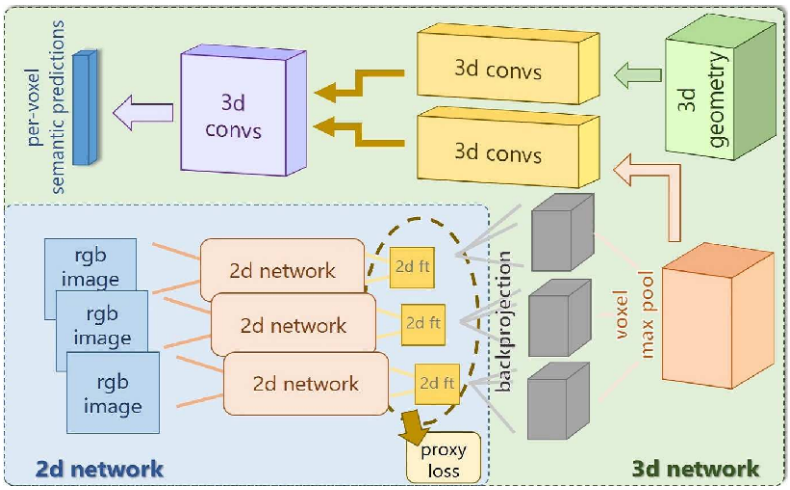


Figure 10. The hybrid architecture[15].

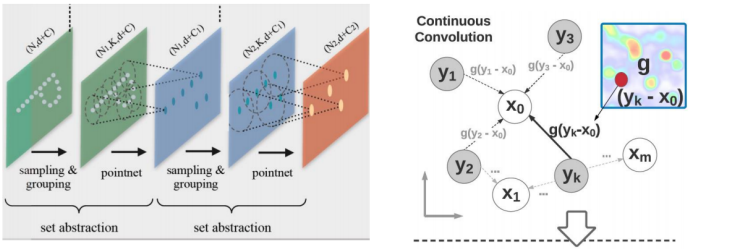
3.4 Point-based Methods

Suffering from the disorder and sparsity of 3D data, it is infeasible to apply standard CNNs on point clouds. To overcome this, Qi et al. proposed PointNet[23] and made the first success. The PointNet consists of shared MLPs and symmetrical poolings, which are designed for learning per-point features and global feature extraction, and achieved satisfying results.

Based on the PoinNnet, many point-based networks have been proposed in the following years. In general, these methods establish their networks on four main tools: pointwise MLP, point convolution, RNN and graph. In the following sections, we will focus on the methods based on the former two tools.

Pointwise MLP Methods

The basic unit of these networks is the shared MLP, which is used for efficient feature extraction. However, the features extracted by its shared MLPs cannot learn the information of local geometry and the mutual interactions between points.



(a) Point-wise MLP Methods (b) Point Convolutions Methods

Figure 11. Point-based methods.

To overcome this problem, attention mechanism, local-global feature concatenation and other methods that can help the context understanding are applied here.

In terms of the attention-based methods, Yang et al. presented the group shuffle attention which helps establish the relations between points[31], and proposed the differentiable Gumbel Subset Sampling (GSS). This module can replace the former FPS approach with less sensitivity to outliers.

Based on the local-global concatenation, Zhao et al. proposed the PS^2-Net to make the mixture of local structures and global context, which is permutation-invariant[31]. It is obvious that the fusion of local and global information is important in almost all computer vision tasks, including the point clouds tasks too.

Point Convolution Methods

Considering the problem of point-based methods from the side the big computation cost caused by the sparsity of point clouds, many methods try to complete the convolutions in a faster way. In these methods, effective convolution operators is often the key point.

Wang et al.[31] proposed a network named PCCN which is made of parametric continous convolution layers. In these layers, the kernel function is parameterized by MLPs. Similarly, Thomas et al. proposed a Kernel Point Fully Convolutional Network called KP-FCNN, based on Kernel Point Convolution (KPConv). The weights of KPConv are determined by the Euclidean distances to kernel points, while how many the kernel points is unfixed. as an optimization problem, the positions of the kernel points would be formulated in a sphere space after convergence. To improve the robustness and the comprehensiveness of fusing the information, the radius neighbourhood makes a consistent field and the grid subsampling is used for a varying densities of point clouds.

Similar to the methods used in 2D semantic segmentation, in [8], En-

gelmann et al. proposed a Dilated Point Convolution (DPC) operation to aggregate features from wider context, instead of the K nearest neighbours. This operation is confirmed to be very effective in expanding the receptive field and easily integrated into existing networks.

4 Conclusion

In the past years, based on the success of deep learning, many methods 3D semantic segmentation have been proposed and achieved great performances. However, how to overcome the sparsity and disorder of the 3D point clouds is still the main points in this area. In order to establish the relation between points with different scales, many new methods will be proposed.

Apart from the original methods on point clouds, many wonderful methods in other areas can be used into point clouds. For instance, transformer has been proved to be very effective on building the relations, not only in words, but also in images.

Besides, Graph Network has also achieved great results in many other areas, which is usually applied on point clouds.

Last but not least, the computing power. 20 years ago, many deep learning algorithms have been proposed, but cannot be verfied due to the limited computing power. Due to the sparsity of point clouds, the current computing power is not enough for applying simple and common CNNs on it. But if there is a way to improve the computing power for point clouds, instead of trying to reduce the computation of 3D algorithms in many complex ways, it would be much easier.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:2274–2282, 2012.

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, C. Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9296–9306, 2019.

[3] Angel X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Qixing Huang, Zimo Li, S. Savarese, M. Savva, Shuran Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015.

[4] Liang-Chieh Chen, G. Papandreou, Florian Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587, 2017.

[5] Liang-Chieh Chen, G. Papandreou, Florian Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587, 2017.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, M. Enzweiler, Rodrigo Benenson, Uwe Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[7] Angela Dai and M. Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. *ArXiv*, abs/1803.10409, 2018.

[8] Francis Engelmann, Theodora Kontogianni, and B. Leibe. Dilated point convolutions: On the receptive field of point convolutions. *ArXiv*, abs/1907.12046, 2019.

[9] Andreas Geiger, Philip Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[10] B. Graham, Martin Engelcke, and L. V. D. Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018.

[11] Timo Hackel, Nikolay Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. Semantic3d.net: A new large-scale point cloud classification benchmark. *ArXiv*, abs/1704.03847, 2017.

[12] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] Ji Hou, Angela Dai, and M. Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4416–4425, 2019.

[14] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[15] Jing Huang and S. You. Point cloud labeling using 3d convolutional neural network. *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2670–2675, 2016.

[16] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and M. Oswald. 3d instance segmentation via multi-task metric learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9255–9265, 2019.

[17] Felix Järemo Lawin, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, F. Khan, and M. Felsberg. Deep projective 3d semantic segmentation. In *CAIP*, 2017.

[18] Tsung-Yi Lin, M. Maire, Serge J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. *ArXiv*, abs/1405.0312, 2014.

[19] J. Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[20] J. Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[21] T. Ojala, M. Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:971–987, 2002.

[22] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *ArXiv*, abs/1606.02147, 2016.

[23] C. R. Qi, H. Su, Kaichun Mo, and L. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.

[24] C. R. Qi, L. Yi, H. Su, and L. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

[25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[26] Mark Sandler, A. Howard, Menglong Zhu, A. Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[27] Zongji Wang and Feng Lu. Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes. *IEEE Transactions on Visualization and Computer Graphics*, 26:2919–2930, 2020.

[28] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and L. Xie. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4383–4392, 2020.

[29] B. Wu, Alvin Wan, Xiangyu Yue, and K. Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893, 2018.

[30] Qizhe Xie, E. Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2020.

[31] Jiancheng Yang, Qiang Zhang, B. Ni, L. Li, J. Liu, Mengdie Zhou, and Q. Tian. Modeling point clouds with self-attention and gumbel subset sampling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3318–3327, 2019.

Usability and Security Tradeoffs in QR Code usage

Yizhou Ye
yizhou.ye@aalto.fi

Tutor: Amel Bourdoucen

Abstract

QR code popularity has increased significantly over the last few years. Emergent use cases that involve sensitive data, such as mobile payment, have increased the need for QR code cybersecurity. Various security vulnerabilities exploit the QR code property of being only machine-readable. As such, discerning a malicious QR code from a benign one is an insurmountable task for humans. However, focusing on security alone is insufficient whenever usability is involved. This paper presents the results of a literature survey that we conducted on the subject of QR code security and usability tradeoffs. We examined promising solutions that employ digital signatures for integrity, methods to increase the security of QR code reader applications, and other security measures suggested by researchers. Our results indicate that further research is necessary in order to establish best practices for QR code security.

KEYWORDS: QR code, security, vulnerability, usability

1 Introduction

QR (Quick Response) codes are two-dimensional matrix barcodes used to encode data [16]. As opposed to traditional one-dimensional barcodes, QR codes offer larger amounts of data storage along with a robust error-correction mechanism that enables scanning in spite of minor physical damage [23]. Their popularity has increased with the recent ubiquity of smartphones. The smartphone camera offers an accessible device to decode the machine-readable barcode, which typically contains a URL. However, QR codes are susceptible to tampering by an attacker. This can result in the URL changing to a malicious phishing site that threatens to steal sensitive information from the user, such as passwords or credit card information. Therefore, implementing security measures is essential. Unfortunately, increasing security in Information Technology (IT) often results in compromises in usability. As such, an ideal solution involves balancing between security and usability.

This paper aims to review existing research on QR code security vulnerabilities. These are viewed through the lens of common use cases. Subsequently, security solutions are analyzed based on their effectiveness and detriment to usability. Finally, potential topics for further research are suggested.

The paper is structured as follows: Section 2 examines the structure and common use cases of QR codes. Section 3 reviews security vulnerabilities and usability challenges. Section 4 analyzes the measures for security vulnerability mitigation while simultaneously considering usability aspects. Section 5 discusses ideas for further research. Section 6 summarizes the paper.

2 QR Code Technology

Quick Response (QR) codes are machine-readable 2-dimensional barcodes that are most commonly accessed via smartphone cameras and subsequently decoded by a QR code reader software [16]. They have a visual appearance consisting of black square dots, containing encoded information, on a white square grid [3]. There are forty different QR code versions, each with a different amount of storage capability: version 1 being the smallest and version 40 the largest [16]. Notably, version 2 is the most frequently used [15]. Typically, alphanumeric characters are encoded for

URLs. However, QR codes can also encode other structures, such as control codes and binaries [16]. The latter enables virtually any computer data to be stored inside the barcode. This can range from simple games to SQL injections if they fit within the size limitations.

The use of QR codes was initially employed by the automotive industry in Japan. Eventually, other fields discovered their benefits, including high encoding capacity, robustness, low manufacturing costs, and ease of deployment. Its robustness is reflected in an error correction mechanism based on Reed-Solomon codes, which has four levels with increasing numbers of error correction bytes [16]. As a result, QR codes are highly flexible to scanning from different angles and possess resistance against dirt and physical damage.

Figure 1. An example of a version 2 QR code. Generated using "https://www.nayuki.io/page/qr-code-generator-library", 24th of November 2020.

2.1 QR Code Use Cases

In the past, the most common use case for QR codes was serving as an attachment to billboard advertisements, where the barcode has the website URL of the company encoded. However, several other use cases have been discovered recently. This section provides a brief overview of such use cases.

Authentication

WebTicket [13] is an account management tool that uses QR codes for authentication. The process begins with a computer creating a ticket with a QR code on it. The QR code contains a script for authentication facilitated by a URL for a website, a user ID, and a password generated by WebTicket. The user can then choose to either print the ticket or scan it into a mobile device using an app. Subsequently, when authentication is required, the user displays the ticket for the computer webcam. The data that is transferred via the QR code is encrypted using a key stored on the computer. Therefore, an attacker has to compromise the user's computer in addition to the ticket.

Mobile Payment

In the context of mobile payments, reading a QR code can redirect the user to an intermediate payment agent. This functionality has been adopted by PayPal in some countries [16]. It is also widely used in China [11].

Video Games

QR codes are used in video games to enhance the playing experience. These either exist in-game or outside the game, for instance, printed in a real-world location [2]. Games that utilize printed QR codes often track the user's geolocation.

Product Tracing and Counterfeit Detection

Based on the initial use case of tracking vehicles during manufacture, QR code usage has since been adopted to track the supply chain of products. According to a study conducted in South Korea [12], consumers generally appreciated the additional information provided by QR codes to aid them in their purchasing decisions. QR codes have also been used by governments [1] to differentiate counterfeit products from legitimate ones.

3 Security and Usability

The QR code technology offers no inherent security measures to differentiate safe codes from malicious ones. Unfortunately, because they are only machine-readable, humans can not differentiate them visually either. Most security vulnerabilities exploit this property. This section discusses known vulnerabilities and associated usability challenges.

3.1 Security Vulnerabilities and Threats

According to the media, the most common attack against QR codes involves the concept of social engineering [16]. The attacker's approach can be divided into two categories: modifying an existing QR code or replacing it with a new one. Modifying, commonly referred to as tampering, involves changing the color of specific square dots, referred to as modules, on the white grid [3]. Replacing requires arguably less effort, as often covering the old code with a new one is sufficient. Furthermore, the attack vectors can be categorized as either automated attacks or attacks on human

interactions [21]. The former capitalizes on the lack of proper sanitation of input. Examples include SQL injections and command injections. The latter relies on the QR code being non-human-readable. Social engineering attacks, such as phishing, fall into this category. Once an attack is successful, the attacker can, for instance, install malware or redirect the user to a malicious phishing website [9].

Borgaonkar [4] demonstrated an attack in which Man-Machine-Interface (MMI) codes were used to delete all data from a Samsung mobile phone. A QR code was encoded with MMI instructions that the phone would then execute upon scanning. QR codes can also serve as vectors for Cross-Site-Scripting (XSS) attacks [8]. Furthermore, attacks targeting the reader software have been demonstrated [6]. These attacks aim to gain access to the sensitive permissions granted to a reader application during its installation, such as device location or the contact list. In addition to attacks initiated by an adversary, QR codes have been shown to possess security threats caused by chance. According to Dabrowski et al. [6], QR code reader software may sometimes mistakenly interpret bits of the barcode as a standalone barcode. The probability of this happening increases when QR code density increases.

Threats against the physical device, such as theft and otherwise unauthorized access, necessitate QR codes to have additional authorization mechanisms, especially when considering mobile payment systems [20]. This is also crucial when QR codes contain sensitive information intended only for authorized access. QR codes involving payment are particularly dangerous because they are typically reused multiple times during their lifetime by the same users. This enables complicated attacks that require multiple instances of exposure.

Physical QR codes associated with video games are highly dangerous if they are modified or replaced by an attacker. This is due to the players having expectations of gaining benefits upon scanning. These can range from advancing in the storyline to obtaining other rewards inside the game.

3.2 Usability Challenges

According to its name, QR codes are supposed to be able to provide a quick response. However, adding layers of security checks often slows down the process. While increasing security, this increase in time impacts other areas of usability, such as performance and user satisfaction, negatively. In

the worst case, increased time may drive away impatient users altogether.

Vidas et al. [24] performed two experiments where they posted flyers in public places across Philadelphia. These flyers had QR codes embedded with URLs following the "URL shortener" convention. They found with their experiments that users often scan QR codes with unknown URL addresses merely out of curiosity or fun. As a result, human curiosity is a major factor that has to be considered when QR codes are designed.

The curiosity caused by observing a QR code can serve as an initial motivator for a user to approach and examine a code more closely. Other factors, such as the visual design of the QR code, may also affect user decision making. Upon scanning, the reading software will typically display the URL address and prompt the user for permission to open a browser. Several studies have examined this interaction. According to Lo et al. [17], users preferred to have shortened versions of the URL displayed alongside the longer version. They also found instructions on how to scan a QR code to be useful for those with no prior experience. Wahsheh and Luccio [25] evaluated current QR code reader applications and found their malicious URL detection features lacking. They propose a solution with improved detection rates and visual security warnings for the user. Although security warnings are often ignored when users form a habit, they increase the user-perceived security of the process, which is important for usability.

One of the methods of improving QR code integrity is through the use of digital signature schemes. Focardi et al. [9] conclude that using cryptography with large keys increases the physical size of the QR code, which in turn increases the rate of failures in scans. As such, proper key size selection is crucial for usability.

4 Security Solutions

4.1 Secure QR Code Reader Software

Since users have no practical way of distinguishing a malicious QR code, QR code reader applications should perform this role. However, studies [25, 26] show that currently prevalent reader applications lack sufficient malicious URL detection tools. Choi et al. [5] propose several such tools ranging from simple blacklists to more sophisticated machine-learning-

based approaches. Blacklists are accessible as several websites provide them. However, new malicious URLs are their largest weakness. This can be remedied to some extent using a "predictive blacklist", which employs a proactive approach to identifying potentially malicious URLs. Initial attempts at predictive blacklists date back to 2008 in the work by Zhang et al. [27]. More recently, Husák et al. [14] achieved a 65% prediction accuracy after applying their sequential rule mining based approach on 12 million alerts from network-based IDS.

Suppose that we assume a theoretically perfect malicious URL detection capability for reader software. In spite of this, according to Wahsheh and Luccio [25], reader software still have security and usability issues to consider. Security-wise, malicious URL detection fails to consider offline attacks, such as SQL injections. Usability-wise, detection techniques consume time and additionally require constant access to an Internet connection, which may not always be available. Furthermore, Wahsheh and Luccio [25] also demonstrate reader applications that inherently threaten their user's privacy. These applications request more permissions from the user than what is required. Typically, such applications only require access to the camera and the Internet. However, the user's privacy may be severely compromised when potentially sensitive information, such as contacts and call history, is exposed.

4.2 Digital Signatures and Certificates

Digital signatures and certificates can be implemented inside the QR code to aid in preserving integrity and authenticity. When suitable algorithms are utilized, such as ECDSA (elliptic curve digital signature algorithm) or RSA (Rivest-Shamir-Adleman) 1,024 with no certificates, usability is not negatively impacted in a noticeable manner [9]. ECDSA can even be combined with a certificate and still maintain acceptable usability. However, larger key sizes, such as RSA 3,072 with one certificate, start to demand physically larger QR codes. The increase in size might be unpractical given the size limitations of a poster. Larger codes also increase the rate of failed scans. Mavroeidis and Nicho [19] propose a solution called the QRCS (secure QR code solution). QRCS uses a secure hash function (SHA-2 or SHA-3) to encrypt a plaintext of arbitrary length into a fixed-length hash. This property is especially useful given the size limitations of QR codes. A digital signature is then added with ECDSA, which is light-weighted compared to other algorithms making it ideal for compu-

tationally limited smartphones. The combination of a hash function and a digital signature covers the basic components of security. Encryption provides confidentiality while a signature offers integrity and authenticity.

Table 1 illustrates the results of the work done by Focardi et al. [9]. Cryptographic solutions for digital signatures are listed with varying key lengths. Each pairing has a security rating ranging from low to high based on the suggestions of the European Union Agency for Network and Information Security (ENISA) [10]. Likewise, we can observe a usability rating ranging similarly from low to high based on the experiments conducted by Focardi et al. [9]. They recommend the use of ECDSA. However, they advise against the use of certificates due to the increased overhead resulting in a detriment to usability. Certificates also require one-time access to an Internet connection after which they can be cached by the reader application. Suitable alternatives to ECDSA include AES (Advanced Encryption Standard) and HMAC (Hash-Based Message Authentication Code). The former can function well in all of its modes of operation. The latter requires suitable mechanisms for shared secret key management.

Solution	Key Length (bits)	Security	Usability
ECDSA	256	High	High
ECDSA	1024	High	High
ECDSA (cert.)	256	High	Medium
ECDSA (cert.)	1024	High	Medium
RSA	1024	Low	High
RSA	2048	Medium	Medium
RSA	3072	High	Medium
RSA (cert.)	1024	Medium	Medium
RSA (cert.)	2048	High	Low
RSA (cert.)	3072	High	Low
HMAC	128	High	High
HMAC	256	High	High
AES	128	High	High
AES	256	High	High

Table 1. Possible cryptographic solutions along with their security and usability tradeoffs. [9]

4.3 Other Suggestions

Figure 2 depicts a safety measure suggested by Lu et al. [18] that involves splitting the QR code into two parts called shadows. This is performed using a VCS (visual cryptography scheme). These shadows are consequently combined with the original code using XOR. The result is two new QR codes, one of which is stored in a cloud server and the other in an arbitrary physical location. When the latter is scanned, the user's device downloads the former code from the server and proceeds to stack them together, thus recreating the original QR code. As a result, the original QR code is never exposed to the attacker to tamper with. Lu et al. recommend implementing this method in mobile payment QR codes, where an attacker could otherwise modify the code to transfer the monetary payment to their bank account instead. This study, however, does not consider the implications of increased scanning times introduced by their methodology. Conversely, Krombholz et al. [16] suggest using complex color schemes in the QR code to support users in distinguishing tampered codes. While replacement is still possible, they hypothesize that a color thematically matching background might increase replacement costs, thus discouraging them.

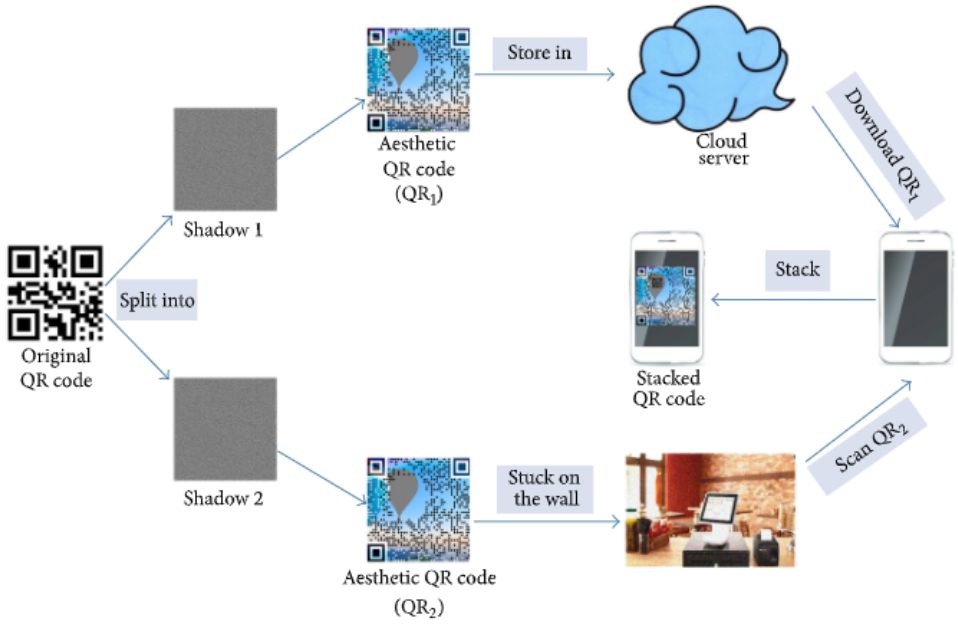


Figure 2. The steps in the solution proposed by Lu et al. [18]

5 Discussion

Perhaps the most comprehensive security solution with regards to QR codes, which manages to remain practical, involves a secure reader application along with digital signatures provided by cryptographic algorithms. This manages to thwart the most prevalent types of attacks, namely phishing attempts involving malicious URLs [16]. However, the human factor is a major factor to consider. Indeed, the design of usable security warnings has been the subject of several studies [7, 22]. With regards to security warnings involving QR codes, however, only Yao and Shin [26] have explored this niche to the best of our knowledge. Therefore, considering the differences in the context of the warnings (QR codes as opposed to other situations), we recommend further research on this subject. Furthermore, less popular digital signature schemes remain unexplored for now. Further research could potentially reveal alternatives to currently established options, such as ECDSA and AES. Moreover, we suggest investigating the effects on usability when multiple security methods are employed at the same time. As for the reader applications, we suggest investigating the possibility of a review system that enables users to leave reviews on QR codes they have scanned. When a user scans a given code for the first time, recent reviews are shown to them, which can potentially serve as indicators of the trustworthiness of the code.

6 Conclusion

To combat the security vulnerabilities found in QR codes, researchers have suggested methods, such as digital signatures and improving the security levels of QR code reader applications. Results have been promising but the potential ramifications on usability present us with another simultaneous issue to consider. While QR codes are generally known for their ease of use, the introduction of multiple layers of security can negatively impact usability by introducing longer scanning times.

In this paper, we performed a literature review to assess the current knowledge of QR code security and usability tradeoffs. Since we only found limited amounts of research on this subject, we conclude that further research is necessary in order to establish widely accepted security standards that manage to maintain high levels of usability.

References

[1] Digital tax stamps will curb fake goods - ura. 2020, accessed October 28, 2020. Available: https://www.monitor.co.ug/Business/Technology/Digital-tax-stamps-will-curb-fake-goods-URA–/688612-5462046-h6c29h/index.html.

[2] QR code in games: See how playing becomes more interactive. accessed October 28, 2020. Available: https://scanova.io/blog/qr-code-in-games/.

[3] Gianmarco Baldini, Igor Nai Fovino, Riccardo Satta, Aris Tsois, and Enrico Checchi. Survey of techniques for the fight against counterfeit goods and intellectual property rights (IPR) infringement. Technical report, Joint Research Center, 2015.

[4] Ravi Borgaonkar. Demo dirty use of USSD codes in cellular network en Ekoparty 2012, accessed October 28, 2020. Available: https://www.youtube.com/watch?v=Q2-0B04HPhs.

[5] Hyunsang Choi, Bin B. Zhu, and Bin B. Zhu. Detecting malicious web links and identifying their attack types. In *Proceedings of the 2nd USENIX Conference on Web Application Development*, WebApps '11, pages 125–136. USENIX Association, 2011.

[6] Adrian Dabrowski, Katharina Krombholz, Johanna Ullrich, and Edgar R. Weippl. QR inception: Barcode-in-barcode attacks. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, SPSM '14, page 3–10. Association for Computing Machinery, 2014.

[7] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. You've been warned: An empirical study of the effectiveness of web browser phishing warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 1065–1074. Association for Computing Machinery, 2008.

[8] R. Focardi, F. L. Luccio, and H. A. M. Wahsheh. Usable cryptographic QR codes. In *2018 IEEE International Conference on Industrial Technology (ICIT)*, pages 1664–1669. IEEE, 2018.

[9] Riccardo Focardi, Flaminia L. Luccio, and Heider A.M. Wahsheh. Usable security for QR code. *Journal of Information Security and Applications*, 48:102369, 2019.

[10] European Union Agency for Network and Information Security (ENISA). Algorithms, key size and parameters report, 2014.

[11] Shang Gao, Xuan Yang, Hong Guo, and Jia Jing. An empirical study on users' continuous usage intention of QR code mobile payment services in China. *International Journal of E-Adoption*, 10:18–33, 2018.

[12] Yeong GugKim and Eunju Woo. Consumer acceptance of a quick response (QR) code for the food traceability system: Application of an extended technology acceptance model (tam). *Food Research International*, 85:266–272, 2016.

[13] Eiji Hayashi, Bryan Pendleton, Fatih Ozenc, and Jason Hong. Webticket: Account management using printable tokens. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems, CHI '12, page 997–1006. Association for Computing Machinery, 2012.

[14] Martin Husák, Tomáš Bajtoš, Jaroslav Kašpar, Elias Bou-Harb, and Pavel Čeleda. Predictive cyber situational awareness and personalized blacklisting: A sequential rule mining approach. *ACM Trans. Manage. Inf. Syst.*, 11(4), 2020.

[15] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, and Edgar Weippl. QR code security. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '10, page 430–435. Association for Computing Machinery, 2010.

[16] Katharina Krombholz, Peter Frühwirt, Peter Kieseberg, Ioannis Kapsalis, Markus Huber, and Edgar Weippl. QR code security: A survey of attacks and challenges for usable security. In *Human Aspects of Information Security, Privacy, and Trust*, pages 79–90. Springer International Publishing, 2014.

[17] Leo Lo, Jason Coleman, and Danielle Theiss. Putting QR codes to the test. *New Library World*, 114:459–477, 2013.

[18] Jianfeng Lu, Zaorang Yang, Lina Li, Wenqiang Yuan, Li Li, and Chin-Chen Chang. Multiple schemes for mobile payment authentication using QR code and visual cryptography. *Mobile Information Systems*, 2017, 2017.

[19] Vasileios Mavroeidis and Mathew Nicho. Quick response code secure: A cryptographically secure anti-phishing tool for QR code attacks. In *Computer Network Security*, MMM-ACNS 2017, pages 313–324. Springer International Publishing, 2017.

[20] S. Nseir, N. Hirzallah, and M. Aqel. A secure mobile payment system using QR code. In *2013 5th International Conference on Computer Science and Information Technology*, pages 111–114. IEEE, 2013.

[21] Kevin Peng, H. Sanabria, D. Wu, and Charlotte Zhu. Security overview of QR codes. Technical report, Massachusetts Institute of Technology, 2014.

[22] Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor. Crying wolf: An empirical study of ssl warning effectiveness. In *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM'09, page 399–416. USENIX Association, 2009.

[23] S. Tiwari. An introduction to QR code technology. In *2016 International Conference on Information Technology (ICIT)*, pages 39–44. IEEE, 2016.

[24] Timothy Vidas, Emmanuel Owusu, Shuai Wang, Cheng Zeng, Lorrie Faith Cranor, and Nicolas Christin. Qrishing: The susceptibility of smartphone users to QR code phishing attacks. In *Financial Cryptography and Data Security*, FC 2013, pages 52–69. Springer Berlin Heidelberg, 2013.

[25] Heider A. M. Wahsheh and Flaminia L. Luccio. Evaluating security, privacy and usability features of QR code readers. In *Proceedings of the 5th International Conference on Information Systems Security and Privacy*, pages 266–273. CITEPRESS – Science and Technology Publications, 2019.

[26] Huiping Yao and Dongwan Shin. Towards preventing QR code based attacks on android phone using security warnings. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, page 341–346. Association for Computing Machinery, 2013.

[27] Jian Zhang, Phillip Porras, and Johannes Ullrich. Highly predictive blacklisting. In *Security: Proceedings of the USENIX Security Symposium*, 17th USENIX Security Symposium, pages 107–122. USENIX Association, 2008.

Characterizing nuclear energy conversations on Twitter

Sean Deloddere
sean.deloddere@aalto.fi

Tutor: Aqdas Malik

Abstract

Nuclear energy could play an important role in containing global warming. However, it is a controversial topic. Public opinion has an impact on technological potential and policy. While there has been prior research on the public opinion on nuclear energy before, there has been no research of this kind using Twitter yet. The use of social media platforms as data sources for research has increased recently; however, as a whole, it is still fairly new. The goal of this paper is to use data gathered from Twitter to characterize nuclear energy conversations. The data is gathered through the Twitter API and pre-processed. Using topic-modelling the most prevalent topics are gathered. Sentiment analysis is performed to analyse the sentiment of the conversations. 5 topics were extracted from the tweets, 'Energy production and climate' was the largest topic. Sentiment was slightly positive and slightly objective.

KEYWORDS: nuclear energy, public opinion, Topic Modelling, Sentiment Analysis, NLP, Twitter

1 Introduction

With 188 parties currently partaking in the Paris Climate Agreement, showing their commitment to containing global warming, nuclear energy could come to play a significant role in the future [1]. Nuclear energy is a clean energy source, meaning it has low CO_2 emissions [2]. In 2018, nuclear energy produced 5% of global energy, more than hydro, wind and solar combined, making it the largest clean energy source available [3]. This could make it an ideal candidate for countries attempting to shift their energy productions towards lower carbon emission alternatives. Nevertheless, nuclear energy has been a polarizing topic since its conception [4]. From safety concerns, its link with nuclear weapons and the issue of radioactive waste, to its high energy returned on investment (EROI) and environmental impact, nuclear energy has been a point of discussion for decades now.

Furthermore, it is not only experts having these discussions, regular people often do too. Unfortunately, common knowledge about nuclear energy is not always accurate. When asked, around 70% of students in a British survey responded that they thought less use of nuclear power would reduce global warming [5]. Nuclear energy is also generally seen as risky and dangerous, referring to major accidents such as the ones in Chernobyl and Fukushima [6]. However, when inspecting deaths per Terawatt-hour, a metric to measure safety of energy production, nuclear energy accounts for a death rate of only 0.07, compared to , e.g., the 24.6 death rate of coal [7]. For reference, 1 Terawatt-hour is the annual energy consumption of 27.000 people in the EU. This makes nuclear energy one of the safest methods of energy production [8]. Other studies show that people worry about developing cancer due to their proximity to a nuclear power plant, or that nuclear scientists do not have enough knowledge to handle nuclear waste, while both claims have been disproved [9]. There is also no evidence that countries with nuclear energy programs are more likely to seek or acquire nuclear weapons [10].

Researching the public opinion on nuclear energy has been done before; however, this research was based on polling and surveys. These surveys usually gather the opinions of a few thousand people at most. The rise of social media in the last decade has opened up opportunities for research on a totally different scale. With its 330 million users worldwide, and an average of 500 million tweets, short messages which are limited to

280 characters, being published every day, Twitter serves as the perfect medium for research into conversations [11]. Due to this considerable amount of users, Twitter has become a very powerful tool to gain real-world insight. In recent years, much research has been done leveraging the large amount of data twitter provides as a tool to examine how people in general discuss certain topics [12, 13, 14, 15, 16, 17, 18]. This large amount of data can be automatically analysed effectively using natural language processing (NLP) techniques, which will be discussed later on in the paper. The machine learning techniques are perfect for these applications as they grow more accurate with more data input.

The goal of this research is to analyze the current public discourse on nuclear energy by defining the most prevalent topics being discussed in nuclear energy conversations and evaluating the sentiment polarity and subjectivity for each of these topics.

2 Methods

2.1 Data collection

In this phase, tweets were collected using Twitter's Application Programming Interface (API), Developer [19]. We used Python version 3.8.3 for both the data gathering as well as processing and analyzing [20]. We employed Tweepy, a library from Python, to interact with the API [21]. There are multiple methods to extract tweets from twitter, the API provides options to either collect historic tweets, that are already on the platform, or stream relevant new tweets. In this paper, the latter was used. From October 28 to November 29, 39400 tweets were extracted using 'nuclear power', 'nuclear energy', 'atomic power', 'atomic energy', 'fission power', 'nuclear fission' and 'thermonuclear energy' as relevant key words. We detected whether a tweet was a retweet or not, and did not save retweets to avoid duplicate bodies of text in the data set. Only English tweets were selected.

2.2 Data cleaning and pre-processing

To optimally prepare the data for NLP analysis, standard pre-processing techniques were performed [22]. Employing re, a Python library, the text was transformed to lower case, and hashtags, punctuation, apostrophes,

numeric values, quotation marks, newline characters, hyperlinks, subsequent spaces and spaces at the beginning of a tweet were removed [23]. The text was then tokenized; this is the process of splitting the text into smaller pieces, in this case into separate words. This was done utilizing Python library sklearn [24]. The result is a Document-Term Matrix with a bag of words for each tweet. This matrix was then transposed to obtain the Term-Document Matrix used in topic modelling. The most frequent words in the data set, as well as for each topic, were displayed using wordcloud and matplotlib, both Python libraries [25, 26].

2.3 Topic Modelling

The first form of processing performed on the data was topic modelling. This was conducted using latent Dirichlet allocation (LDA), a probabilistic model to organize collections of discrete data, e.g., text, based on latent topics [27]. Most similar research also uses LDA and it is proven to be effective [28]. Python library gensim was employed for performing LDA [29]. LDA requires a number of topics as input and returns that amount of lists of words with weights that are representative of a topic. The topic label itself is assigned based on these words and their weights. Several different models were trained with different inputs, including the complete Term-Document Matrix, only nouns, only nouns and adjectives, and only nouns and adjectives without the most common words. Selecting only nouns and adjectives from the Term-Document Matrix was achieved employing Python library nltk [30]. For each of these models a number of topics ranging from 2 to 20 was evaluated based on semantic similarity of the list of words and comparison of representative tweets for each topic. Ultimately the model trained on only nouns and adjectives without the most common words, 'nuclear', 'power' and 'energy', with input number of topics as 5 was chosen. This model was then employed to determine for each tweet the probabilities of it belonging to each topic. Tweets were classified as belonging to a certain topic according to the highest probability.

2.4 Sentiment Analysis

In order to get a grasp of how people felt about the topic of nuclear energy sentiment analysis was performed. Python library TextBlob was employed for this purpose [31]. We inspected the sentiment polarity and

subjectivity of each tweet. TextBlob determines the polarity and subjectivity utilizing the AFINN lexicon, in which each word has been assigned a polarity and subjectivity score for each of its meanings [32]. The mean of the values of each meaning is then returned, and the mean of the values of all the words in the tweet constitute the polarity and subjectivity values of the tweet itself. The polarity value ranges from -1.0, most negative, to 1.0, most positive. The subjectivity value ranges from 0.0, most objective, to 1.0, most subjective [33]. Polarity and subjectivity of the entire data set as well as of each topic independently were evaluated. Determining the polarity and subjectivity of each topic was done through assigning weights to each tweet within a topic based on the probability determined by the LDA model and calculating weighted means.

3 Results

3.1 Word Frequency

Before any processing, the 39400 tweets that were gathered contained 51144 unique words, out of 551479 words in total. The 200 most occurring words are illustrated in Fig. 1. After processing, the 20 most occurring words are 'nuclear' (31206 times), 'power' (20663 times), 'energy' (13365 times), 'plant' (3591 times), 'plants' (2692 times), 'new' (2403 times), 'just' (2101 times), 'like' (2064 times), 'need' (1942 times), 'atomic' (1860 times), 'wind' (1790 times), 'solar' (1714 times), 'waste' (1587 times), 'world' (1506 times), 'people' (1475 times), 'trump' (1459 times), 'green' (1417 times), 'going' (1336 times), 'years' (1321 times) and 'climate' (1300 times).

3.2 Topic Modelling

The words contributing to the topic model for each topic and their subjectively assigned label are displayed in Fig. 2. The largest topic that was detected was labeled 'Energy production and climate', and contained tweets that discussed nuclear energy in the context of climate change and alternative energy sources. The second largest topic, labeled 'Trump and international conflict', contained tweets referring to Trump's nuclear policy, the Iran nuclear deal and Pakistan's nuclear weapons. The third largest topic was labeled 'Power plant near nature reserves' and contained tweets mostly petitioning against power plants being built near nature re-

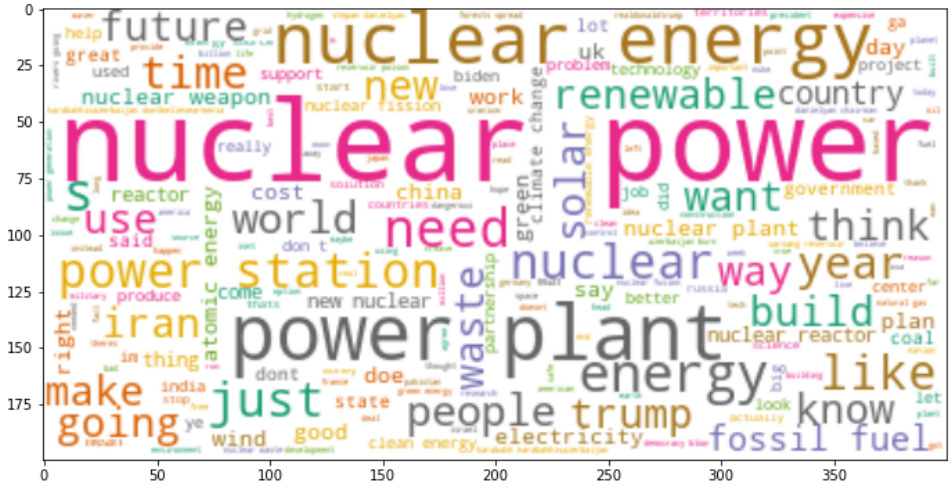


Figure 1. Wordcloud of top 200 words.

serves, particularly near RSPB Minsmere, a nature reserve in England. The forth largest topic was labeled 'science' and contained a variety of tweets discussing technicalities of nuclear energy or using terms such as 'fission' and 'fusion'. The smallest topic was labeled 'conflict between Azerbaijan and Armenia' and contained tweets about a recent quote of Stepan Danielyan, Chairman of the Center for Partnership for Democracy, allegedly urging the spread of nuclear waste in territories of Karabakh. For each topic the amount of tweets it contains and a representative tweet (with 95% or higher probability of belonging to the specific topic) was selected and displayed in Table. 1.

3.3 Sentiment Analysis

The mean polarity sentiment of all tweets was evaluated at 0.0695, indicating a slight positive sentiment. There were 16591 positive tweets (polarity score > 0), 8386 negative tweets (polarity score < 0) and 9202 neutral tweets (polarity score = 0). The mean subjectivity of all tweets was 0.379, indicating the tweets are slightly more objective than subjective. The mean polarity and subjectivity of each topic are displayed in Table. 2. 'Energy production and climate' and 'Power plant near nature reserve' have a more positive sentiment while 'Azerbaijan and Armenia conflict' has a more negative sentiment, when compared to the total data set. 'Energy production and climate' and 'Trump and international conflict' are the most subjective topics, while 'Azerbaijan and Armenia conflict' and 'Science' are the most objective topics. A scatter plot of the polarity and subjectivity score of all tweets is illustrated in Fig. 3.

Topic Label	Tweets/ Topic	Representative Tweet
Energy production and climate	13162	@fmeikle Germany is full of ironies - massive build out of wind and solar (good) but emissions still very high. Because shut nuclear early, and burning lignite for power (bad) - to keep industry with reliable supply.It's a nonsense for a supposedly rational country
Trump and international conflict	7803	Just a reminder that Sen. Jim Inhofe (R) blasted Trump Energy Sec. Dan Brouillette Friday for forcing out National Nuclear Security Admin chief Lisa Gordon-Hagerty, saying it showed Brouillette "doesn't know what he's doing in national security matters."
Power plant near nature reserve	3660	@SZCConsortium Not welcome right up against the best nature reserve in the UK PLEASE sign & retweet v.important petition below against proposed new nuclear power station right on border with Minsmere - station would have a big impact on the reserve itself
Science	5402	Current forecast is sunny so I'm shining orange. Current temperature is 69.3 degrees, current humidity is 67%. Random sunny fact: The energy created by the Sun's core is nuclear fusion.
conflict between Azerbaijan and Armenia	2649	@bbcazeri Stepan Danielyan, Chairman of the Center for Partnership for Democracy: "Blow up the Sarsang reservoir, poison the rivers going to Azerbaijan, burn all forests, spread the waste of the nuclear power plant" in territories of Karabakh." #KarabakhisAzerbaijan #DontBelieveArmenia

Table 1. Topic Clusters and their most representative tweet.

Topic Label	Mean Polarity	Mean Subjectivity
Energy production and climate	0.0896	0.404
Trump and international conflict	0.0515	0.391
Power plant near nature reserve	0.0840	0.367
Science	0.0709	0.364
Azerbaijan and Armenia conflict	0.0169	0.249

Table 2. Polarity and Subjectivity score of each topic.

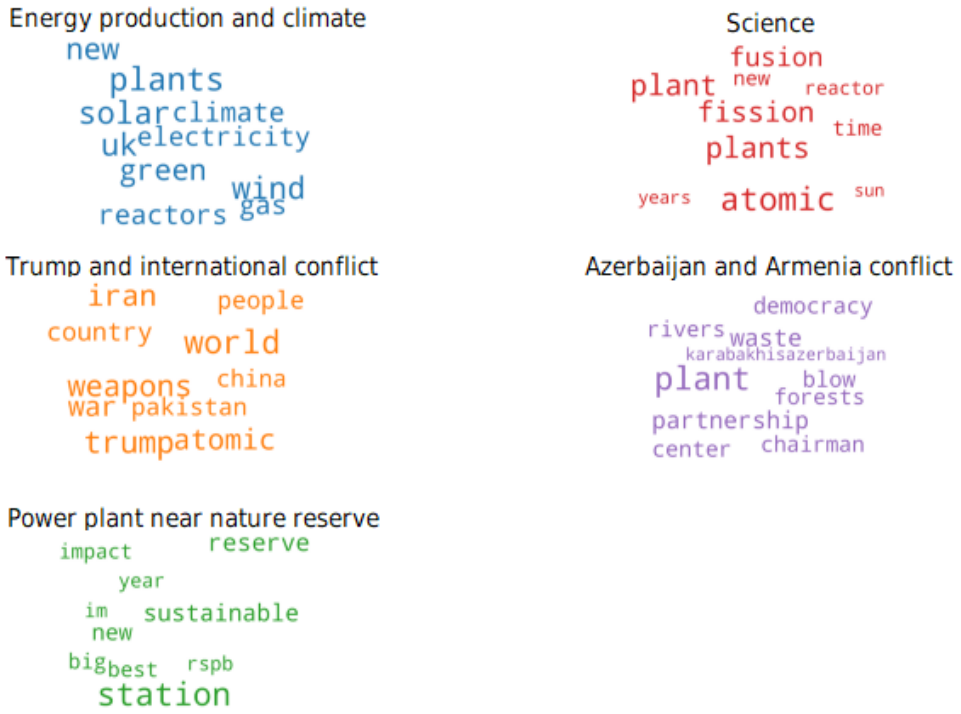


Figure 2. Wordcloud of top 10 words for each of the 5 topics.

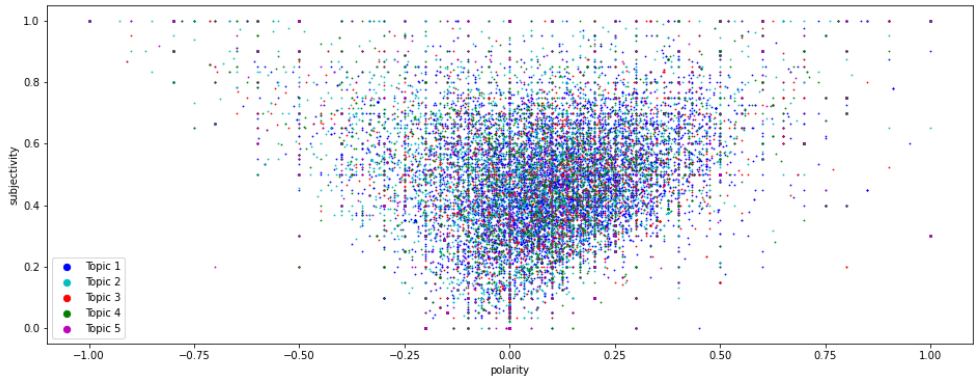


Figure 3. Scatter plot of subjectivity and polarity values of each tweet and to which topic they belong.

4 Discussion

Knowing the most prevalent topics in nuclear energy conversations and what the public's sentiment is about them can be helpful for decision makers in politics as well as in industry. Analyzing Tweets allows investigation of public discourse on a scale previously impossible. Gathering tweets over a 1 month period and analyzing them, we were able to distinguish 5 topics out of 39400 tweets and evaluate their polarity and subjectivity. Furthermore, utilizing machine learning approaches to group tweets into topics and analyse their sentiment avoids the introduction of bias from the researcher.

The topic modelling reflected general topics as well as topics that were

discussed in the media during the period of data gathering [34, 35, 36]. This demonstrates that it can be a useful and accurate tool when investigating public opinion. Energy production and climate is the most discussed topic, and solar, wind, new and green are some of the most important words in those conversations. Interestingly, safety or price was not among the most frequent words, nor a frequent topic in conversations.

While many tweets tend to be opinion-based, tweets about nuclear energy are more on the objective side. They are also generally positive, particularly when it comes to energy production and climate. However, tweets are short messages, and variance in polarity and subjectivity is high, as can be seen in Fig. 3. Topic modelling can be an effective tool to group similar tweets together, thus reduce complexity and allow meaningful differences in polarity and subjectivity to be extracted.

There are several limiting factors to this study. Firstly, even though Twitter has 330 million users, there is some representation lost for people with no access to internet or social media. Secondly, while the key words to gather the tweets are nuclear energy related, it is possible that tweets are gathered that are not necessarily related to a discussion on nuclear energy, e.g., song lyrics. Furthermore, while the model was able to accurately capture the topics that were discussed in the tweets, a longer study would have to be done to distinguish more general topics in the nuclear energy conversation. 'Energy production and climate' and 'Science' could be considered perennial topics when investigating nuclear energy conversations, while 'Trump and international conflict', 'Azerbaijan and Armenia conflict' and 'Power plant near nature reserve' are more indicative of topical subjects. This does not mean that these topics are uninformative. The first two could have a shared underlying topic of 'Politics' and the third 'Nature preservation', which are more general subjects independent of current affairs. Gathering data over a longer period of time would also result in more data in general, which would likely improve performance even further. In addition, more pre-processing could have been explored, such as removing spelling errors and lemmatization. This could result in a better performance of the LDA model and the sentiment analysis. Lastly, although the most appropriate input and number of topics for the LDA model was carefully chosen through thorough analysis of representative tweets and semantic similarity of contributing words of each topic, a more objective criteria could be created to select the optimal model.

References

[1] United nations treaty collection: 7. d paris agreement. https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en. Accessed: 12.11.2020.

[2] Kojo Menyah and Yemane Wolde-Rufael. Co2 emissions, nuclear energy, renewable energy and economic growth in the us. *Energy Policy*, 2010.

[3] Explore energy data by category, indicator, country or region. https://www.iea.org/data-and-statistics/data-tables?country=WORLD&energy=Balances&year=2018. Accessed: 12.11.2020.

[4] Shuji Takashina. *Nuclear Power in an Age of Uncertainty*, chapter 8, pages 211–224. 1984. Public Attitudes Toward Nuclear Power.

[5] Bronwen Daniel. How can we best reduce global warming? school students' ideas and misconceptions. *International Journal of Environmental Studies*, 2007.

[6] Younghwan Kim, Minki Kim, and Wonjoom Kim. Effect of the fukushima nuclear disaster on global public acceptance of nuclear energy. *Energy Policy*, 2013.

[7] Anil Markandya and Paul Wilkinson. Electricity generation and health. *THE LANCET*, 2007.

[8] Hannah Ritchie. What are the safest and cleanest sources of energy? *Our World in Data*, 2020.

[9] Shirley S. Ho, Tsuyoshi Oshita, Jiemin Looi, Alisius D. Leong, and Agnes S.F. Chuah. Exploring public perceptions of benefits and risks, trust, and acceptance of nuclear energy in thailand and vietnam: A qualitative approach. *Energy Policy*, 2019.

[10] Nicholas L. Miller. Why nuclear energy programs rarely lead to proliferation. *International Security*, 2017.

[11] Ying Lin. 10 twitter statistics you need to know. 2020.

[12] Sameh N. Saleh MD, Christoph U. Lehmann MD, Samuel A. McDonald MD, Mujeeb A. Basit MD, and Richard J Medford MD. Understanding public perception of coronavirus disease 2019 (covid-19) social distancing on twitter. *Infection Control & Hospital Edidemology*, 2020.

[13] Amir Karami, Alicia A Dahl, and Hadi Kharrazi. Characterizing diabetes, diet, exercise, and obesity comments on twitter. *International Journal of Information Management*, 2018.

[14] Hea-Jin Kim, Yoo Kyug Jeong, and Min Song. Topic-based content and sentiment analysis of ebola virus on twitter and in the news. *Journal of Information Science*, 2015.

[15] Lauren E. Sinnenberg, Christie L. DiSilvestro, Christina Mancheno, Karl Dailey, Christopher Tufts, Alison M. Buttenheim, Fran Barg, Lyle Ungar, H Schwartz, Dana Brown, David A. Asch, and Raina M. Merchant. Twitter as a potential data source for cardiovascular disease research. *JAMA Cardiol*, 2016.

[16] Salvatore Pirri, Valentina Lorenzoni, Gianni Andreozzi, Marta Mosca, and Giuseppe Turchetti. Topic modeling and user network analysis on twitter during world lupus awareness day. *International Journal of Environmental Research and Public Health*, 2020.

[17] Amir Karami, Vanessa Kitzie, and Frank Webb. Characterizing transgender health issues in twitter. 2018.

[18] King-Wa Fu, Hai Liang, Nitin Saroha, Wion Tsz Ho Tse, Patrick Ip, and Isaac Chun-Hai Fung. How people react to zika virus outbreaks on twitter? a computational content analysis. *American Journal of Infection Control*, 2016.

[19] Twitter developper. https://developer.twitter.com/en. Accessed: 12.11.2020.

[20] Python software foundation. https://www.python.org/psf/. Accessed: 29.11.2020.

[21] Tweepy. https://www.tweepy.org/. Accessed: 12.11.2020.

[22] Xiaobing Sun, Xiangyue Liu, Jiajun Hu, and Junwu Zhu. Empirical studies on the nlp techniques for source code data preprocessing. 2014.

[23] Python re. https://docs.python.org/3/library/re.html. Accessed: 12.11.2020.

[24] Scikit-learn. https://scikit-learn.org/stable/. Accessed: 29.11.2020.

[25] Wordcloud. http://amueller.github.io/word_cloud/. Accessed: 29.11.2020.

[26] matplotlib. https://matplotlib.org/. Accessed: 29.11.2020.

[27] David M. Bei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. 2003.

[28] Carina Jacobi, Wouter van Atteveldt, and Kasper Welbers. Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, 2015.

[29] Gensim. https://radimrehurek.com/gensim/. Accessed: 12.11.2020.

[30] Natural language toolkit. http://www.nltk.org/. Accessed: 29.11.2020.

[31] Textblob. https://textblob.readthedocs.io/en/dev/. Accessed: 12.11.2020.

[32] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, 2011.

[33] Textblob documentation. https://textblob.readthedocs.io/en/dev/quickstart.html. Accessed: 29.11.2020.

[34] Trump 'asked for options on strike on iran nuclear site'. https://www.bbc.com/news/world-middle-east-54972269. Accessed: 12.11.2020.

[35] Fresh fears for minsmere as pm prepares special environment speech. https://www.eadt.co.uk/news/rspb-minsmere-sizewell-c-damage-1-6926669. Accessed: 12.11.2020.

[36] Armenian political scientist calls for ecological terrorism. https://defence.az/en/news/148530. Accessed: 12.11.2020.

A Comprehensive Survey for Deep Learning Compilers

Sinan Lin
Sinan.lin@aalto.fi

Tutor: Hiroshi Doyu

Abstract

The deep learning compiler is a domain-specific compiler for parsing, optimizing, and generating binaries for a trained neural network model on a wide range of hardware platforms. This article comprehensively introduces the general compilation process of deep learning compilers and provides an in-depth comparison of three widely-used compilers, TVM, ONNC, and MLIR. These three compilers are widely used, but have significant difference in their designs. Specifically, this work summarizes the overall design, workflow, and optimization strategies of the three different compilers. Finally, the paper highlights what are similarities, differences and possible complementarity among the three compilers.

KEYWORDS: Deep Learning Compiler, Model Inference, Deep Learning

1 Introduction

In recent years, the rising popularity of Artificial Intelligence applications has created the demand for deploying trained deep learning models for inference. Currently, there are two approaches for deploying a neural network model.

One approach is that hardware vendors develop and provide their deep learning inference frameworks aiming at their specific hardware design [17]. For example, Nvidia provides TensorRT for Nvidia Turing architecture [28], and Intel develops OpenVINO for its processors [6]. However, these frameworks highly depend on vendor-specific mathematical computing libraries, which results in losing interoperability for reusing optimization strategies from other inference frameworks [13]. Therefore, further optimizations have to be implemented in all inference frameworks from scratch, which leads to a great number of redundant efforts. Therefore, this solution is not sustainable in the long term [5].

The second approach to deploying a deep learning model is to use a deep learning compiler. Deep learning compilers are extensions to traditional compilers. They can take data with more complicated structure as input, lower the representations with the high-level abstraction (such as mathematical functions and operations), and generate optimized and executable code for diverse hardware, such as CPU, GPU, and accelerators. The deep learning compiler solution contains several benefits. First, it reduces the cost of deploying on heterogeneous hardware. Second, it provides graph-level optimization for neural network models, while conventional compilers cannot do such domain-specific optimizations. Third, the optimization strategies for high-level intermediate representation (IR) can be reused for deploying to different hardware. The strengths of deep learning compilers have grasped attention from companies and research institutions, increasing the number of deep learning compilers implemented. This paper provides a comprehensive review and comparison for three mainstream deep learning compilers: TVM [5], ONNC [14], and MLIR [11]. The paper analyzes the front-end design, optimization techniques, scheduler, and back-end solution.

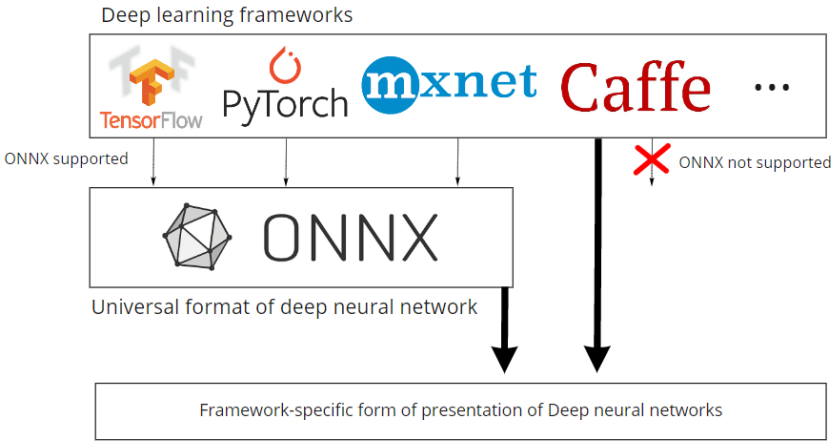
2 Background

2.1 Deep Learning Frameworks

Deep learning frameworks are used for training and inference of deep neural networks, and they can be categorized based on the way they create, represent, and run computations [19]. This section introduces two

typical frameworks, Tensorflow [1] and PyTorch [19] and one support framework ONNX [15] that is designed to be a universal open standard deep learning format.

Figure 1. An overview of deep learning frameworks



Tensorflow [1] is an end-to-end platform for deep learning. It has a comprehensive, flexible ecosystem of tools for engineers and researchers to build state-of-the-art models and employs static computational graphs with primitive operators to represent a neural network model.

PyTorch [19] is the next generation of Torch, a Lua-based deep learning framework. It constructs deep neural models with dynamic computational graphs, where the primitive functions in Python can use inside the models.

Open Neural Network Exchange(ONNX) [15] is an open standard format for representing deep learning models. Models from different deep learning frameworks can be converted into the ONNX format, which helps to achieve the model interoperability between different frameworks.

2.2 Deep Learning Hardware

The hardware for deep learning inference can be categorized into two types based on the purpose of their designs. 1) General-purpose hardware that is designed for multiple tasks, and it has been designed and created with extensive knowledge and experience, and 2) AI-customized hardware that is specialized for performing deep learning inference with customized circuit design.

General-purpose Hardware

modern **CPUs** support vector instructions (SIMD instruction) for faster processing of vectorized data. For example, the AVX instructions in Intel x86 architecture enables the CPU to speed up the matrix multiplication task [8]. However, the CPU is designed for a wide range of applications and tailored to fit both data-intensive and compute-intensive programs. Thus, it results in a moderate number of arithmetic logic units (ALU) and cores and a relatively large cache.

The **GPU** offers fewer instruction sets, but instead, it has a larger number of ALUs and cores and a smaller storage space for running limited operations on a smaller data-set with massive parallelism. Also, GPUs have a shared memory design for reducing the overhead of context switching.

AI-specific hardware is designed to utilize the property of the deep neural network, and it significantly diverges in terms of architecture and computes primitives. For example, the hardware Tensor Processing Unit (TPU) [9] was developed by Google in 2015 for deep neural network training and inference. It contains matrix multiply units, which enables matrix multiplication to be a primitive operation of hardware, and speed up the inference to 15-30 times faster than contemporary general-purpose hardware [9].

3 Deep learning compilers

3.1 TVM

TVM is an end-to-end compiler designed for engineers who want to optimize and deploy their models to production. It is a mature toolchain for all compilation stages from parsing a model from deep learning frameworks, automated optimization on both high-level and low-level, to code generation, and it is well-encapsulated so that developers with no professional knowledge on compilers can manipulate it effortlessly.

3.2 ONNC

ONNC focuses on the compilation on NVDLA-based accelerators. ONNC has similar functionalities with TVM, but it only supports the ONNX format and does not support automated operator fusion and auto-tuning. In addition, ONNC is the first compiler that has primitive support for

NVDLA-based accelerators, and it provides a memory scheduling algorithm, which is useful for the edge device with limited memory.

3.3 MLIR

MLIR is an extension of LLVM compiler infrastructure [12]. MLIR has a 'dialect' component that provides extensions to represent complex data structure system and offers high-level abstraction of IRs. It also offers a standard dialect as a bridge for the conversion between different IRs. It offers high-level abstraction of data types and IRs to increase the reusability between different compilers. This allows the optimization passes from other compilers to be reusable. For example, the TensorFlow ecosystem has several compilers for different target hardware, such as XLA HLO to the general-purpose device, TensorFlow Lite to the mobile device, and CoreML to the neural engine of Apple, and the implementation of corresponding dialects and their mapping to the standard dialect enables the conversion among different compilers. Therefore, developers can avoid a large amount of redundant work, and the compiler on one domain can benefit from the optimization passes of other domains.

4 An Overview of the Architecture Design

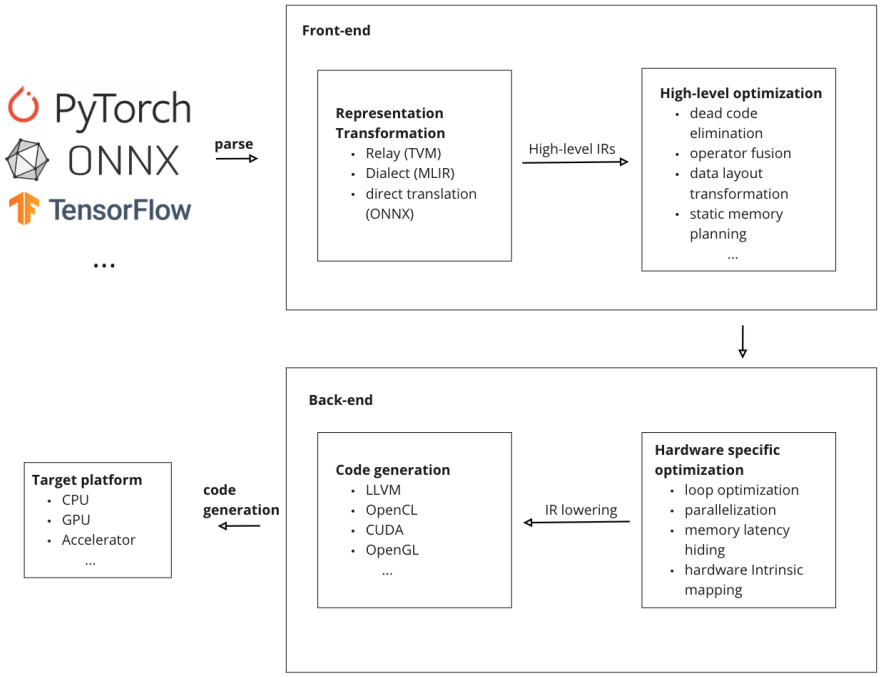


Figure 2. The general workflow for the front-end of deep learning compilers

Deep learning compilers are used for deep learning tasks, especially deploying trained models from deep learning frameworks for inference. The workflow of the compiler can be categorized into two compilation processing stages, the front-end, and the back-end, as shown in Figure 2. The front-end stage is responsible for parsing the model to a representation that the compiler can recognize and perform graph-level optimization. In the front-end stage, the trained models in text form are translated into the specific intermediate representation (IR) of deep learning compilers in the form of the directed acyclic graph, control flow graph, or static single assignment form [2] that represent computation operation and data dependency between operations. Therefore, graph-level optimizations can be used on the IR to fuse operations and optimize data layouts [5]. The back-end stage aims to emit optimized machine code of models. In this stage, the optimized IR from the front-end can be lowered to loop-based tensor expression and then be further optimized for characteristics of target hardware. For example, in deep neural networks, the convolution and fully connection operation can be decomposed into matrix-matrix multiplication and matrix-vector multiplication [27] [25]. In the back-end stage, the optimized IRs can be further optimized for characteristics of the target hardware, and then they can be 1) translated into source code (such as CUDA [22], and OpenCL [23]) and compiled by using general-purpose compilers, or 2) mapped to the instruction of accelerators that has custom instruction set architectures.

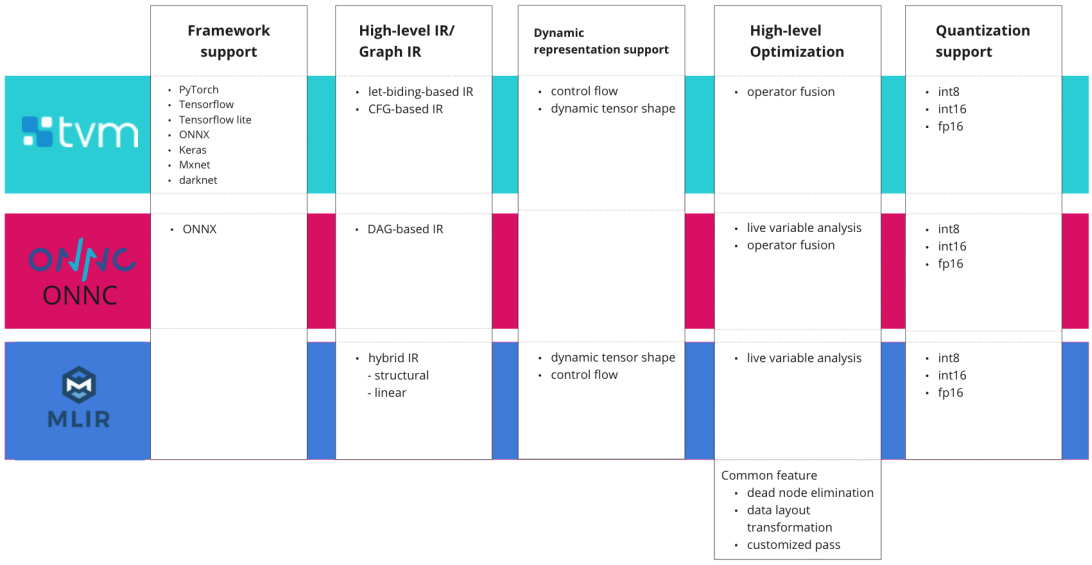
5 Front-ends

The front-end of the deep learning compiler is mainly responsible for two tasks. First, it translates the input data that can be the trained models in Python and also can be the data-serialization format [15] of a computation graph, to its intermediate representation. Second, it performs graph-level optimizations, including hardware-independent optimization and data layout transformation.

5.1 Representation Translation

In order to represent the computation of neural network models, compilers should be able to translate the models from the text form or directly

Figure 3. Comparison of frontend among TVM, MLIR and ONNC



from the Python interface into intermediate representations of compilers. In this way, these compiler-specific IRs can be further recognized and built to be a computational graph [13] for profiling. The front-end of TVM is Relay [21], and it is responsible for translating work for TVM. It provides a flexible Python interface for the support of deep learning frameworks, including TensorFlow, PyTorch, MxNet, and ONNX, and translate models into Relay IR that is a purely-functional, statically-typed intermediate representation based on the design of OCaml programming language [18] [21]. However, ONNC only supports the models in ONNX format. The front-end of ONNC performs translation by using a one-to-one mapping from ONNX IR to ONNC IR. In terms of MLIR, it offers no built-in support for translating work, but Tensorflow implements a parser for MLIR to parse models of Tensorflow to MLIR IR [7] [11].

5.2 Graph-level Optimization

The intermediate representation of a deep neural network can be viewed as a computational graph where the nodes correspond to one or several arithmetic operations, and edges show the dependency between data flow. It provides a global view of operators and avoids specifying how each operator is implemented. The graph-level optimization includes semantics-preserving optimizations and data layout optimizations. Semantics-preserving optimizations rewrite computational graphs by removing redundant nodes and computation or by folding multiple nodes into a single node. These

graph-level optimization passes in the deep learning compiler are similar but different from the optimization strategies of traditional compilers.
(1) **Constant folding** is to statically compute parts of the graph that rely only on constant initializers, avoiding the need to compute them during runtime.
(2) **Redundant node elimination** is to remove all redundant nodes without changing the graph structure. Unlike dead code elimination [26], redundant node elimination works for nodes that contain one or more arithmetic operations in a computational graph.
(3) **Operator fusion** that folds several operators into a single kernel, so multiple operators can be performed without saving the intermediate result back to the memory, and thus reduce the execution time [5]. For example, a Conv-Relu fusion folds the Relu operator as the weight of the convolution operator.
The implementations of operator fusion are different for ONNC and TVM. TVM divides graph operators into four types based on the change in the shape of input and output dimensions. (1) injective operators (one-to-one mapping, such as Add operator), (2) reduction operators (several-to-one mapping, such as Sum operator), (3) complex-out-fusable operators (element-wise map, such as convolution), (4) infusable operators (such as Sort) and use specific fusion rules upon the combinations of these four types of operators. For example, multiple injective operators can be fused into a single one compound injective operator [5]. Differently, ONNC does not group these operators. Instead, it offers fusion rules for a range of specific combinations of operators.

5.3 Layout Transformation

This optimization changes the data layout to optimize access locality on the target hardware for performance improvements. For example, TVM targeting on CPUs optimizes the layout of convolution operations from NCHW into N[C/c]HWc , in which N, H, and W stand for the size of the batch, height, and weight respectively, and c means the split sub-dimension of channel C. Therefore, it is convenient to have smaller pieces of a channel as the innermost dimension as a channel is often larger than the width of SIMD instructions of x86 CPUs [16].

6 Back-ends

The back-end is responsible for hardware-dependent optimization and emitting binary code for target devices. In general, the optimized IRs from the front-end part can be lowered to LLVM IR and reuse the optimizer and code generator of LLVM infrastructure. The computation in neural network models is basically matrix multiplication, which benefits from aggressive loop optimization, including loop parallelization, tiling, and reordering [2]. However, general-purpose compilers usually generate poorly performing code when loop-based programs are directly passed to them [13]. In order to avoid this situation, deep learning compilers usually apply two approaches before delivering code to general-purpose compilers. First, the back-end maps a certain set of IR instructions to hardware intrinsics that are highly optimized for a specific combination of arithmetic operations. Second, the back-end performs target-specific loop optimization. As for the second scene, the optimal loop transformation depends on the characteristics of the hardware design, and each combination of choices of marking the implementations of loop tiling, reordering, vectorization is a scheduling option for the back-end. This leads to huge search space for the back-end optimizer.

6.1 Scheduling

Deep learning compilers apply different scheduling approaches to tuning the parameter to determine the loop parameters with the best performance for the target hardware. In the following, different scheduling techniques are discussed.

(1) **Black-box tuning** requires no machine-specific information and randomly try a different combination of loop transformation parameters, such as blocking size, loop unrolling factors, and loop order, on the target device. If a configuration achieves better performance than the previously-stored result, it is updated to be the current optimal configuration, and the scheduling process repeats these steps until all possible combinations are explored. This approach costs a significant amount of time and might lead to over-fitting, but the result is unbiased since it requires no prior information of hardware.

(2) **Pre-defined cost tuning** requires the information of the target hardware and pre-defines the weight of each loop transformation parameter for each supported hardware. It heuristically searches for the optimal

loop configuration based on the weights, instead of running all possibilities of loop configuration and testing on the target hardware [20]. TVM reuses the pre-defined cost scheduler from Halide and extends the primitives from CPU only to a wide range of hardware, including GPU and specialized accelerator [5] [20]. This scheduling method is efficient, and the result is optimal for the commonly used hardware. However, it is biased and not sustainable since the weights are added manually, and new weights should be provided to every new hardware.
(3) **Machine-learning-based tuning** predicts the performance after a set of loop transformation using a machine learning model. For each schedule configuration, the model takes the loop parameter as input and predicts its execution time on hardware. The model is updated by the measured runtime on the target device, and it does not require the detail of hardware. TVM offers a machine-learning-based auto-tuning module, AutoTVM, for tuning the result [4]. This approach is less biased than the pre-defined cost tuning method and more efficient than the black-box tuning method. Besides, the collected data can be stored for further use.
(4) **Polyhedral-based tuning** models loop-based source code into mathematical abstractions and uses linear programming to find the optimal affine transformations on the loop-based program without violating the program behaviors [3] [11]. This method provides a powerful mathematical framework to reason about loops in programs. MLIR uses techniques from polyhedral compilation to make dependence analysis and loop transformations efficient and reliable [11].

6.2 Code generation

As the target hardware has become increasingly diverse, the task of code emitting becomes more complicated than the traditional compilers. The back-end offers multiple code generators that work for different target devices.

General-purpose hardware
After the optimization, deep learning compilers can generate machine code from the lowered representation for target hardware. The common approach for generating CPU machine code is to lower or map the IRs to LLVM IR and reuse the LLVM compiler infrastructure to generate machine code for target CPU [12]. As for GPUs, TVM and ONNC implement a code generator that emits code with PTX [10] instructions, and the emit-

ted code can be delivered to the CUDA compiler for generating executable code.

AI-specific hardware
There is not a generic way that generates code for AI-specific hardware because of the diversity in their design. However, if the target hardware has a manually optimized C/C++ library, such as Intel MKL to CPU and NVIDIA cuBLAS to GPU, the code generator of this hardware can also be customized. Hardware providers need to implement a code generator that generates C code (TVM) or ONNC IRs (ONNC) for subgraphs and integrates the code generator into the runtime module. Beside, TVM provides a graph representation generator that can generate optimized graph representation into other forms, and it can be used for the hardware that is built on a complete graph execution engine, such as TensorRT [24]. In addition, TVM and ONNC offer primitive support for TPU and NVDLA-based accelerator, respectively.

7 Discussion

Although there is a discussion on the difference between these three deep learning compilers in sections 4 and 5, it is helpful to have an analysis on their unique feature and collaborative work.

7.1 TVM

Flexible interface for model parsing
The front-end of TVM provides users with a wide range of interfaces, maximizing the flexibility for development. Models from deep learning compilers or the text form can be parsed without manually manipulating.

Automated graph-level optimization
Unlike ONNC, the automatic operator fusion for TVM is not only simple string matching, but also means the more complex graph matching. The front-end of TVM, Relay, groups operators into different groups, and apply different approaches on the combination of groups, instead of simply the names of operators.

7.2 ONNC

Specific optimization for some hardware
ONNC primitively supports NVDLA-based accelerators [15], and ONNC provides optimized kernels for specific operations. Thus, these accelerators can be high-performing with a limited amount of time spent on auto-tuning [5].

7.3 MLIR

Multi-level IRs support
With the dialect mechanism, MLIR supports multiple IRs for one single compilation. This enables compilers developers to bridge two different compilers and reuse the optimization passes existed. Therefore, MLIR compiler tends to be popular as a tool for compiler-related development, instead of direct deployment.

7.4 Collaboration among TVM, ONNC and MLIR

The design of TVM, ONNC and MLIR are modulized, and this allows the collaboration. For example, ONNC can intergrate TVM as its front-end by simply adding a TVM IR parser, which combines the flexibility of the front-end of TVM and specific optimization passes of NVDLA-based accelerators from the back-end of ONNC. Also, the dialect mechanism can allow this portability to be even more powerful and scalable, because of the dialect mechanism from MLIR.

8 Conclusion

In summary, these three compilers have different emphasis. TVM is a tool for general developers, it provides the best flexibility in terms of the forms of models and the number of target hardware it primitively supports. ONNC compiler has a specific support for a range of hardware, which allows ONNC to be the preference for a specific group of users. However, MLIR is the most distinct among these three compilers. It does not contain a front-end for deep learning models, but it can bridge the gaps of deep learning or other domain-specific compilers to reuse optimization passes from each other. Furthermore, these three compilers can collaborate to address problems efficiently.

References

[1] Martín Abadi and et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, Savannah, GA, November 2016. USENIX Association.

[2] Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Addison-Wesley Longman Publishing Co., Inc., USA, 2006.

[3] Riyadh Baghdadi and et al. Tiramisu: A polyhedral compiler for expressing fast and portable code. In *2019 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, pages 193–205. IEEE, 2019.

[4] Tianqi Chen and et al. Learning to optimize tensor programs. In *Advances in Neural Information Processing Systems*, pages 3389–3400, 2018.

[5] Tianqi Chen and et al. Tvm: An automated end-to-end optimizing compiler for deep learning, Oct 2018.

[6] A. Demidovskij, Y. Gorbachev, and et al. Openvino deep learning workbench: Comprehensive analysis and tuning of neural networks inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 783–787, 2019.

[7] Roy Frostig, Matthew James Johnson, and Chris Leary. Compiling machine learning programs via high-level tracing. *Systems for Machine Learning*, 2018.

[8] Hassan and et al. Performance evaluation of matrix-matrix multiplications using intel's advanced vector extensions (avx). *Microprocessors and Microsystems*, 47:369–374, 2016.

[9] N. Jouppi, C. Young, N. Patil, and D. Patterson. Motivation for and evaluation of the first tensor processing unit. *IEEE Micro*, 38(3):10–19, 2018.

[10] Andrew Kerr, Gregory Diamos, and Sudhakar Yalamanchili. A characterization and analysis of ptx kernels. In *2009 IEEE international symposium on workload characterization (IISWC)*, pages 3–12. IEEE, 2009.

[11] Chris Lattner and et al. Mlir: A compiler infrastructure for the end of moore's law, 2020.

[12] Chris Arthur Lattner. *LLVM: An infrastructure for multi-stage optimization*. PhD thesis, University of Illinois at Urbana-Champaign, 2002.

[13] Mingzhen Li, Yi Liu, Xiaoyan Liu, Qingxiao Sun, Xin You, Hailong Yang, Zhongzhi Luan, Lin Gan, Guangwen Yang, and Depei Qian. The deep learning compiler: A comprehensive survey, 2020.

[14] W. Lin, D. Tsai, L. Tang, C. Hsieh, C. Chou, P. Chang, and L. Hsu. Onnc: A compilation framework connecting onnx to proprietary deep learning accelerators. In *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 214–218, 2019.

[15] W. Lin, D. Tsai, L. Tang, C. Hsieh, C. Chou, P. Chang, and L. Hsu. Onnc: A compilation framework connecting onnx to proprietary deep learning accelerators. In *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 214–218, 2019.

[16] Yizhi Liu, Yao Wang, Ruofei Yu, Mu Li, Vin Sharma, and Yida Wang. Optimizing {CNN} model inference on cpus. In *2019 {USENIX} Annual Technical Conference ({USENIX}{ATC} 19)*, pages 1025–1040, 2019.

[17] S. m. Yoo and et al. Structure of deep learning inference engines for embedded systems. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 920–922, 2019.

[18] Yaron Minsky, Anil Madhavapeddy, and Jason Hickey. *Real World OCaml: Functional programming for the masses*. " O'Reilly Media, Inc.", 2013.

[19] Adam Paszke, Gross, and et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019.

[20] Jonathan Ragan-Kelley and et al. Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. *Acm Sigplan Notices*, 48(6):519–530, 2013.

[21] Jared Roesch and et al. Relay: A new ir for machine learning frameworks. In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, 2018.

[22] Jason Sanders and Edward Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.

[23] John E Stone, David Gohara, and Guochun Shi. Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in science & engineering*, 12(3):66–73, 2010.

[24] Han Vanholder. Efficient inference with tensorrt, 2016.

[25] J Welser, JW Pitera, and C Goldberg. Future computing hardware for ai. In *2018 IEEE International Electron Devices Meeting (IEDM)*, pages 1–3. IEEE, 2018.

[26] Hongwei Xi. Dead code elimination through dependent types. In *International Symposium on Practical Aspects of Declarative Languages*, pages 228–242. Springer, 1999.

[27] Xing and et al. An in-depth comparison of compilers for deep neural networks on hardware. In *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*, pages 1–8. IEEE, 2019.

[28] R. Xu, F. Han, and Q. Ta. Deep learning at scale on nvidia v100 accelerators. In *2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pages 23–32, 2018.

Biometric authentication using brainwaves

Tommi Pulli
tommi.pulli@aalto.fi

Tutor: Sanna Suoranta

Abstract

Did you know that each individual has their own unique brainwaves that can be used for authentication? This paper shows that brainwaves can be recorded with EEG using scalp electrodes and by generating models of these recordings, they can be used as a basis of novel biometric authentication method. EEG-based authentication provides several advantages over conventional biometrics: EEG only exists on living individuals, stealing the secret is difficult, forcing an authentication is almost impossible and unlike many other biometrics, the secret can be changed.

There has been an increasing interest in research community to generate accurate way of identifying individuals with EEG. For example, a study with a group of 50 subjects were each exposed to identical set of pictures, there was found a combination of three scalp locations that was able identify all 50 subjects from that group with 100% accuracy. Another research found a way to identify 108 subjects from each other with 100% accuracy by comparing coherences of different scalp location of a pre-recorded EEG set. Even though studies have shown promising results, more research is needed to verify the results in scale and evaluate and develop the usability of the applications.

KEYWORDS: authentication, biometrics, EEG, brainwaves

1 Introduction

Biometric authentication or biometrics have been adopted as one of the key authentication methods for devices and systems. Fingerprint scanners and facial recognition can be found in almost every new phone, and voice recognition and eye-based biometrics, such as iris and retina scanners, have their own applications. However, these conventional physical biometrics have fundamental flaw of being neither confidential information nor a secret to an individual [21]. Fingerprints can be found anywhere we touch and faces and voices are being recorded all the time. One widely researched biometrics for identification and authentication are brainwaves. Brainwaves are unique for each individual and they can be read by using electroencephalogram (EEG) [14]. Utilizing EEG waves for authentication, flaws in conventional biometrics could be overcome.

To be able to use EEG waves in authentication, a Brain-Computer Interface (BCI) is needed between the brains and an application [15]. A standardized way of building BCI systems is to separate the tasks in five different phases: brain activity measurement, preprocessing, feature extraction, classification, and command translation. In the first phase, brain activity is measured with EEG. The acquired data contains noise and artifacts that are then preprocessed to minimum in the next preprocessing phase. The cleaned data is then fed to the feature extractor that extracts the most relevant components or features of the signal. These features are set of values that can be used to represent the original data in a concise and uniform way. In classification phase a machine learning algorithm is utilized to classify the signal in a certain class. This class represents some command in the used system and in the final phase this class is then translated in to the selected system command and executed.

This paper focuses on the first phase, the brain activity measurement, of the above listing in a context of subject identification. I also evaluate the current state of the research. The rest of the paper is organized as follows: Section 2 will explain the EEG method and relevant properties of EEG. Section 3 presents how EEG can be used for identification and authentication purposes. In sections 4-8 the latest proposed solutions are reviewed. Finally, section 9 summarizes the most relevant findings of this paper.

2 Electroencephalogram

EEG is a method for recording the neural oscillation of the brain using scalp electrodes [20]. The oscillation that can be observed from the scalp is generated by the pyramid cells due to their perpendicularity to the skull. Because the cells generate only a minuscule potential difference, single-cell potentials cannot be recorded with EEG. In fact, EEG always perceives a sum of oscillation happening under the skull which makes the analysis of EEG more difficult. Different stimuli activates different parts of the brain but since the EEG records the sum of the activations, finding the brain's response to the certain stimulus is difficult. However, there are similarities in brain activation that applies for all individuals. These similarities can be utilized for creating an overview of EEG behavior with certain stimulus. With deeper analysis of these similarities, Poulos et al. [14] have found empirical evidence that EEG waves actually carry genetic information by being able to distinct the EEG signals of subjects which was generated by identical stimulus.

This section presents how the neural oscillation is turned into an EEG waves and what kind of characteristics these EEG waves have. Furthermore, different protocols of acquisition methods are presented.

2.1 Acquisition of EEG signals

In EEG, the activity is measured as the potential difference between two selected scalp locations, the other being the reference location [11, 10]. Choosing the locations of the measurement points is critical for achieving the greatest amplitudes for the relevant brain response. The stimulus type determines where the relevant EEG signal is the most prominent but choosing the right reference is not as straightforward. Following two properties should be considered when choosing the best reference point or points available: Reference should contain the same noise levels as the signal points which causes the noise to be cancelled out from the signal, and the reference should not contain the measured signal because this will cause the signal to be subtracted. This leads to a trade-off situation where the reference should be located near to the signal recording points where the noise levels match but simultaneously reference should not record the actual signal of interest and thus be as far from the signal recording point as possible. Some common reference points are mastoids (the bone in front of the ear canal) and the tip of the nose. Reference can also be

calculated as an average of certain locations, for example the mastoids, or all measured locations. Furthermore, Yao [24] have presented a method for referencing the EEG signals to a point at infinity named the reference electrode standardization technique (REST).

To help to universally determine the scalp locations, an internationally recognized method 10-20 system has been standardized [2]. 10-20 system is based on landmark points nasion, inion and the left and right preaurical points, and the distance between those landmark points in percentages. Sagittal and coronal lines are drawn at 0, 10 and 20 percent from the landmark points. The electrode positions are found at the intersections of the drawn lines. The positions are named with letters F, C, P, O and T referring to the underlying frontal, central, paretial, occipital and temporal lobes respectively and integers 0-10, zero being on midline, odd numbers on the left hemisphere and even on the right. There can be a mix of letters if the electrode position is between two lobes. Position of electrodes are shown in Fig. 1.

The activity from these locations can be recorded by using either wet or dry electrodes [3]. Wet electrodes are used with conducting gel that improves the signal quality over the dry electrodes. However, the dry electrodes have an advantage from the usability point-of-view since they can be used without leaving the hair full of gel and they are also faster to set up. Usability of the electrodes are discussed later in the evaluation section.

The signal strength of EEG is usually between 50 and 100μV [16]. Many of the EEG applications use so called event-related potentials or ERPs as the control signal in the application and these ERPs have signal strength of 0.1-5μV. This fact creates a challenge of recording the relevant signal with needed accuracy i.e. having great enough signal-to-noise ratio (SNR). SNR can be increased with multiple methods, for example signal averaging, referred as preprocessing but that stage is not covered in this paper.

2.2 Types of EEG waves

EEG waves can be categorized in five different frequency bands, each of which have their own unique characteristics [2, 20]. The lowest band waves are called the delta-waves and their frequency is from 0.5 to 4Hz. Delta-waves are dominant in deep sleep and also increased during internal processing of a mental task. Second lowest band is called the theta-band. They have relative low amplitudes and frequency of 4-8Hz and are active

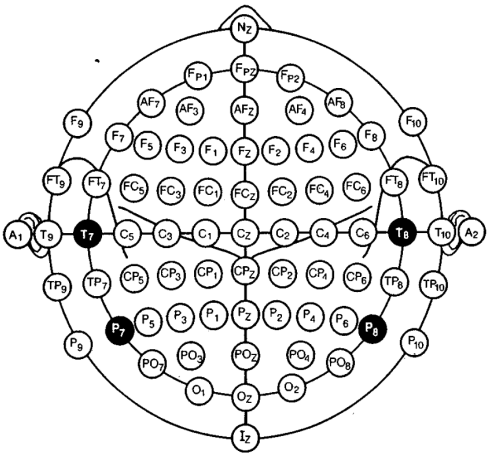


Figure 1. Top view of the 10-20 system. Nz represents the nasion, Iz the inion and A1 and A2 shows the preaurical points. [2]

during light sleep and spatial navigation, in addition to memory related tasks. The middle level frequency waves (between 8 and 14Hz) are called the alpha-waves. Alpha-waves have amplitude increase in the posterior part of the skull when visual attention is reduced. In addition, same frequency waves, called the mu-waves, are generated in motor cortex when performing or visualizing a body movement. Alpha and mu-waves are easily mixed due to the similar frequency. Fourth frequency band, the beta-waves, are oscillating at 14 to 30Hz and are amplified in mental states of focused attention and increased alertness. Beta-waves are generated in different cortical areas depending on the nature of the stimulus. The last and fifth band, the gamma-waves, has the frequency of 30Hz and above. Gamma-waves are also associated with focused attention and active processing of information. With EEG using scalp electrodes it is possible to record frequencies up to 45Hz.

2.3 Event-related potential (ERP)

ERP can be described as a brain's response to a certain event that is generated either internally (endogenous) or externally (exogenous) [20, 1]. ERPs are visible in EEG as changes that have a latency characterized by the type of event. In addition to latency, ERP waves are described with polarity and amplitude [2]. Human ERPs can roughly be divided into two different categories. When the waves peak roughly within the first 100ms after the event or stimulus, the ERP is then called exogenous or sensory since they are highly dependent on the physical attributes. ERPs happening later than 100ms after stimulus are categorized as endogenous

or cognitive ERPs [22]. The name endogenous reflects to the fact that these ERPs are generated by information processing in the brain. In literature, exogenous ERPs, or more generally EEG signals, that are visible practically immediately after the event, are often referred as evoked potentials (EP) and ERPs are called the averaging of EPs. Furthermore, event-related potentials and evoked potentials are occasionally mixed together or used as synonyms.

ERPs can be categorized also in more detailed way, though still matching the higher level properties mentioned above [12]. More detailed naming of different ERPs is based on the polarity and latency of the signals. This listing is meant to be infromative in the context of this paper rather than exhaustive. In addition, it is important to define the cortical location where the wanted ERP can be read since there might be multiple locations where the same ERP component is generated though still not correlating to the correct stimuli.

ERPs stimulated by visual stimuli are called visual evoked potentials (VEP) [12]. ERP component P100 is widely known VEP which can be recorded from the occipital area. The P100 component is most consistent in pattern-reversal VEPs [23]. Pattern-reversal stimulus can be for example a checkerboard pattern where the adjacent checkes are interchanged periodically. If the period lasts long enough that EEG signal is returned to a resting state, the recording is called the transient VEP (T-VEP) which only covers one single response. If the change period of the stimulus is shortened, response of the brain will mimic the oscillation frequency of the stimulus which leads to a periodic-like response. This type of response to a visual stimulus is called steady-state VEP (SSVEP) which generates harmonics, in addition to the stimulus frequency [9]. SSVEP can be evoked with pattern-reversal in addition to flashing, or other type of periodic visual stimulus. Factors that affect the VEP amplitude are luminance and contrast of the stimuli, subjects level of attention and the ability to keep the gaze on the stimuli. In addition, subject's possible diseases or visual disabilities naturally affects the VEPs.

Longer latency ERP component P300 is related to cognitive processing most prominent in the Fz, Cz and Pz location according to the 10-20 system [13]. P300, although named to have latency of 300ms, can in fact occur between 250 to 600ms after the stimulus [12]. This is mostly due to the fact that P300 can be evoked by any type of stimuli and the response also depends on the age, the level of attention and cognitive capability of

the subject. Duration of the P300 peak is proportional to the stimulus duration. Furthermore, a greater P300 amplitude is elicited in situations where the difference between the stimuli is greater. The difference can be between modalities, i.e. visual stimulus followed by random acoustic stimulus, or inside modalities, for example, having repeated identical visual stimulus with random easily detectable anomaly. This type of stimuli are the most used when researching the P300 ERP and it has been named the oddball paradigm. Oddball paradigm can be executed by using one easily distinguishable anomaly or with two anomalies, another being the target and second one being a distracter. Subject is instructed to react on the target anomaly only. In the two anomaly oddball paradigm, distracter stimulus elicits shorter peak than the target stimulus.

2.4 Acquisition protocols

In general, different acquisition protocols can be divided into three categories of resting states, externally stimulated tasks and mental tasks [7]. Resting state signals of EEG are recored when a subject is fully relaxed in a quiet and comfortable environment usually sitting in a chair. Subject can either have eyes open (EO) or eyes closed (EC) which affects the frequency and the location of the generated signal detected [8, 2]. Resting states are the most popular EEG acquisition protocol due to the simplicity of collection. There are no other equipment than the EEG or extra instructions needed. However, resting states are prone to errors because they have relatively low signal-to-noise ratio (SNR). Low SNR means that external factors, such as environment noises or subject's mental preoccupation, are likely to invalidate the recording. Resting states are still widely used independently and as a baseline for other protocols.

In stimulated tasks a subject is exposed to a trigger and the response is read with EEG [7]. Most widely researched stimulated tasks are visual stimuli which contains for example VEPs, SSVEPs and RSVPs. In general, stimulated tasks need equipment for stimuli generation and measuring the time after the generation to read the ERPs but visual stimuli have proven to be consistent over time.

In mental tasks subject is asked to perform a certain task in their mind and the EEG signal will be recorded [7]. Mental task can be, for example, imagined or actual body movement, imaginating a taste of food, generation of words, mentally calculating multiplications or visualizing a rotation of a figure around an axis. Mental tasks provide higher SNR for EEG signals

than resting states. Mental tasks generate ERPs so they are easier to detect than resting states due to higher SNR, however mental tasks lack an explicit starting time which impedes the measurement of latency of the ERP. Furthermore, mental tasks have been described as difficult to execute or they were not interesting enough.

3 EEG as authentication method

Electroencephalogram can potentially provide great benefits for biometric authentication [18]. Brain activity, that is the recording of EEG, has the inherent feature of only being present for alive individuals. To be able to record brain activity, one must get in touch with subject's scalp and most likely in a very precise location. This makes it very difficult to steal without the subject knowing it is happening. The brain activity and eventually the EEG waves reflects the mood and stress of an individual so they are highly unlikely to be able to reproduced them by force. Finally, it is one of the few biometrics that can be changed if the system is compromised.

When evaluating suitability of biometrics, a seven factor list of properties is used [6]. The properties are universality, uniqueness, collectability, permanence, performance, acceptability and circumvention. Universality evaluates that every one should have the characteristic, uniqueness indicates that there should not be similarities inside the population, collectability states the difficulty of collecting the biometric, permanence shows the consistency of the biometric over time, performance refers to the maximum accuracy of the biometric, the resources needed to achieve required accuracy and the external factors that affects the accuracy, acceptability evaluates if individuals are ready to use the biometrics and finally circumvention indicates the difficulty to cheat the system.

The following five sections reviews different studies that have been utilizing EEG signals for identification or authentication. Reviewing is focusing on the brain activity measurement phase: This includes the protocol for generating the brain activity, the type of the stimulus and the scalp locations of the elicited activity. Other phases, such as the classification, are briefly explained. First three researches are utilizing visual stimuli and the last two are using resting states for stimulation.

4 Exploring EEG-based biometrics for user identification and authentication

Gui et al. [5] had 32 subjects who read silently 75 of each words, pseudowords, acronyms and illegal strings in addition to 150 different writings of subject's own name. This experiment induced VEP in subjects and the EEG was read using channels Fpz, Cz, Pz, O1&2 and Oz by the 10-20 system. Eventually, only the best suited Cz -channel in acronym stimulus was further analyzed.

After preprocessing and feature extraction, artificial neural network (ANN) was used to classify the 15 extracted features for each subject. Correct classification rate (CCR) in identifying one subject from the rest was between 79.06 and 99.87%. Furthermore, the authors analyzed the preformance of identifying all of the subjects in the group. This only had CCR of 5.75-10.68%, thus showing that identification is a much more complex task than just verification that is used in authentication.

5 A High-Security EEG-Based Login System with RSVP Stimuli and Dry Electrodes

Chen et al. [3] studied authentication with RSVP signals using both wet and dry scalp electrodes. In addition to the authentication performance, this research explicitly studied the usability of the authentication application by analyzing the lower performing dry electrodes as well as the time needed for the authentication process. 29 subjects took part in this study where first each subject selected three password symbols (PS), which were mixed with other random symbols. Then symbols were shown sequentially one by one with 200ms intervals and EEG was recorded during the stimulation. The three selected PS worked as an oddball stimulus. Lastly, the EEG was measured with 28 wet electrodes or 16 dry electrodes.

Wet electrodes performed better as hypotized and could authenticate a user with 100% true acceptance rate (TAR) on average with 10.7-second stimulus. For the dry electrodes, the same time was 27 seconds. The time could be reduced in an actual application by using early-stopping method, which will only stimulate the subject for the needed time to achieve the required TAR.

6 CEREBRE protocol

An experiment conducted for the testing of the CEREBRE protocol [18] showed promising results for user identification with an accuracy of 100% with 50 subjects.

In CEREBRE, subjects are exposed to five different categories of visual stimuli, including black and white pictures of sine gratings, low frequency words, food, faces of celebrities and oddball stimulus, that was a colored version of the images listed before. Each category had 100 pictures and for each oddball stimulus, subjects were instructed to press a button as fast as possible. In addition, subjects selected a strong memory, called a pass-thought, which they were instructed to remember when a black key icon was shown. It is worth mentioning that the words used in CEREBRE were constrained to 10 characters and were taken from the Graduate Record Exam (GRE). Furthermore, the 10 foods and celebrities were chosen by conducting a separate study where different group of 44 participant listed their 10 most loved and hated foods and celebrities, from which 10 the most controversial ones were chosen for each category.

EEG was recorded from 26 scalp locations and was referenced to the average of the mastoids and the scalp electrodes were evenly distributed to cover the whole scalp rather than following the 10/20 system. For the analysis, no feature extraction was used and lightweight crosscorrelator was applied for the classification. Minimum requirements for the 100% identification accuracy was to use four of the stimulus categories including words, colored food, black and white celebrities and the oddball color targets recorded with three electrodes from the middle occipital area. The minimal classifier needed all of the available trials to be able to produce the 100% accuracy which means that the identification process took 40 seconds.

The authors state that identification accuracy can be increased more efficiently by adding different stimulus types than recording more data from one type. It was also shown in later study [19] that same accuracy can be achieved even after 516 days. It should be noted that these numbers are for identification. For authentication only binary classification is needed which could further improve the results shown here.

7 Robustness Analysis of Identification Using Resting-State EEG Signals

Di et al. [4] experimented the permanence of the resting states EC and EO with a two-week interval between the runs. In each run, 17 subjects had three sessions where their EEG was recorded from 18 scalp locations of Fz, F3&4, F7&8, Cz, C3&4, Pz, P3&4, PO7&8, TP7&8, Oz, O1&2 in reference to mastoid average. Each session contained two measurements, one in state EO and one in EC. Features were extracted from 1-40Hz with power spectral density (PSD) and Eucledian distance, support vector machine (SVM) and linear discriminant analysis (LDA) was separately used for classification.

Accuracy achieved with data from one run was 98-99% and 96-99% for EO and EC respectively where LDA classifier had the best accuracy. When adding the data from the second run, the accuracies dropped to 93-94% and 90-93% for EO and EC. However, when training and test data for the classifier came from different runs and not from a mixture of them, the accuracy dropped significantly to 30-40% when using the same features as before. After optimizing the extracted features, accruacy was increased to approximately 80%.

8 Human Brain Distinctiveness Based on EEG Spectral Coherence Connectivity

Another 100% accuracy on identifying subjects was achieved by La Rocca et al. [17] with 108 recordings from open database in resting states EO and EC by using features of PSD and spectral coherence connectivity (COH). COH contains the coherence of two EEG signals and it was calculated for each pair of the 56 recorded channels.

By using the COH features and further selecting the most suitable channel pairs, the accuracy could be improved to 100% with every macro-area of the scalp, i.e. frontal, central and paretio-occipatial in EC state. Minimum of ten locations was needed from central area of the scalp to achieve perfect identification. PSD features did not succeed as well, although reaching over 95% when fusing the PSDs from different channels.

It should again be stated that these results are for multi-class classification. For authentication purposes, minimum requirements could be lower than presented here.

9 Conclusion

In this paper I presented the basics of brainwaves, how to acquire them and how they can be used for identifying individuals. Because brain activity is constantly ongoing, the EEG signals can be read from all living individuals at anytime. Identification and authentication based on EEG signals show promising results with perfect identification among the test groups of up to 108 subjects. 100% identification was achieved in resting states EO and EC [17] but also by using different visual stimuli to generate a unique ERP [18]. These two experiments achieved the performance from very different starting points: Resting state experiment focused on the feature extraction and classification methods and ERP study in the stimulation protocol that was named CEREBRE.

Most of the studies read for this paper were solving a multi-class identification problem rather than a binary one. Multi-class classification is more complex task and the requirements for 100% identification are most probably lower than those presented. However, if EEG is the dominant authentication method in the future, it should be robust enough to work within a group of billions which will further change the minimum requirements for the eventual authentication application.

Section 3 introduced a list of seven factors that can be used to evaluate the authentication methods. The researches reviewed in this paper are focusing on the performance factor of the biometric. Even though many studies have reached the perfect accuracy in the subject group, it can still be argued that overall performance of the proposed systems is not enough for scalable and usable authentication system. Firstly, 3-56 electrodes were needed to achieve the perfect accuracy. From the usability perspective, one electrode system should be the target if multiple electrodes cannot be integrated to the mono-electrode system or as part of some existing everyday wear as the acceptability and the set up speed would decrease. Secondly, the minimum time of 10.7 seconds for perfect accuracy authentication could only be acceptible if there is a highly secure system where authentication time is not the priority. For everyday use, for example in mobile phones, even the fastest system would be too slow as there are most probably some sort of set up needed before the actual authentication process to start. Lastly, dry electrodes will be needed to achieve the acceptability for scalable system as wet electrodes would not be suitable for everyday usage.

References

[1] Michael J. Aminoff. Evoked Potentials in Clinical Medicine. *QJM: An International Journal of Medicine*, 59(1):345–362, 04 1986.

[2] P. Campisi and D. L. Rocca. Brain waves for automatic biometric-based user recognition. *IEEE Transactions on Information Forensics and Security*, 9(5):782–800, 2014.

[3] Y. Chen, A. D. Atnafu, I. Schlattner, W. T. Weldtsadik, M. Roh, H. J. Kim, S. Lee, B. Blankertz, and S. Fazli. A high-security eeg-based login system with rsvp stimuli and dry electrodes. *IEEE Transactions on Information Forensics and Security*, 11(12):2635–2647, 2016.

[4] Y. Di, X. An, F. He, S. Liu, Y. Ke, and D. Ming. Robustness analysis of identification using resting-state eeg signals. *IEEE Access*, 7:42113–42122, 2019.

[5] Q. Gui, Z. Jin, and W. Xu. Exploring eeg-based biometrics for user identification and authentication. In *2014 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–6, 2014.

[6] A.K. Jain, R. Bolle, and S. Pankanti. *Biometrics: Personal Identification in Networked Society*. The Springer International Series in Engineering and Computer Science. Springer US, 2006.

[7] Amir Jalaly Bidgoly, Hamed Jalaly Bidgoly, and Zeynab Arezoumand. A survey on methods and challenges in eeg based authentication. *Computers & Security*, 93:101788, Jun 2020.

[8] Daria La Rocca, Patrizio Campisi, and Gaetano Scarano. Stable eeg features for biometric recognition in resting state conditions. In Mireya Fernández-Chimeno, Pedro L. Fernandes, Sergio Alvarez, Deborah Stacey, Jordi Solé-Casals, Ana Fred, and Hugo Gamboa, editors, *Biomedical Engineering Systems and Technologies*, pages 313–330, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[9] Jin Lee, Deirdre Birtles, John Wattam-Bell, Janette Atkinson, and Oliver Braddick. Latency Measures of Pattern-Reversal VEP in Adults and Infants: Different Information from Transient P1 Response and Steady-State Phase. *Investigative Ophthalmology & Visual Science*, 53(3):1306–1314, 03 2012.

[10] Laura Leuchs. Choosing your reference - and why it matters, May 2019.

[11] Ya Li, Yongchun Wang, Baoqiang Zhang, Yonghui Wang, and Xiaolin Zhou. Electrophysiological responses to expectancy violations in semantic and gambling tasks: A comparison of different eeg reference approaches. *Frontiers in Neuroscience*, 12:169, 2018.

[12] B.S. Oken and T.S. Phillips. Evoked potentials: Clinical. In Larry R. Squire, editor, *Encyclopedia of Neuroscience*, pages 19–28. Academic Press, Oxford, 2009.

[13] John Polich. Updating p300: An integrative theory of p3a and p3b. *Clinical Neurophysiology*, 118(10):2128 – 2148, 2007.

[14] M. Poulos, M. Rangoussi, V. Chrissikopoulos, and A. Evangelou. Person identification based on parametric processing of the eeg. In *ICECS'99. Proceedings of ICECS '99. 6th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.99EX357)*, volume 1, pages 283–286 vol.1, 1999.

[15] Mamunur Rashid, Norizam Sulaiman, Anwar P. P. Abdul Majeed, Rabiu Muazu Musa, Ahmad Fakhri Ab. Nasir, Bifta Sama Bari, and Sabira Khatun. Current status, challenges, and possible solutions of eeg-based brain-computer interface: A comprehensive review. *Frontiers in Neurorobotics*, 14:25, 2020.

[16] D. Regan and M.P. Regan. Evoked potentials: Recording methods. In Larry R. Squire, editor, *Encyclopedia of Neuroscience*, pages 29 – 37. Academic Press, Oxford, 2009.

[17] D. L. Rocca, P. Campisi, B. Vegso, P. Cserti, G. Kozmann, F. Babiloni, and F. D. V. Fallani. Human brain distinctiveness based on eeg spectral coherence connectivity. *IEEE Transactions on Biomedical Engineering*, 61(9):2406–2412, 2014.

[18] M. V. Ruiz-Blondet, Z. Jin, and S. Laszlo. Cerebre: A novel method for very high accuracy event-related potential biometric identification. *IEEE Transactions on Information Forensics and Security*, 11(7):1618–1629, 2016.

[19] Maria V. Ruiz-Blondet, Zhanpeng Jin, and Sarah Laszlo. Permanence of the cerebre brain biometric protocol. *Pattern Recognition Letters*, 95:37 – 43, 2017.

[20] Donald L. Schomer and Fernando H. Lopes da Silva. *Niedermeyer's ElectroencephalographyBasic Principles, Clinical Applications, and Related Fields*. Oxford University Press, Oxford, UK, 11 2017.

[21] Yogendra Narain Singh, Sanjay Kumar Singh, and Amit Kumar Ray. Bioelectrical signals as emerging biometrics: Issues and challenges. *ISRN Signal Processing*, 2012:712032, Jul 2012.

[22] Shravani Sur and V. K. Sinha. Event-related potential: An overview. *Industrial psychiatry journal*, 18(1):70–73, Jan 2009. 21234168[pmid].

[23] François-Benoît Vialatte, Monique Maurice, Justin Dauwels, and Andrzej Cichocki. Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives. *Progress in Neurobiology*, 90(4):418 – 438, 2010.

[24] Dezhong Yao. A method to standardize a reference of scalp EEG recordings to a point at infinity. *Physiological Measurement*, 22(4):693–711, oct 2001.

A review of deep reinforcement learning for game AI development

Atte Viitanen
atte.viitanen@aalto.fi

Tutor: Anton Debner

Abstract

Deep reinforcement learning (DRL) has seen massive growth in the AI field for the last few years. Like many other methods in the machine learning space, DRL also has significant ties to video games and simulations due to their use in training neural networks. This paper discusses the current uses and future prospects of the use of deep reinforcement learning in game AI development and compares its advantages to more traditional methods. This is achieved by conducting a literature review on recent papers regarding DRL and game AI. The review aims to provide analysis on both the current and future states of DRL usage in game AI development from both a technological and commercial point of view.

KEYWORDS: game development, AI, deep reinforcement learning, neural networks

1 Introduction

Through significant advances since its proposal, deep reinforcement learning (DRL) has seen a steady increase in use cases across a variety of fields. These fields include robotics and autonomous vehicles, natural language

processing, computer vision, finance, healthcare and many more [12]. Another use case is in the field of video games and video game development, where this paper focuses.

Video games in general have been a a great testbed [16] and proving ground for DRL in general. DRL-based AI has seen many known achievements such as beating world champions in both traditional turn-based games such as Go in the case of AlphaGo in 2016 [20] and even teamwork-based real-time esports games such as Dota 2 in the case of OpenAI Five in 2019 [5]. In video game development, DRL has been proposed especially as a solution to traditionally more algorithmic problems such as procedural content generation [21] and game AI [24]. In addition, recently it's been proposed for use in the design process itself for problems such as adaptive gameplay [10] and automated game testing [4] as well.

This paper conducts a literature review of recent papers on the use of DRL in game development, with a focus on game AI. The aim of this review is to provide insight into what uses cases DRL has in game AI development, how it performs in said cases compared to more traditional approaches from both a quality and an ease of implementation standpoint as well as what the future potential of DRL in game AI development looks like.

This paper is organized into 3 main sections as follows. First is a background section that introduces the main concepts of DRL and it's related technologies. Second is a literature review of recent papers discussing DRL use in game AI. Lastly is a conclusion section where the some findings from the review are discussed and conclusions are drawn.

2 Background

This section briefly introduces the main concepts and fundamentals of machine learning directly related to DRL as well as provides a general description DRL itself. The section is divided into three sections. The first two sections concern the main components of deep reinforcement learning: reinforcement learning and deep learning. The last section then describes deep reinforcement learning and how it relates to the previous concepts.

2.1 Reinforcement learning

Reinforcement learning (RL) is an area of machine learning focused on learning what to do—how to map situations to actions—to maximize a scalar reward or reinforcement signal [22]. The learner is not told which actions to take, and must independently discover a mapping from situations to actions which yields the highest reward by trial and error. The reward is not necessarily immediate, and can be delayed for example in a situation where an action may have an effect on both an immediate reward and the following situation or state, and through that all the following rewards as well. These complex delayed rewards and the aforementioned trial-and-error search are typically considered the two most important distinguishing features of RL [23].

Reinforcement learning was first coined by Marvin Minsky in 1961, but was largely forgotten until the early 1980s when it became an active and well-known area of machine learning research [23]. Today reinforcement learning is used and studied in a multitude of disciplines such as game theory, control theory, economics and information theory.

2.2 Deep learning

Deep learning (DL), also known as deep structured learning, is a broad family of methods in machine learning that enables computers to learn from experience and understand hiearchies and concepts in raw natural data with minimal supervision [13]. DL solves the limitations of more conventional machine learning techniques where constructing pattern recognition or other machine learning systems required considerable expertise and care. These limitations were mainly due to having to design feature extractors to tranform raw input data into suitable internal representations or feature vectors for the learning system to use and recognize patterns in. [15].

In a simplified sense, DL approaches date processing by attempting to build complicated hierarchies and concepts present in input data out of smaller and simpler ones. These smaller hierarchies are contained in a graph many layers deep. [13] These graphs, known as deep artificial neural networks, imitate the brain in how it processes natural data.

The first general DL solution was published by Alexey Ivakhnenko and Lapa in 1967 [14]. Research quickly picked up speed especially in the 2000s as the advantages of deep neural networks became more clear as

the increased processing power of computers enabled use of more complex networks. Today deep learning is used widely throughout the marchine learning space, most typically in applications where natural data is processed raw.

2.3 Deep reinforcement learning

Deep reinforcement learning (DRL) is a form of machine learning that combines the concept of reinforcement learning with the methods used in deep learning. This integration has a long history, but with recent advancements in computation processing power, software packages and big data, deep reinforcement learning has been growing increasingly popular [16]. The data processing capabilities of neural networks help mitigate the reliance on specific domain expertise when building reinforcement learning systems as they allow for automatic feature learning straight from raw input data.

Despite its popularity, DRL has had its fair share of problems to overcome. Deep learning applications would typically require large amounts of hand-labelled training data, while RL algorithms had to be able to learn from a scalar signal that could frequently be sparse, noise and delayed. Typical deep learning algorithms would also assume data samples to be independent, while in reinforcement learning there would often be sequences of highly correlated states. Lastly, the changing of data distribution as an RL algorithm learned new behaviours also proved problematic for DL methods which would often assume a fixed distribution. [17]

3 Deep reinforcement learning in game development

This section goes over the use of deep reinforcement learning in the field of game development. The section is divided into a general section and a literature review that focuses on the more specific subject area of DRL game AI. First the general section briefly discusses the use of deep reinforcement learning in the field of game development. The literature review then aims to crasp the current state and use cases of deep reinforcement learning in game AI with examples cases. These example cases are then compared to traditional non-ML game AI solutions and their associated problems.

In recent years, the realization that video games are perfect testbeds

for modern artificial intelligence methods has spread widely in the AI research community [24]. A variety of games have since been effectively conquered, starting with classic turn-based ones and later advancing to more complex ones and even games requiring realtime input. Following the research interest, a multitude of software frameworks, environments and tookits such as the General Video Game AI (GVGAI) framework [18], the Arcade Learning Environment (ALE) [3] and the OpenAI Gym [7] have been developed. In addition to establishing benchmarks for AI comparison and development, these software platforms have even stirred up a lively competitive video game AI scene.

The allure of Video games for DRL and machine learning or AI development in general comes from the isolated and controllable playground it provides. A video game offers perfect or near-perfect repeatability and control over the game state. AI can be integrated directly into a game so that metrics on the game state may be measured and collected directly without for example monitoring the visual output of the game, simplifying the setup greatly. Even the speed of the simulation may be altered to scale with available processing power to speed up or slow down when training networks.

With all this focus on video games in the machine learning research space, its's no surprise that game development has found various uses for machine learning as well, including DRL. One major use case is in developing video game AI, to which the AI research discussed previously applies directly. While this paper will focus game AI, it should be noted that it's by no means the only use case for DRL in game development.

Due to the substantial amount of DRL and other machine learning research using games as a test bed, game AI development is a more or less an obvious use case for DRL. Traditionally game AIs intended to provide players an opponent or challenge have been done algorithmically using solutions such as finite-state machines, various pathfinding algorithms, triggers, sets of predefined, scripted actions and such. While these implementations can work when done well, they typically simply imitate some expected behaviour patterns of a real player and do not truly emulate the actual thought process a real player might go through in a given situation. As such, their behavior can seem simplistic, near-sighted or break down in atypical scenarios. In more complex games, the in-game AI is still generally considered easily distinguishable from a real player.

3.1 Game AI

Improving RTS Game AI by Supervised Policy Learning, Tactical Search, and Deep Reinforcement Learning

Barriga et al. [2] proposed a new method of using deep neural networks to select between action choices in Real-Time Strategy (RTS) game AI systems and conducted a case study investigating its use with DRL in the modern RTS game Total War: Warhammer by Creative Assembly. The case study's AI produced notably higher win rates than known state-of-the-art machine learning RTS AI solutions. Against the game's built-in traditional AI opponent, Barriga et al.'s [2] solution reached over 90% winrates in more simple macthes and $82\pm3\%$ and $77\pm3\%$ winrates in more complicates scenarios with various mixed unit types with 3v3 and 6v6 matchups respectively.

Barriga et al.'s [2] setup was of course not flawless, though: For scenarios with more than 3 opponents per side, learning was unstable and as such the results were poor. Barriga et al [2] presumed it might have been due to the game environment changing too rapidly with so many players. When using a hierarchical RL approach, Barriga et al [2] also failed to obtain obtain good results when training both high-level and low-level policies or actions jointly, failing to reach over 40% winrates against the default AI.

The network training methodology in Barriga et al.'s [2] study highlighted similarities to real humans learning a game: the best learning results were achieved by slowly introducing more complicated scenarios and difficulty as the training progressed. The paper [2] noted that the AI quickly learned techniques and strategies often used by human players such as pinning, flanking and using each unit's unique strengths to their advantage. This kind of learning inherent to the ML apprioch is something that traditional game AI methods completely lack. To achieve similar results with traditional means, the developer would have to observe human players and then manually configure state machines, scripts, or some other action templates to reproduce those behaviors.

Barriga et al. [2] also stated that the ability for a human player to learn from such an AI's increasingly complex strategies was a way to create more satisfying game experiences. In other words, a DRL-based AI could learn to perform intuitive strategies not doable with traditional AI

methods and essentially teach those strategies to the human player, drastically improving the experience. Barriga et al. [2] concluded that the accelerated research efforts in the ML game AI space together with the availability of frameworks to take advantage of readily available game replay data would likely revolutionalize both game design and playing in the near future.

Developing Combat Behavior through Reinforcement Learning in Wargames and Simulations

Boron and Darken [6] explored the use of DRL in RTS games for achieving optimal offensive behavior in small tactical engagements. However, Boron and Darken [6] took a more sophisticated approach to performance validation. Instead of focusing on the end results of the engagements, performance was evaluated based on two real-world principles of war — mass and economy of force. In simpler terms, mass is the the concentration of forces in the decisive place at the decisive time, while economy of force means allocating the minimum combat power to secondary efforts to achieve a superiority in mass. Three RL algorithms were tested; Vanilla Policy Gradient (VPG), Proximal Policy Optimization (PPO), and Trust Region Policy Optimization (TRPO). Using said algorithms, four different combat models were examined in thee different scenarios in a custom time-stepped, turn-based simulation.

While Boron and Darken's [6] results are hard to quantify in relation to traditional AI as no such AI was used, they found that the AI agents successfully employed combat strategies to maximize the two measured principles of combat. By varying the learning parameters the AI agents would switch between favoring mass or economy of force in their behaviour. The benefits of DRL were highlighted here again, as in traditional AI programming it would be very difficult to impossible to have AI intuitively maximize such real-world principles of war, especially considering that the balance of a game can change with updates and so on. It also worth noting that, depending on the implementation, with traditional game AI methods, the easy switch between favoring either mass or economy of force seen here could likely require significant amounts of work in manually re-weighing and fine-tuning various predetermined actions and patterns.

Boron and Darken [6] noted that for engagements bigger and more complex, evaluation of the AI's behavior and assessing whether a global performance optimum has been reached becomes increasingly difficult. There

is also a distinction between the concepts of optimal and tactical behaviour to consider. Due to these factors, Boron and Darken [6] resorted to simpler, smaller engagements but stated that future work would aim to validate performance in larger and more complex sccenarios. It was also noted that other kinds of missions such as defence or raids should be considered as well.

Beating the World's Best at Super Smash Bros. Melee with Deep Reinforcement Learning

Firoiu et al. [11] demonstrated a DRL-based AI capable of beating professionals in the popular retro fighting game Super Smash Bros. Melee (SSBM) by HAL Laboratory. Contrary to what's usually seen in most other fighting game AI research, SSBM is a multi-player game, and as such features more complex mechanics and a constantly evolving metagame. For training, Firoiu et al.'s [11] AI was pitted agaist both SSBM's built-in traditional AI at the highest difficulty setting as well as itself.

Firoiu et al. [11] found their AI beating the built-in version relatively quickly with various methods. Depending on the methods used, the DRL-based AI could either beat its traditional counterpart fair and square or even employ a relatively unorthodox but impressive multi-step tactic to trick the enemy into killing itself by falling off the stage. In the case of professinal players, the AI performed favorably against anyone who was willing to face it, with a total of 10 top-100 ranked players in the world having a try with multiple attempts each.

Firoiu et al.'s [11] solution was not without faults, though. The network used was only trained to play as and against specific characters at once. Firoiu et al. [11] also did not have success in training the AI with experiences from multiple game character's points of view. Such attempts resulted in the networks failing to adapt to different character strategies and ended with inferior strategies solely utilizing a set amount of basic actions that were common to the trained characters.

Firoiu et al. [11] also noted that the AI exhibited some strange weaknesses, one example being a loophole a player found where the network would behave very oddly when faced with an enemy crouching at the edge of the stage, resulting in the AI character falling of the stage. Firoiu et al. [11] hypothesized that these anomalies were likely due to a lack of diversity in training, where such odd situations were unlikely to happen.

4 Conclusions

In general, it seems that DRL is not yet, at least widely, in use commercially as an AI solution that could provide real players an opponent or challenge in a game. While current research seems to be advancing at a quick pace, it's lacking in references to any commercial or real-world use cases. While ML methods have been used for game AI in the past, an example being integrating the popular game engine Unity with a popular ML framework [8], DRL has not seen nearly as much real-world use.

Apart from massive scale AI's such as OpenAI Five [5], most research, including what was reviewed in this paper, were typically limited in some scope. As an example, Boron and Darken [6] had to resort to very simplistic engagement scenarios due to training becoming unstable and Firoiu et al. [11] only managed to train their network to play as and against specific characters at once. However, in both cases the conclusions were optimistic in regards to resolving these issues and broadening the scope.

The lack of commercial and other real-world use-cases could be due to current research only focusing on improving the AI's general performance for now. As such the goal is often to simply come up with solutions that play the best. This is not that useful for commercial game applications though, as games typically use AI to provide a challenge and thus need balancing and to adapt to the human player's skill level. Andrade et al. [1] describe game balancing as a key feature of subbesful games. The problem of balancing is described to consist in changing parameters, scenarios and behaviours in order to avoid the player either getting frustrated at the the game being too hard or bored because it's too easy. This is something that current DRL game AI in research lacks completely.

Another clear issue with commercial DRL use in games is performance. Excluding various cloud gaming services, games run on consumer devices with limited power and processing power budgets and capabilities. The performance impact of running neural networks associated with DRL are by no means light. In research, these networks are typically run on server hardware in a highly parallelized fashion with much less worry over performance constraints. As an example, Firoiu et al. [11] used over 50 or more servers in parallel when training their SSBM AI, and in an even more extreme case the famous OpenAI Five [5] AI utilized a peak of 1536 graphics processing units in parallel during training.

While there might be optimizations to be made for commercial applica-

tions, there is also the issue that neural network processing is typically accelerated with graphics processing units. However, in a consumer device, the graphics processor tends to already be under heavy stress in a game workload as it's used for graphics rendering.

The above issues are mostly temprorary though, and overall the future for DRL use in game AI looks very promising. The issue of taking into account game balancing is simply pending further research, and game AI balancing has already familiar topic in traditional RL context, for example in the studies of Andrade et al. [1] and more recently Pfau et al. [19] The issues regarding performance is also simply pending computing power advancements. In addition, graphics processor manufacturers have already started including AI-dedicated harware in consumer devices as well with the example of Nvidia's tensor cores [9]. Such developments could get rid of the need to cut back on graphics processing to be able to do AI computation.

Currently, the main gripe for game studios in regards to seriously adopting DRL for a game project would likely be the general growing pains of DRL. As everything related to state-of-the-art DRL AI is so new, it's also hard to place expectations on the performance and viability of the solution without first investing serious effort into it. There is always the danger that later on when training a network it's realized that the AI is simply too unstable like in Barriga et al.'s [2] situation in scenarios with more than 3 opponents per side. Similarly, the AI might turn out to have some quirk that makes it unsuitable for built-in AI use, such as the odd edge crouching issue with Firoiu et al.'s [11] SSBM AI. As DRL and it's related technologies are evolving at such a fast pace, there's also the uncertainty of an implementation becoming obsolete if some major breakthrough or improvement is figured out later on. All this uncertainty does not play in DRL's favor when game studios consider the commercial viability of any prospective technologies when starting out a project.

In regards to the impact DRL could have for game AI, Barriga et al. [2] claimed that the ability to construct DL-based high-performance game AI systems to act against real players would revolutionize the video game industry. This makes sense, as in an ideal case developing an DRL game AI could be analogous to training a human to play a game and then asking them to perform as the enemy for other players. In terms of training, the training methodology for DRL networks can already resemble training a real human as seen in Barriga et al.'s [2] training methodology. In regards

to the AI's behaviour, the resulting AI's strategies also tend to already resemble human gameplay in both their ability and intuitivity as seen in all of the reviewed papers.

Such a simplistic way of building a game AI is a far cry from the very manual and labor intensive processes that traditional game AI development can entail, especially for more sophisticated ones. It also avoids the pitfalls that traditional AI can have where the AI needs to be carefully adjusted if game mechanics change, get added or are removed. In such a situation, an DRL AI would either be able to adapt without changing anything or at the worst case would need to be re-trained, which is a relatively repeatable and even automatable process assuming the game mechanics changes were not too groundbreaking.

With the above in mind, it's no wonder DRL and other methods in the ML space are so often seen as the future for AI in general. Considering the ideal game AI discussed, it certainly seems to apply to game AI as well. As seen in Barriga et al.'s [2] study, DRL based AI clearly have the capability to reach and exceed the level of traditional AI. It's simply a matter of adapting it to the role of an adjustable and suitably challenging opponent for real players. What that entails is waiting for these DRL methods and associated frameworks to mature even further and for consumer hardware to catch up to their requirements.

References

[1] Gustavo Andrade, Geber Ramalho, Hugo Santana, and Vincent Corruble. Extending reinforcement learning to provide dynamic game balancing. In *Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games, 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 7–12, 2005.

[2] Nicolas A Barriga, Marius Stanescu, Felipe Besoain, and Michael Buro. Improving rts game ai by supervised policy learning, tactical search, and deep reinforcement learning. *IEEE Computational Intelligence Magazine*, 14(3):8–18, 2019.

[3] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, Jun 2013.

[4] Tollmar Gisslen Quirin Hashme Shariq Hesse Chris Bergdahl, Gordillo et al. Augmenting automated game testing with deep reinforcement learning. 2020.

[5] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[6] Jonathan Boron and Chris Darken. Developing combat behavior through reinforcement learning in wargames and simulations. In *2020 IEEE Conference on Games (CoG)*, pages 728–731. IEEE, 2020.

[7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[8] Ciro Continisio and Alessia Nigretti. *UnityTechnologies/MachineLearningRoguelike: A small Roguelike game that uses Machine Learning to power its entities. Originally used in talks by Ciro Alessia.*, (accessed November 29, 2020). https://github.com/UnityTechnologies/MachineLearningRoguelike.

[9] Nvidia Corporation. *Tensor Cores | NVIDIA Developer*, (accessed November 29, 2020). https://developer.nvidia.com/tensor-cores.

[10] Aline Dobrovsky, Uwe M Borghoff, and Marko Hofmann. Improving adaptive gameplay in serious games through interactive deep reinforcement learning. In *Cognitive infocommunications, theory and applications*, pages 411–432. Springer, 2019.

[11] Vlad Firoiu, William F. Whitney, and Joshua B. Tenenbaum. Beating the world's best at super smash bros. with deep reinforcement learning. *CoRR*, abs/1702.06230, 2017.

[12] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.

[13] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[14] Alexey Grigorevich Ivakhnenko and Valentin Grigorevich Lapa. Cybernetics and forecasting techniques. 1967.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[16] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[18] Diego Perez-Liebana, Spyridon Samothrakis, Julian Togelius, Simon M Lucas, and Tom Schaul. General video game ai: Competition, challenges, and opportunities. 2016.

[19] Johannes Pfau, Antonios Liapis, Georg Volkmar, Georgios N Yannakakis, and Rainer Malaka. Dungeons & replicants: automated game balancing via deep player behavior modeling. In *2020 IEEE Conference on Games (CoG)*, pages 431–438. IEEE, 2020.

[20] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[21] Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games*, 10(3):257–270, 2018.

[22] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[23] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[24] Ruben Rodriguez Torrado, Philip Bontrager, Julian Togelius, Jialin Liu, and Diego Perez-Liebana. Deep reinforcement learning for general video game ai. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2018.

A Scalable Serving System for a Deep Neural Network

Giulio Marcon
giulio.marcon@aalto.fi

Tutor: Thanh-Phuong Pham

Abstract

Serving a Deep Neural Network (DNN) efficiently on a cluster of Graphical Processing Unit (GPU) is an important problem.
When we think about ML, we usually only think about the great models that we can now create. But when we want to take that amazing model and make it available to the world we need to think about all the things that a production solution requires, including scalability, consistency, modularity, and testability, as well as safety and security.

KEYWORDS: Machine Learning, Tensorflow, Serving, BentoML

1 Introduction

Consider a Reinforcement Learning agent that has to learn how to play the game of Pong just from the pixels that form a frame of the video-game, and has to analyze thousands of these images to successfully beat the opponent. The core computations of this workload are Deep Neural Network (DNN), which are networks of dense linear algebra computations where several layers of nodes are used to build up progressively more abstract

representations of the data. Graphical Processing Units (GPU) are specialized hardware accelerators for DNNs that have been used to solve the complex computations, and in the recent years Tensor Processing Units (TPUs) have emerged to further increase the computational power available.
A fundamental problem is therefore to distribute the large incoming workload onto the machines whilst improving the workflow of the data scientist that has to develop models that will be deployed into production.
This paper does not offer novel ML serving algorithms, but instead seeks to increase the community's awareness on the importance of designing an infrastructure that is able tackle the three key challenges of prediction serving: latency, throughput, and accuracy.

2 What is serving?

Serving is the process of applying machine learning models after they have been trained.
If we analyze what a typical workflow of a data scientist looks like, we will notice that the process of deploying models requires additional effort and attention, which results in them being distracted from their problem at hand. Apart from that, having many data scientists build and maintain their own serving solutions means that there may be a lot of duplicated effort.
All applications of machine learning depend mainly on two stages: training and inference.
Training is the process of building a model from data, which is often computationally expensive and requires multiple passes over potentially large datasets.
Inference is the process of using the model to make a prediction given an input, is typically part of user-facing applications, and must run in real-time, often on orders of magnitude more queries than during training.

3 Current state-of-the-art solutions of the problem.

Most of the solutions that try to tackle this problem are done using custom code developed by individual teams for specific use cases, leading to duplicated effort and fragile systems with high technical debt.

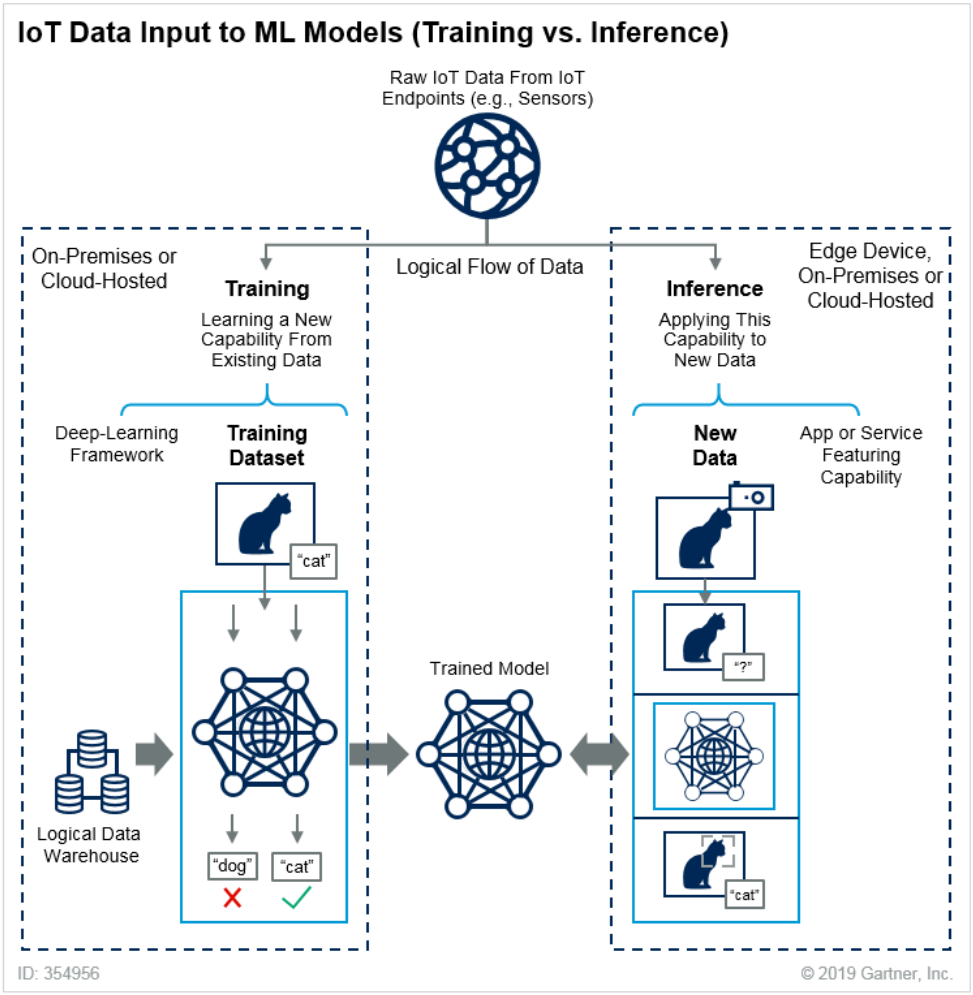


Figure 1. Inference is where capabilities learned during deep learning training are put to work. Source: [12]

Over the last few years, some companies have tried to develop an infrastructure that was able to successfully produce and deploy machine learning models, such as Clipper [14], a prediction serving system with a layered architecture that abstracts away the complexity associated with serving predictions, a set of novel techniques to reduce and bound latency while maximizing throughput, and a model selection layer that enables online model selection and composition to provide robust and accurate predictions for interactive application.

Another team developed Nexus [19], a scalable and efficient system that operates directly on models and GPUs, instead of serving the entire application in an opaque CPU-based container with models embedded in it, enabling several optimizations in batching and allowing more efficient resource allocation.

InferLine [16] was developed as a system which efficiently provisions prediction pipelines subject to end-to-end latency constraints by combining a

low-frequency Planner that finds cost-optimal configurations, with a high-frequency Tuner that rapidly re-scales pipelines to meet latency Service Level Objectives (SLOs) in response to changes in the query workload.
Other solutions include: TorchServe [11], designed specifically for PyTorch models, NVIDIA TensorRT [4], an SDK build to work the the company's own GPUs, and Microsoft Custom Decision Service [13] [5], which provides a cloud-based service for optimizing decisions using multi-armed bandit algorithms and reinforcement learning.

Unfortunately, most of them are now deprecated, and Google's TensorFlow Serving [17] and BentoML [1] seem to be the two most advanced ones currently available to the public.
In this seminar paper, I will compare the platforms mentioned above to test their performance while keeping in mind the advantages and disadvantages of each, and in the end, try to propose a solution that can improve the existing solutions.

4 Tensorflow Serving

Tensorflow Serving [10] is an open-source ML model serving project by Google. It aims to be a lexible, high-performance serving system, designed for production environments, that facilitates the deployment of new algorithms and experiments, while keeping the same server architecture and APIs. Tensorflow Serving provides out-of-the-box integration with Tensorflow [8] models, but can be easily extended to serve other types of models and data.
Some of the advantages of using this technology include:

- High performance. It has proven performance handling tens of millions of inferences per second at Google [9].

- High availability. It has a model versioning system to make sure there is always a healthy version being served while loading a new version into its memory

- Actively maintained by the developer community and backed by Google

5 BentoML

BentoML is an open-source platform for high-performance ML framework for serving, managing, and deploying machine learning models.
The main advantages of the project are:

- Ability to create basic API endpoints for serving trained models
- High-Performant online API for serving with adaptive micro-batching support.
- Provides support for all major machine learning training frameworks.
- Flexible deployment orchestration that follow DevOps best practices, such as Docker, Kubernetes, Kubeflow, Knative, AWS Lambda, SageMaker, Azure ML, and GCP.

6 Comparing the two platforms

The two frameworks have been compared using the Fashion MNIST dataset [2], a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples, often used to compare machine learning models, and used in both Tensorflow and BentoML's documentation.
Both frameworks, both written in Python, have some important differences:

- BentoML has multi-framework support, and is fully compatible with Tensorflow, PyTorch, Scikit-Learn, XGBoost, FastAI. Tensorflow-serving, on the other hand, only supports Tensorflow framework as of now, even though it can be adapted for other frameworks with some workarounds [6].
- Tensorflow loads the model from a tf.SavedModel format, which means that all the graphs and computations must be compiled into the SavedModel. BentoML keeps the Python run time in serving time, making it possible to do pre-processing and post-processing in serving endpoints.

- Tensorflow-serving can serve different versions of the same model, which can be useful when comparing changes during testing.

Some of the code used in the comparison can be found in this GitHub repository: [3].

7 Possible improvement

MLOps is an emerging field that aims to solve the technical debt that currently exists in machine learning systems [15].
They consist in a combination of philosophies and practices designed to enable data science and IT teams to rapidly develop, deploy, maintain, and scale out Machine Learning models. By following these guidelines, new versions of the models can be deployed into production constantly and reliably.
Important improvements could be done in the testing phase of the cycle shown id Figure 2. The validation of the trained model is still a tedious task that takes a lot of effort and is prone to error [18].
A specific testing support and methodology for detecting ML-specific errors needs to be established and followed in order to successfully tackle this problem.
Some of the important type of testing includes:

- Stress tests to ensure that the infrastructure can handle high volumes of data.

- Model staleness test, to check whether the trained model includes up-to-date data.

- Checking that the calculated metrics are satisfactory and improve the previous version of the same model, by measuring both loss metrics, fairness, and bias.

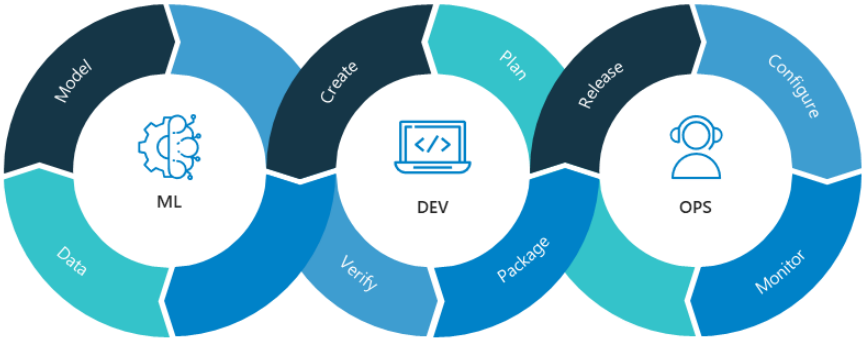


Figure 2. Proof of Concept MLOps cycle. Source: [7]

8 Conclusion

Concepts such as continuous delivery and integration, immutable infrastructure, and serverless computing, have been the focus of DevOps engineers in the last few years, and they can be adapted for the development, test, and serving of DNNs and ML models.
The so-called MLOps practises are still being designed and refined to this day, due to how innovative and current they are, but improvements can already be noticed, even if the practices are still in the early stages of development.

References

[1] Bentoml - the easiest way to build machine learning apis. https://github.com/bentoml/BentoML. Accessed: 29-10-2020.

[2] Fashion mnist - an mnist-like dataset of 70,000 28x28 labeled fashion images. https://www.kaggle.com/zalando-research/fashionmnist. Accessed: 29-11-2020.

[3] Github - giuliomarcon/a-scalable-serving-system-for-a-deep-neural-network. https://github.com/giuliomarcon/A-Scalable-Serving-System-for-a-Deep-Neural-Network. Accessed: 29-11-2020.

[4] Nvidia tensorrt - programmable inference accelerator. https://developer.nvidia.com/tensorrt. Accessed: 29-11-2020.

[5] Project custom decision. https://www.microsoft.com/en-us/research/project/custom-decision/. Accessed: 29-11-2020.

[6] Running pytorch models in production.
https://medium.com/styria-data-science-tech-blog/running-pytorch-models-in-production-fa09bebca622. Accessed: 29-11-2020.

[7] Running pytorch models in production.
https://nealanalytics.com/expertise/mlops/. Accessed: 29-11-2020.

[8] Tensorflow.
https://www.tensorflow.org. Accessed: 29-10-2020.

[9] Tensorflow extended.
https://www.tensorflow.org/tfx. Accessed: 29-10-2020.

[10] Tensorflow extended - serving models.
https://www.tensorflow.org/tfx/guide/serving. Accessed: 29-10-2020.

[11] Torchserve - a flexible and easy to use tool for serving pytorch models.
https://pytorch.org/serve/. Accessed: 29-11-2020.

[12] Training versus inference.
https://blogs.gartner.com/paul-debeasi/2019/02/14/training-versus-inference/. Accessed: 29-11-2020.

[13] Markus Cozowicz Luong Hoang John Langford Stephen Lee Jiaji Li Dan Melamed Gal Oshri Oswaldo Ribas Siddhartha Sen Alex Slivkins Alekh Agarwal, Sarah Bird. Making Contextual Decisions with Low Technical Debt. Technical report.

[14] G. Zhou M. J. Franklin J. E. Gonzalez I. Stoica D. Crankshaw, X. Wang. Clipper: A Low-Latency Online Prediction Serving System. Technical report.

[15] Daniel Golovin Eugene Davydov Todd Phillips Dietmar Ebner Vinay Chaudhary Michael Young Jean-Francois Crespo Dan Denniso D. Sculley, Gary Holt. Hidden Technical Debt in Machine Learning Systems. Technical report.

[16] Simon Mo Corey Zumar Joseph E. Gonzalez Ion Stoica Alexey Tumanov Daniel Crankshaw, Gur-Eyal Sela. InferLine: ML Prediction Pipeline Provisioning and Management for Tight Latency Objectives. Technical report.

[17] Heng-Tze Cheng Noah Fiedel Chuan Yu Foo Zakaria Haque Salem Haykal Mustafa Ispir Vihan Jain Levent Koc Chiu Yuen Koo Lukasz Lew Clemens Mewald Akshay Naresh Modi Neoklis Polyzotis Sukriti Ramesh Sudip Roy Steven Euijong Whang Martin Wicke Jarek Wilkiewicz Xin Zhang Martin Zinkevich Denis Baylor, Eric Breck. TFX: A TensorFlow-Based Production-Scale Machine Learning Platform. Technical report, Google Inc., 2017.

[18] Eric Nielsen Michael Salib D. Sculley Eric Breck, Shanqing Cai. The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction. Technical report, Google Inc.

[19] Y. Jin L. Zhao B. Kong M. Philipose A. Krishnamurthy R. Sundaram H. Shen, L. Chen. Nexus: A GPU Cluster Engine for Accelerating Neural Networks Based Video Analysis. Technical report.

Deep Learning for Electroencephalography (EEG) classification in Virtual Reality (VR)

Thomas Weikert

Tutor: Matti Siekkinen

Abstract

Electroencephalography (EEG) classification became important in commercial applications, such as virtual reality (VR). In practice, Machine Learning algorithms have been commonly deployed, but given the enhanced accessibility of large EEG datasets and the current trend in Deep Learning, attention starts to shift towards the applicability of Deep Learning algorithms. To pave the way for commercialisation of EEG analysis in VR, a robust and fully automatized approach that is independent from domain experts is required. This paper provides an analysis of EEG classification tasks with Deep Learning algorithms, its underlying architecture, and specific input formulations for training purpose. Reviewed paper used Deep Learning methods for EEG classification in motor imagery, mental workload, cyber sickness and emotional responses. Regardless of the underlying task type, Convolutional Neural Networks (CNN) has been the most popular design choice among reviewed papers, followed by Multilayer Perceptrons (MLP). For both design choices, a description of each specific input formulation, major characteristics, and end classifier has been performed.

KEYWORDS: Virtual Reality, Electroencephalography, Deep Learning

1 Introduction

In recent years, virtual reality (VR) has been declared as a mainstream technology [1], and public interest has been increasing as VR technology offers opportunities for various commercial, research and industrial applications. Current VR headsets provide the incorporation of electroencephalography (EEG) sensors. Towards the use of EEG signals within applications, an automatic classification is required without dependence on domain experts.

In the last decade, classic Machine Learning algorithms have been adopted for EEG analysis. Such procedure contains artefact removal, feature engineering and classification for task solving. As Subasi, Abdulhamit and Gursoy and Ismail [2] stated, independent component analysis (ICA) is a popular method for artifact removal. A common method for feature extraction is the use of time-frequency transformation, such as wavelet transformation. Following extraction, features can be passed as input for classic Machine Learning algorithms, such as a Support Vector Machine (SVM). Nevertheless, these methods come with certain disadvantages as stated by Yue and Wang [3]. First, the accuracy mainly relies on the choice of feature extraction methods. Second, the process of feature engineering requires the knowledge of domain experts. Third, the overall performance of classic Machine Learning algorithm is not satisfying, although features were correctly identified. As large EEG datasets became more accessible and high performance graphic processing units were developed, attention has been drawn towards Deep Learning. The performance achieved with Deep Learning exceeded those achieved with Machine Learning algorithms.

As such, this paper endeavoured to investigate how Deep Learning may be used for EEG classification in a VR domain, and provide examples how these signals can be used to model VR user experience. Therefore, this paper investigates which EEG classification tasks with Deep Learning has been explored in VR and reviewed their EEG pre-processing methods, architecture and performance. The organization of this paper is as follows. The background section introduces the use of EEG in VR, reviews classic Machine Learning approaches and recent Deep Learning methods for EEG classification. The following section outlines signal-acquisition and

data pre-processing. Next, the research highlights will be laid out dealing with Deep Learning architecture trends, followed by a discussion on design choices and its usage in VR. The conclusion is provided in the last section.

2 Background

This section reviews the use of EEG in VR and earlier work in machine and Deep Learning in terms of EEG classification.

2.1 Use of EEG in VR

The application of EEG in VR devices has been examined over a couple of years within research related to brain computer interfaces (BCI). Focus has been on interaction monitoring, user behaviour and learning environment. Using EEG, the hope was to design a VR experience in which the environment adapts to the user workload. For example, the use of BCIs in VR has been investigated to achieve avatar control with EEG or to transform the shape of the Avatar [4, 5]. Also, NASA scientists used EEG to enhance users vigilance and composure during gameplay [6]. Other research has been carried out to adapt games level of difficulty based on changes in EEG signals [7] or to use EEG signals to keep track of items that were detected by the user [8]. Nevertheless, major challenges must be addressed for BCIs in order to grow into an established technology for VR applications as stated by Lecuyer et al. [9]. Nowadays, companies such as Neureble and Looxid are developing VR HMDs that come with integraded EEG electrodes created for BCI, illustrating the successful integration into broader EEG-based devices. Successfull progress has also been made regarding EEG to measure user response. Abdessalem and Frasson developed a VR game for neurological feedback in real-time, classifying player's emotional response to adapt scenarios accordingly [10]. Other researchers have used EEG in a clinical environment using mental workload of users with Autism in a VR driving scenario [11]. To design a training adaptive system, EEG has been used by Gerry et al. [7] to monitor mental load during a visual search task. Although this was an important step forward, there has been little research on the type of tasks and procedures that should be used for EEG classification in a VR scenario.

2.2 Classic Machine Learning

As Karácsony et al. [12] stated, the majority of current BCI-VR systems implement Machine Learning (ML) methods to build classifiers, which involves signal pre-processing, feature extraction and classifiaction steps.

Signal pre-processing helps to reduce noise in EEG data and make the underlying dataset ready for feature engineering. Within this necessary step, various combination of filter technique are being used. To reduce noise coming from differences in electrical activity between electrodes, referencing methods have been applied, such as spatial filter algorithms like the Laplacian-filter or the Common Average Reference (CAR). For illustration purposes, figure 1 shows the principle of several spatial filter algorithm for electrode position C3. On the very left side the C3 electrode is shown with a reference electrode on the earlobe. Next to it, the CAR method is used, which subtracts the mean value of all electrodes from C3. In the middle of the figure the principle of the small Laplacian filter is shown in which the value of C3 is subtracted from the value of the average surrounding electrodes. The principle is repeated with the nearest surrounding electrode in the large Laplacian filter on the right. On the very right side the principle of bipolar derivation is shown, in which the signal from the one electrode is subtracted to the nearest one.

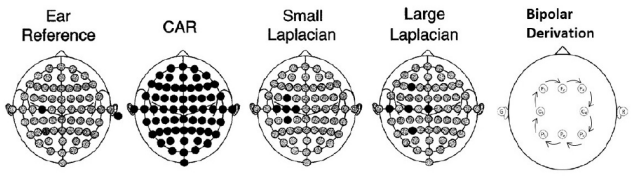


Figure 1. Different approaches of spatial filter algorithm adopted from [13].

As cited by Tremmel [14] most common classification algorithm in BCIs are linear classifiers, such as linear discriminant analysis (LDA) or support vector machines (SVM). Other typical approaches include the use of KNN and tree-based classifiers, achieving results ranging between 65-80% [1].

2.3 Deep Learning

Attention in Deep Learning has not been directly drawn to applications in neural classification given a lack of practicality by a very long computation time and problems with gradient descent algorithms [15]. With re-

cent development in computational resources neural networks became appealing, offering a powerful and inexpensive solution and leading to an enhanced interest and applications of Deep Learning in the past years. [16]. Recently, significant improvements in performance by using Deep Learning techniques can be observed in areas, such as images, videos, speech, and text. As Deep Learning techniques require less domain knowledge on the data due to its automatic optimization of it parameters [16], applications in medical imaging were adopted, requiring traditionally expert knowledge for interpretation [17]. With the appearance of accessible EEG data sets online, Deep Learning techniques started to get applied to decoding and classification of EEG signals that is conventionally correlated to a low signal to noise ratios and high dimensional data. Regardless of recent developments current research on BCI systems applying Deep Learning is very limited and mostly involves non-VR settings, thus further exploration of this particular field is required, including optimal preprocessing methods and architecture design.

2.4 Summary

This section shows that EEG has been successfully used as a tool in adaptive scenarios for VR. Current research in EEG classification shifts from classic Machine Learning methods towards Deep Learning. However, there is limited research available using EEG in VR settings with Deep Learning.

3 Measuring EEG

EEG signals contain several overlapping frequencies that can be separated by signal processing techniques. The frequencies can be separated into different bands, these are usually the bands of delta (2-4Hz), theta (4-8Hz), alpha (8-12Hz), beta (15-30Hz), lower gamma (30-80Hz) and upper gamma (80-150Hz) [18]. This section describes the overall approach adopted in conducting EEG analysis consisting of signal acquisition and pre-processing steps, each of will be explained in the following sections below.

3.1 Signal Acquisition

As stated by Tremmel [14], electrodes EEG signal tracking record usually in a bandwidth of 128-512 Hz, measuring the voltage between scalp and electrode, which are systematically placed in the target region of the scalp. In order to compensate for interfering signals, such as those caused by the heartbeat, reference electrodes are placed on the head, which are normally placed in expected inactive places, such as the earlobe of the forehead. It is necessary to fulfill the Nyquist frequency criterion when recording EEG signals, which requires a sampling rate twice as high as the expected frequency of the observed signal [14].

The most popular approach to place electrodes on the scalp is the international standard 10-20 system [19, 20]. The name of the system refers to the space of the adjacent electrodes, where the distance respectively 10% or 20% of the total right-left or front-back distance of the skull is covered [14]. Electrodes for workload estimation are usual placed in the premotor, prefrontal and parietal brain areas, but vary with respect to the underlying task. [21].

3.2 Filtering

A common pre-processing step is filtering, which is necessary, if feature classification is based on time frequency domain. As stated by [22] a widely used filter is a low-pass filter, that passes low frequencies and attenuate high frequencies. The corresponding opposite is a high-pass filter, which passes high frequencies and attenuate low frequencies. In order to pass an intermediate range of frequencies, a band-pass filter is used. Also, a notch filter has been applied in [7], that attenuate only a narrow band of frequencies. In case feature classification is processed in the frequency domain, the undesirable frequency band can be ignored by not considering them as characteristic's for classification. The majority of reviewed studies applies frequency domain filters to process the bandwidth of the aspired EEG signal. A thorough search of the relevant literature yielded no article, which investigates weather the usage of raw EEG signals is able to achieve comparable results.

3.3 Artefact correction

EEG data is naturally noisy and can be caused by unwanted muscle tensions, like eye-blinks or body part movements, which challenges the usability of EEG in the context of VR. Artefact correction remains one of the biggest challenges in the analysis of EEG signals in VR, as it allows users to move freely and interact with their virtual environment [14]. To bridge this problem, VR controllers, with additional tracking sensors to record body movements, offer a pragmatic method that can be used to reduce motion induced contamination in EEG as previously achieved in gait related EEG research [23]. As stated in [22], the most frequent artifact removal algorithms used were independent component analysis (ICA) and discrete wavelet transformation (DWT). Methods for artefact correction has been widely investigated by a number of studies in the past, and will not be repeated in depth within this paper due to the limited scope.

3.4 Feature Extraction

Feature Extraction is the most important prerequisites to build reliable classifiers and requires domain knowledge and the right techniques. In the case of cognitive workload estimation, feature extraction techniques create power band features of the epochs in the desired frequency bands. Reviewed paper used Fast Fourier Transform (FFT) or Welch's method. FFT transforms the data by projecting it onto sinusoidal basis function which shifts the signal from a time domain to a frequency domain. In practice FFT is widely used, but it comes with some disadvantages, as the potential loss of temporal information from data due to stretching into sine waves [24]. Moreover, the same window size is used to calculate power in different frequencies, although higher precision can be achieved by using different windows size for low and high frequencies [25]. The importance of the window size is emphasized by [26], stating that a non-optimal width of the window causes poor frequency if too narrow, or poor time localization that violates the stationary assumption, if too wide. Another very widely used method to gain a spectral estimation of the signal is the Welch method, which is a method that calculates a periodogram for windowed sections of data using FFT and then averaging these windows to reduce the variance of the estimate.

4 Deep Learning algorithms

Craik et al. [22] clusters EEG classification techniques with Deep Learning into the following categories: Convolutional Neural Networks, Deep Belief Networks, Recurrent Neural Networks, Stacked Auto-Encoders, Multi-layer Perceptron Neural Network and hybrid architectures. For the understanding of the reviewed articles, this sections briefly introduces two techniques used for EEG classification with Deep Learning.

4.1 Multilayer Perceptrons (MLP)

Multilayer Perceptrons, also called Feed Forward Neural Networks, is a subclass of Artificial Neural Networks (ANN). The neurons within an ANN can be interconnected arbitrarily in principle. MLPs are ANNs where the neurons are grouped into consecutive groups or 'layers' and only connections between neurons in consecutive layers are allowed (with some exceptions such as 'skip connections') [27]. They are called feedforward as an information passes through the function being assessed from x, through the intermediate computations used to define f, and at last to the output y [27]. Each MLP is a network of interconnected units, which are defined as neurons. The input of a neuron is a weighted sum, which gets activated through a non-linear activation function. Each layer consists of several individual neurons, whose input is taken from the output of the previous layer. The layers are piled up on each other to embrace a more complex MLP. Figure 2 illustrates a MLP with two hidden layers enclosed by an input and output layer. The word 'hidden' refers to the fact that neurons in hidden layers are not directly accessible. The input layer takes input features, which gets activated by using an activation function. In contrast to the other layers, it does not contain a non-linear activation function. The illustrated example has three feature values. The following two hidden layers are called fully connected, densely connected layer or dense layer. Finally, the output layer consists of three neurons, which are representative for three class classification. Its value represent its confidence in assigning this data point to its label.

4.2 Convolutional Neural Networks (CNN)

CNNs is a specialized kind of neural network for processing data inspired by the human visual cortex [28]. Consisting of several layers, a CNN is

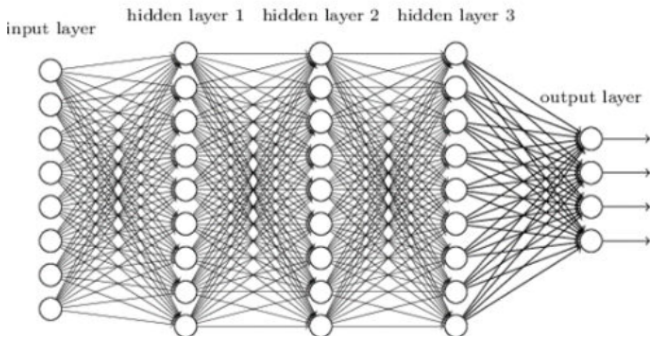


Figure 2. Illustration of a Deep Neural Network [**?**]

characterized by a varying number of convolutional and pooling layers, with a fully connected layer placed at the output. In general, Convolution can be done in two or three dimensions, e.g. for images and video, however to analyse raw EEG data, a one-dimensional Convolution is needed. One refers to raw EEG data as the time domain. The most important building block is the Convolutional Layer, that performes covolutional operations among the image and the kernels, computing the weighted sum of the patch of the image. Pooling layers aim to down sample feature maps [27]. Another layer is a fully-connected (FC) layer, which gets the feature map from a previous convolutional or pooling layer as an input, flattened as a single vector of values. The FC layer has the same mathematical operations as an ANN [27]. An exemplary CNN architecture is illustrated in Figure 3. CNN framework choices contains kernels regulated using back-propagation algorithm. Due to a multi folded feature extraction of different layers and filters, CNN features are robust to spatial translation [29] and make analysis for task solving possible, such as the correct representation of cognitive states over different brain zones.

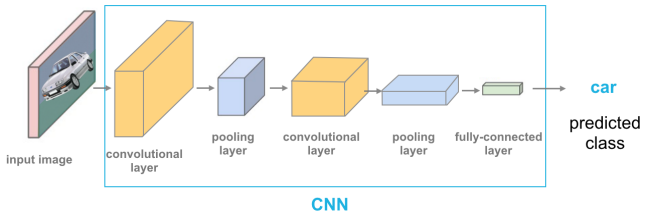


Figure 3. Illustration of a Convolutional Neural Network [30].

5 Research Highlights

This section first details the EEG classification tasks with Deep Learning found within reviewed papers. Then, the input formulations, and archi-

tecture trends are analyzed.

5.1 EEG classification tasks

EEG classification tasks that have been examined with Deep Learning can be distinguished in three categories. The classification of emotion recognition, motor imagery, and mental workload. Emotion recognition tasks requires subjects to watch video clips, which have been labeled with a particular emotion by an expert prior to the experiment. Meanwhile EEG signals were recorded, which allows an assessment on emotion recognition. In virtual reality, a better understanding of the user's emotion will help to determine, if a particular movement was the intended movement and help computers to enhance their understanding of the user's emotional status, which enables further applications.Motor imagery tasks are related to intended user's movements, such as the limb or the tongue, and are often in the context of stroke treatment. Mental workload classification involves EEG analysis while the subject is in the process of task solving with variant level of difficulty. Many studies aim to perform mental workload classification in the context of driving simulation studies [31], live pilot studies [32], and responsibility tasks [33]. Based on statistical behaviour of pilots and drivers, such as reaction time and path deviation, mental workload has been classified. Conversely, various studies classified mental workload for responsibility tasks, for subjects dealing with an increasing number of actions the subject was responsible for.

5.2 Deep Learning Architecture Trends

This section reviews used EEG classification algorithm with Deep Learning in the context of VR and its most prominent design frameworks, then analyzes its characteristics and compare its results.

Looking at Architecture Design Choices, the center of attention is on outlining the tendency in creation of specific Deep Learning architectures used for VR applications, principally its most important characteristic and end classifier. Among the reviewed papers, the most prominent design framework has been CNNs. Its most important design choice is the amount of different layers and kind of end-classifier. The second identified design framework has been MLPs are composed by a number of

layers and a variety of neurons per layer. With regard to Activation Functions, across all studies, rectified linear unit (Relu) has been deployed as an activation function unrelated to the initial architecture design choice. In combination to the Relu activation function the Softmax and Sigmoid activation function has been used, too. Activation functions for fully-connected layers can be assembled in subsections, which are non-classifier fully-connected and fully-connected layers. While the most popular non-classifier fully-connected applied the Sigmoid activation function, the fully-connected layers deployed a Softmax activation function. Looking at the Input formulation by Deep Learning architecture, across all studies the input formulation differ considerably. In general three types of input formulation has been identified: Signal values, calculating features and images. While CNN studies utilized all three types of input formulation, MLP were distributed only among signal values and calculated features. Over all studies and architecture types a preferred choice was towards calculating features. In terms of accuracy, CNN studies which had images as input were not differ from studies that deployed feature calculation as inputs. Both input formulation achieved accuracy over 90% In contrast, raw signals were performing less good and achieving only an accuracy of 74%.

6 Discussion

The following sections derives recommendations for design decisions on Deep Learning architectures for EEG classification tasks and usage in VR. Recommendations are based on the reviewed papers and relate to classification algorithm, its input formulation and practicality for its use in VR. Recommendations are not given by tasks as the number of studies was too low, which is also the main limitation of this paper.

In general a comparison among different architecture design choices, its accuracy's and underlying EEG data sets comes with difficulties. Moreover, most studies were designed for different task solving, which effects the choice of design or input processing. Nevertheless the provided analysis can assist and encourage future research to make use of Deep Learning methods in the context of VR and a variety of tasks. When using CNN for task solving, the use of signal values or images comes with advantages. The majority of the studies used the maximum available chan-

nels, supporting the assumption that CNN are capable of handling high dimensional EEG data and size of data sets better compared to other algorithms as stated by Craik et al. [22]. Using images as CNN's input, spectograms were the preferred choice, achieving the highest accuracy. For signal values, the number of convolutional layers vary from three to six layers. When using MLP for task solving, signal values is the preferred choice. Similar to CNN, the channels were not limited. As end classifier, a single dense layer is recommended, which is the most popular choice in most studies. For the final fully-connected layer in all studies, the softmax activation function is the preferred choice, while for fully-connected non classifier layers, the sigmoid activation function is a used.

Despite difficulties of a general comparison, a tendency of a superior performance of Deep Learning strategies compared to classic Machine Learning algorithms has been found. For exemplary illustration of the performance, Table 1 summarizes work on the Physionet EEG Dataset [34]. The created CNN classifiers performance surpassed classic Machine Learning algorithms. Therefore, Deep Learning models should be the prefered choice, when it comes to the highest accuracy. Apart from this, Deep Learning algorithm present an other advantage over classic Machine Learning algorithm as they have the ability to automatically detect features, and thus, do not require cumbersome feature engineering or domain expert.

Table 1. Overview of works performing classification tasks on the Physionet [34] EEG dataset.

Study	#Channels	Max. accuracy	Methods
[35]	16	63.62 %	SVM
[36]	3	68.21 %	Wavelet transform Feed-forward MLP
[37]	9	71.55 %	Phase information
[38]	58	72.55 %	SVM
[39]	14	80.05 %	Random forest
[40]	64	86.13 %	CNN
[40]	14	82.66 %	CNN
[12]	64	88.50 %	CNN
[12]	16	84.13 %	CNN

With regard to the implications for VR technology and EEG classification tasks, reviewed articles illustrated the potential of combining them. This research is representative for the current desire to enhance immersiveness and improve the usability of VR technology. The spectrum of differ-

ent tasks, setups and experiments is high, which is also reflected in the variety of architectures for Deep Learning classifiers.

7 Conclusion

Researcher in the domain of VR have successfully adopted Deep Learning methods for EEG classification. The implementation ranges over different tasks, including mental workload, motor imagery, emotional responsive and cyber sickness. Among all tasks a popular choice of design framework are CNNs, followed by MLPs. Significant differences occur with respect to input formulization, whereas CNN's performed best when using signal values or spectrogram as inputs. Further research is encouraged on extending the number of studies, the implementation of other methods in the domain of VR, such as hybrid networks, and the assessment of raw EEG signals, as this has not been sufficiently evaluated.

Acknowledgements

I am grateful to Matti Siekkinen, my tutor for the course CS-E4000 Seminar in Computer Science at Aalto University, for giving me the opportunity to write this paper, and for providing me with feedback throughout the semester.

Appendix

An overview of reviewed papers is laid out in the appendix. While Table 2 summarizes methods used for pre-processing and signal acquisition, Table 3 gives an overview on architecture trends for EEG classification tasks in the domain of VR.

Table 2. Signal Acquisition and Pre-Processing Methods

Study	**Deep Learning Strategy**	**Signal Acquisition**			**Pre-Processing**	**Input formulation**	**Accuracy**
		EEG Device	Channel strategy	Electrode Positions			
[3]	CNN 6 conv 1 FC OUT (3)	Synamps 2 system	Total 34 30xEEG 4xEOG	Reference: A1 Rest: n.a.	Notch filter CAR Raw	19x1200x30x375 (subjects x trials x channels x timing samples)	74.00%
[12]	CNN 4 conv 6 FC OUT (2,3,4)	n.a.	All(64)	n.a.	6th order Butterworth BP Normalization	n.a.	90.14% 89.86% 77.71%
[12]	CNN 4 conv 6 FC OUT (2,3,4)	n.a.	All(16)	FC3, FCz, FC5, C1-C6, Cz, CP3, CPZ, CP4, P3, Pz and P4	6th order Butterworth BP Normalization	n.a.	75.40% 80.81% 61.63%
[41]	CNN 1 FC OUT (2)	Looxidlabs VR	All(6)	n.a.	Notch filter STFT	n.a.	82.00%
[42]	CNN 3 conv 1 FC 0.25, 0.5 DR OUT (2)	Emotiv Epoc+	Total 84 14xRaw EEG 70xBand power data	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6,F4, F8, and AF4	Normalization Filtering	128x84x1	98.82%
[42]	MLP 3 hid. (128,256,128) Um 0,5 DR OUT (2)	Emotiv Epoc+	Total 84 14xRaw EEG 70xBand power data	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6,F4, F8, and AF4	Normalization Filtering	Input Dense layer (84)	98.02%
[1]	MLP 2 hid. 10 Un 0.1 DR	Muse headband	Total 5 4xEEG 1xReference	TP9, AF7, AF8, TP10, Fpz	FFT Butterworth's 4th Order Filter	n.a.	88.07%
[1]	MLP 2 hid. 200 Un 0.1 DR	Muse headband	Total 5 4xEEG 1xReference	TP9, AF7, AF8, TP10, Fpz	FFT Butterworth's 4th Order Filter	n.a.	92.31%
[1]	MLP 2 hid. (2x200) Un 0.7 DR	Muse headband	Total 5 4xEEG 1xReference	TP9, AF7, AF8, TP10, Fpz	FFT Butterworth's 4th Order Filter	n.a.	96.32%

Table 3. Deep Learning Architecture Designs

Study	Deep Learning Strategy	Activation function	Input formation	Frequency range	Artifact strategy	Task information	Accuracy
[3]	CNN 6 conv 1 FC OUT (3)	Relu	Raw All (30)	1 - 100 Hz	AR	ER 20 subj.	74.00%
[12]	CNN 4 conv 6 FC OUT (2,3,4)	Relu (FC) SoftMax (FC)	N All (64)	0.5 - 75 Hz	none	MI 105 subj.	90.14% 89.86% 77.71%
[12]	CNN 4 conv 6 FC OUT (2,3,4)	Relu (FC) SoftMax (FC)	N All (16)	0.5 - 75 Hz	none	MI 105 subj.	75.40% 80.81% 61.63%
[41]	CNN 1 FC OUT (2)	Relu	Spect. All(6)	55 - 65 Hz	AR	ML 9 subj.	82.00%
[42]	CNN 3 conv 1 FC 0.25, 0.5 DR OUT (2)	Relu / Sigmoid	N / All(84)	128 Hz	AR	CS 24 subj.	98.82%
[42]	MLP 3 hid. (128,256,128) Um 0,5 DR OUT (2)	Relu / Sigmoid	N / All(84)	4-45 Hz	AR	CS 24 subj.	98.02%
[1]	MLP 2 hid. 10 Un 0.1 DR	Relu / Softmax	All(4)	N.A.	AR	ER 24 subj.	88.07%
[1]	MLP 2 hid. 200 Un 0.1 DR	Relu / Softmax	All(4)	N.A.	AR	ER 24 subj.	92.31%
[1]	MLP 2 hid. (2x200) Un 0.7 DR	Relu / Softmax	All(4)	N.A.	AR	ER 24 subj.	96.32%

References

[1] Jason Teo and Jia Tian Chia. Deep neural classifiers for eeg-based emotion recognition in immersive environments. In *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, pages 1–6. IEEE, 2018.

[2] Abdulhamit Subasi and M Ismail Gursoy. Eeg signal classification using pca, ica, lda and support vector machines. *Expert systems with applications*, 37(12):8659–8666, 2010.

[3] Kang Yue and Danli Wang. Eeg-based 3d visual fatigue evaluation using cnn. *Electronics*, 8(11):1208, 2019.

[4] Liu Mingyu, Wang Jue, Yan Nan, and Yang Qin. Development of eeg biofeedback system based on virtual reality environment. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 5362–5364. IEEE, 2006.

[5] Ilkka Kosunen, Mikko Salminen, Simo Järvelä, Antti Ruonala, Niklas Ravaja, and Giulio Jacucci. Relaworld: neuroadaptive and immersive virtual reality meditation system. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 208–217, 2016.

[6] Game and simulation control - control modification through user physiological state. https://ntts-prod.s3.amazonaws.com/t2p/prod/t2media/tops/pdf/LAR-TOPS-88.pdf. Accessed: 2020-10-27.

[7] Lynda Gerry, Barrett Ens, Adam Drogemuller, Bruce Thomas, and Mark Billinghurst. Levity: A virtual reality system that responds to cognitive load. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2018.

[8] Thorsten O Zander, Matti Gaertner, Christian Kothe, and Roman Vilimek. Combining eye gaze input with a brain–computer interface for touchless human–computer interaction. *Intl. Journal of Human–Computer Interaction*, 27(1):38–51, 2010.

[9] Anatole Lécuyer, Fabien Lotte, Richard B Reilly, Robert Leeb, Michitaka Hirose, and Mel Slater. Brain-computer interfaces, virtual reality, and videogames. *Computer*, 41(10):66–72, 2008.

[10] Hamdi Ben Abdessalem and Claude Frasson. Real-time brain assessment for adaptive virtual reality game: A neurofeedback approach. In *international conference on brain function assessment in learning*, pages 133–143. Springer, 2017.

[11] Lian Zhang, Joshua Wade, Dayi Bian, Jing Fan, Amy Swanson, Amy Weitlauf, Zachary Warren, and Nilanjan Sarkar. Cognitive load measurement in a virtual reality-based driving system for autism intervention. *IEEE transactions on affective computing*, 8(2):176–189, 2017.

[12] Tamás Karácsony, John Paulin Hansen, Helle Klingenberg Iversen, and Sadasivan Puthusserypady. Brain computer interface for neuro-rehabilitation with deep learning classification and virtual reality feedback. In *Proceedings of the 10th Augmented Human International Conference 2019*, pages 1–8, 2019.

[13] Dennis J McFarland, Lynn M McCane, Stephen V David, and Jonathan R Wolpaw. Spatial filter selection for eeg-based communication. *Electroencephalography and clinical Neurophysiology*, 103(3):386–394, 1997.

[14] Christoph Tremmel. Estimating cognitive workload in an interactive virtual reality environment using electrophysiological and kinematic activity. 2019.

[15] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[17] Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.

[18] Jean Gotman. A few thoughts on "what is a seizure?". *Epilepsy & Behavior*, 22:S2–S3, 2011.

[19] Herbert H Jasper. The ten-twenty electrode system of the international federation. *Electroencephalogr. Clin. Neurophysiol.*, 10:370–375, 1958.

[20] Frank Sharbrough. American electroencephalographic society guidelines for standard electrode position nomenclature. *J clin Neurophysiol*, 8:200–202, 1991.

[21] Adrian M Owen, Kathryn M McMillan, Angela R Laird, and Ed Bullmore. N-back working memory paradigm: A meta-analysis of normative functional neuroimaging studies. *Human brain mapping*, 25(1):46–59, 2005.

[22] Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. Deep learning for electroencephalogram (eeg) classification tasks: a review. *Journal of neural engineering*, 16(3):031001, 2019.

[23] Kristine L Snyder, Julia E Kline, Helen J Huang, and Daniel P Ferris. Independent component analysis of gait-related movement artifact recorded using eeg electrodes during treadmill walking. *Frontiers in human neuroscience*, 9:639, 2015.

[24] Luck SJ. Setting up and running an erp lab. *An introduction to the event-related potential technique*, pages 16–1, 2014.

[25] Steven J Luck. *An introduction to the event-related potential technique*. MIT press, 2014.

[26] Andrea Varsavsky, Iven Mareels, and Mark Cook. *Epileptic seizures and the EEG: measurement, models, detection and prediction*. Taylor & Francis, 2011.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[28] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

[29] Kai Kang and Xiaogang Wang. Fully convolutional neural networks for crowd segmentation. *arXiv preprint arXiv:1411.4464*, 2014.

[30] Cezanne Camacho. Convolutional neural networks, June 2018. [Online; posted 03-June-2018].

[31] Mohammad A Almogbel, Anh H Dang, and Wataru Kameyama. Cognitive workload detection from raw eeg-signals of vehicle driver using deep learning. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pages 1–6. IEEE, 2019.

[32] Feng Li, Guangfan Zhang, Wei Wang, Roger Xu, Tom Schnell, Jonathan Wen, Frederic McKenzie, and Jiang Li. Deep models for engagement assessment with scarce label information. *IEEE Transactions on Human-Machine Systems*, 47(4):598–605, 2016.

[33] Zhong Yin and Jianhua Zhang. Cross-session classification of mental workload levels using eeg and an adaptive deep learning model. *Biomedical Signal Processing and Control*, 33:30–47, 2017.

[34] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

[35] Vikram Shenoy Handiru and Vinod A Prasad. Optimized bi-objective eeg channel selection and cross-subject generalization with brain–computer interfaces. *IEEE Transactions on Human-Machine Systems*, 46(6):777–786, 2016.

[36] Martina Tolić and Franjo Jović. Classification of wavelet transformed eeg signals with neural network for imagined mental and motor tasks. *Kinesiology: International journal of fundamental and applied kinesiology*, 45(1):130–138, 2013.

[37] Ana Loboda, Alexandra Margineanu, Gabriela Rotariu, and Anca Mihaela Lazar. Discrimination of eeg-based motor imagery tasks by means of a simple phase information method. *International Journal of Advanced Research in Artificial Intelligence*, 3(10), 2014.

[38] Cheolsoo Park, Clive Cheong Took, and Danilo P Mandic. Augmented complex common spatial patterns for classification of noncircular eeg from motor imagery tasks. *IEEE Transactions on neural systems and rehabilitation engineering*, 22(1):1–10, 2013.

[39] Youngjoo Kim, Jiwoo Ryu, Ko Keun Kim, Clive C Took, Danilo P Mandic, and Cheolsoo Park. Motor imagery classification using mu and beta rhythms of eeg with strong uncorrelating transform based complex common spatial patterns. *Computational intelligence and neuroscience*, 2016, 2016.

[40] Jingnian Chen, Houkuan Huang, Shengfeng Tian, and Youli Qu. Feature selection for text classification with naïve bayes. *Expert Systems with Applications*, 36(3):5432–5435, 2009.

[41] Dongjae Kim, Jiseong Park, Jeongseok Hwang, Wan Hee Cho, and Sang Wan Lee. Decoding prefrontal cognitive states from electroencephalography in virtual reality environment. *2020 8th International Winter Conference on Brain Computer Interface (BCI)*, pages 1–3, 2020.

[42] Daekyo Jeong, Sangbong Yoo, and Jang Yun. Cybersickness analysis with eeg using deep learning algorithms. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 827–835. IEEE, 2019.

Current Use Cases and Implementations of Multi-user XR

Péter Dános
peter.danos@aalto.fi

Tutor: Esa Vikberg

Abstract

Multi-user Extended Reality is the collection of hardware and software systems that enable its users to share, or participate in some sort of virtual environment. To better understand the state of the technology and what it is capable of, this paper looks at current solutions and discusses its potential. The paper introduces the background of the technologies used in the field of extended reality, and gives an overview of its categories. Then the paper looks at relevant papers published in recent years to show use cases, hardware and software solutions, and the industries that can take advantage of the technology. This is followed by a discussion of the benefits, potential, and disadvantages of these solutions and multi-user extended reality in general. Finally the paper is concluded with a summary of the current state and possible future of the technology.

KEYWORDS: XR, VR, AR, MR, extended reality, virtual reality, augmented reality, mixed reality, multi-user, virtual environments

1 Introduction

Extended Reality (XR) is becoming more and more widely used [1], as the technologies enabling it mature. XR is an umbrella term, which includes three distinct categories of technologies: Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (VR).

Virtual and mixed reality usually require the user some sort of head-mounted device that has a display in front of the user's eyes. Augmented reality can also be used this way, but there are more accessible alternatives for it, for example a smartphone. With the rise of extended reality came the need for multi-user supported applications. This paper is a much needed overview of the state of the technology today, its uses, benefits and problems. First, the differences and similarities of the different categories of extended reality are presented (section 2). This is followed by a literature review of scientific articles, discussing different use-cases and applications of the technology (section 3). The discussion section looks at the problems with current applications as well as the benefits and the general potential in the technology (section 4). Finally, I summarize my findings and leave you with my personal conclusion (section 5).

2 The types of XR

The expression extended reality is an umbrella term that includes three different categories of technologies, with the goal of constructing and displaying some sort of virtual or augmented environments using computer graphics. The three categories are: Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR).

2.1 Virtual Reality

In **Virtual Reality (VR)** the user wears a Head Mounted Device (HMD), which has a screen and appropriate lensen in front of the eyes of the user [2]. The HMD also contains sensors to track the direction and movement of the head of the user, and this tracking data is used to adjust the displayed projection of the virtual environment. This creates an illusion of full immersion for the user to place them in a virtual or environment. This environment can be fully computer generated, such as a videogame, or a scene captured from the real world using special, 360 degree cameras.

2.2 Augmented Reality

Augmented Reality (AR) takes a virtual object and places it in a real environment, by overlaying the rendered image of the object on a picture of the environment [3]. However, in AR the real and virtual parts do not interact with each other. The hardware required for AR applications is usually a smartphone, as these devices are widely available and contain the parts necessary for AR, such as a screen, camera, gyroscope etc. Thus the goal of the technology is not to be immersive, but to add a few virtual or augmented elements to the surrounding environment.

2.3 Mixed Reality

Mixed Reality (MR) is sometimes hard to differentiate from AR, as the basic concept is the same, virtual objects overlaid on the real word [4]. MR, however, uses head mounted devices similar to VR HMDs, but with the key difference of also letting the user see their surroundings. This can be achieved by either a special screen that is not enclosed and lets light through, or having cameras on the front of the device. The virtual scene is anchored to the real environment, meaning these projected objects are fixed to their designated space in the real world, or can move relative to it.

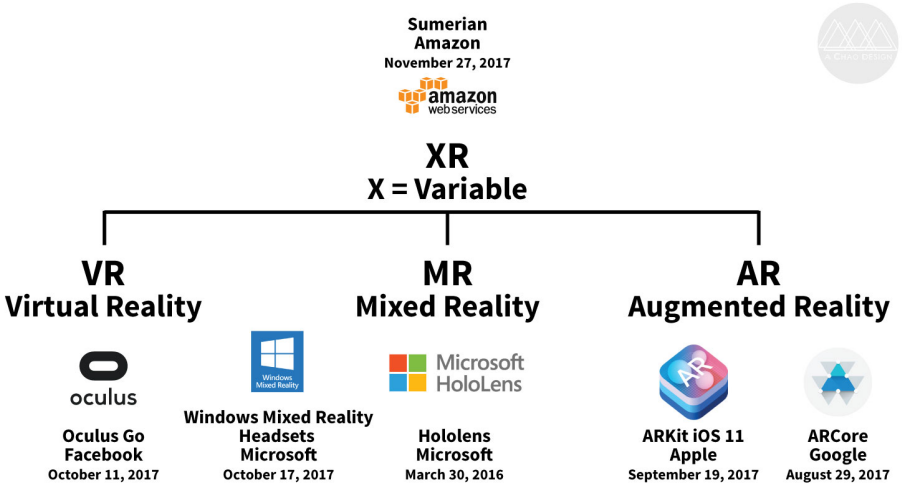


Figure 1. Categories of XR. [5]

3 Current Multi User XR applications

Having multiple users in the same virtual environment opens up new possibilities and applications for extended reality. This section explores what use-cases and implementations have been developed for virtual, augmented, and mixed reality.

3.1 Virtual Reality

The following papers and XR solutions are related to Virtual Reality.

MuVR

For the past few years, VR technology has been maturing rapidly, however the accessibility of the technology did not increase similarly, especially for multi-user applications. To address this problem, Jerald Thomas et al. have developed MuVR, a multi-user virtual reality platform, with the goal of making it as accessible as possible [6]. The platform uses reasonably cheap, off-the-shelf parts to decrease cost and increase accessibility. The hardware setup is also designed to be portable, self-contained, and quick to set up. The software solution is based on Unity [7], a 3D computer graphics engine, and any type of virtual environment or application compatible with it can be loaded and run on the system. Once a server is created, multiple users can join simultaneously and interact with the environment and each other.

In conclusion, the platform does not improve on current bleeding-edge VR technology, but proves that a similar level of performance can be achieved accessibly.

CoVR

One of the most important reasons to implement a multi-user vr system is to make interpersonal communication easier for long-distance meetings. This is especially important in the Architecture, Engineering and Construction (AEC) industry. To make remote meetings of stakeholders at AEC projects easier and more productive, in 2007, Jing Du et al. developed a multi-user VR system for this purpose [8]. This system, called CoVR, works by loading a Building Information Model (BIM) into a cloud-based game engine, creating a multi user interactive virtual environment. The stakeholders can then join the environment, where they can carry out virtual inspections and communicate with each other efficiently using their environment as visual aid. In other words, the stakeholders can dis-

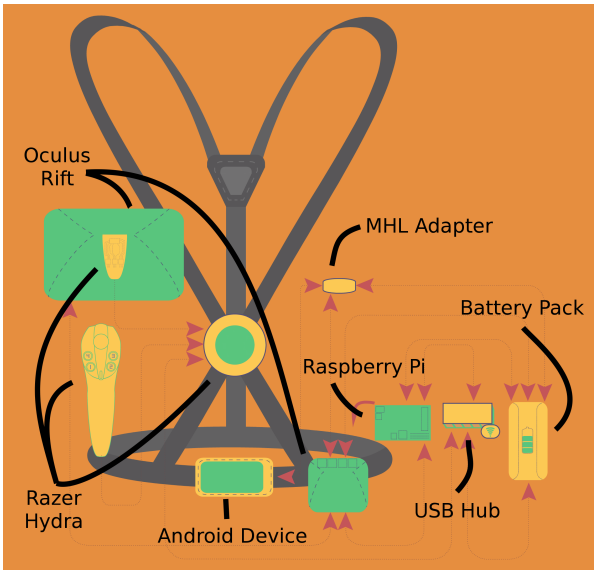


Figure 2. The components of the MuVR system. [6]

cuss details of a planned building, while standing inside the 3D plans of said building.

The system was tested with 71 participants, using a real project. The researchers have found that CoVR has enhanced communication, and users performed better at building inspection tasks compared to traditional methods. All of this illustrates that multi-user VR has a great potential in the AEC industry.

Figure 3. The CoVR system in use. [8]

EPICSAVE

Another very important use of multi-user VR is in medical training. In paramedic training it is hard to prepare for life-threatening conditions that only happen rarely, such as an anaphylactic shock. To tackle this problem, Jonas Schild et al. have developed a multi-user VR paramedic training system called EPICSAVE [9]. In this publicly funded project the goal was to develop a system where paramedic trainees can practice in a virtual environment including a patient with simulated symptoms. The system uses the HTC Vive HMD and controllers and software developed in-house. In a training situation two trainees work as a team, sharing

the same virtual and physical space to carry out their tasks. Meanwhile, a trainer controls the virtual environment by, for example, changing the symptoms of the patient.

The researchers have found that "positive learning experience depends on a high presence experience which gains from high interactivity and VR usability". Furthermore, the lack of standards in VR made development more difficult, and raised issues with user experience and usability.

Figure 4. The two trainees sharing the physical as well as the virtual space. [9]

3.2 Augmented/Mixed Reality

These papers and XR solutions are related to Augmented or Mixed Reality.

Second Surface

Second Surface, an augmented reality system, developed by Shunichi Kasahara et al. at the MIT Media lab, aims to create an environment for creative collaboration [10]. This novel multi-user AR application enables user-generated content, such as 3D drawings or text, to be displayed over real world environments and also feature real-time interactions. It uses an image-based AR recognition software called Vuforia [11] to map the surroundings, and builds a dictionary of data from the surfaces, which acts like a fingerprint, identifying surfaces as unique objects. These dictionaries are shared with other users through a cloud-based server. The hardware used for the client side are off-the -shelf tablets, using the touchscreen and camera of the device. This makes the use of the system easy and intuitive for the user. Use-cases mentioned in the paper include ex-

planation for objects, describing their role or some additional information. The researchers believe that this novel system can enable a new form of interactive communication within cities or buildings.

Figure 5. Multiple users using Second Surface. [10]

ARfract

In the paper Hybrid Augmented Reality for Participatory Learning: The Hidden Efficacy of the MultiUser Game-based Simulation, Seungjae Oh et al. discuss how augmented reality can be used to help the scientific learning process [12]. As part of their research they have developed two multi-user AR simulations that teach users about light refraction, one in a gamified way, the other not. The goal was to design an application that encourages in-depth observations, accepts natural body movement as input and naturally guides the user in understanding scientific phenomena. The system, called ARfract, uses a hybrid approach, with Meta One, a pair of see-through AR glasses, and a projector. The simulation supports two distinct types of multi-user interaction, one between the users, and one between the users and bystanders. The game based simulation has a strong cooperative element, with users trying to achieve the same goal working together, while the non-game version includes less interaction between users. The experiment was carried out with pairs of participants either starting with the game based version, then using the non-game version, or the other way around. The researchers have found that ARfract was an effective way of learning about light refraction and the group of participants trying the game based version first performed better than the group experiencing the simulations in reverse order.

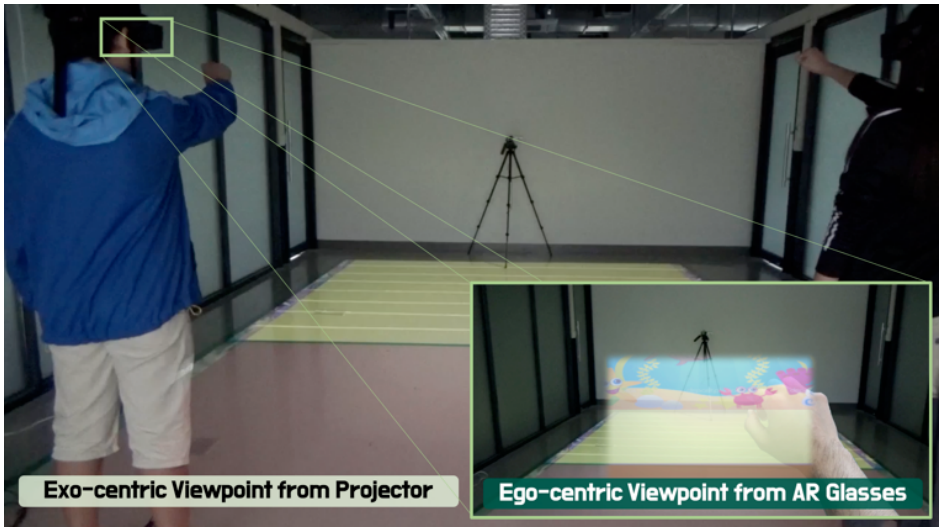


Figure 6. The ARfact system's hybrid solution. [12]

4 Discussion

Having looked at a number of a number of research papers exploring the topic, this section will discuss the benefits and problems of the technology, based on the papers referenced in the previous section.

4.1 Benefits and Potential

Communication

One of the main benefits of the technology is the potential to enhance interpersonal communication and interaction over the internet, especially in certain industrial environments. CoVR enables users to share the same virtual space from different physical locations, and allows them to freely talk using a built-in VOIP service and point at parts of the 3D building plan using a virtual laser pointer [8]. Compared to traditional 2D blueprints or 3D models, this system provides an improved way of communication, potentially reducing miscommunication, saving time and money in the process.

Immersiveness

Another benefit of the technology lies in its immersiveness, situations rarely seen can be reproduced with greater accuracy than ever before. EPICSAVE is a great example of this [9]. Virtually recreating rare medical emergencies in paramedic training provides a more effective way of preparing for these situations. Trainees participating in the study were observed to be more present and more enthusiastic, while also performed better at the training evaluation test. Furthermore, because paramedics

work as a team, having a multi-user training environment is perfect to learn cooperation in these situations.

Co-creation and the social element

Multi-user augmented reality applications can tie virtual objects to physical locations, having the potential to add a new, virtual space to the places we live our lives. This can be a collaborative creative canvas, like in the case of Second Surface [10]. Art, in a way, is meant to be shared, and applications like this bring a whole new dimension to co-created art. On a personal note, I used to play location-based multiplayer AR game, Ingress, developed by Niantic Labs [13]. In this game, the player has to physically travel to places to be able interact with virtual objects tied to that place. To work toward the goal of the game, players have to cooperate with each other, both in the real and the virtual world. During my time playing, I had met several other players, people who I never would have met otherwise. It motivated me to explore places in my surroundings and admire buildings, statues and street art. The augmented reality and multiplayer aspects of the game work together to achieve this unique gameplay, and the end result is a prime example of what can be achieved with this technology, in my opinion.

4.2 Problems

Accessibility

While augmented reality apps for smartphones are relatively accessible nowadays, virtual and mixed reality are held back by the high price of the required hardware and the limited number of applications for an average user. Solutions like MuVR try to lower this barrier to entry by taking relatively cheap, off-the-shelf parts and combining them into a more accessible multi-user VR solution [6]. The problems with MuVR is that it is only a proof of concept and that while the problem of price is addressed, the complicated installation process raises usability questions. A more viable solution could be standalone VR HMDs, like the Oculus Quest or the new Oculus Quest 2 [14]. These devices have integrated computers capable of rendering the 3D visuals necessary for VR applications, negating the need for a powerful and expensive personal computer. Multi-user software can also be run on these devices, as they also contain the wireless networking hardware required. Furthermore, the tracking implemented requires no additional external tracking sensors, making setup even easier.

Usability

Figure 7. The Oculus Quest 2 HMD and controllers. [14]

The other problem with multi-user XR is that the lack of standards creates a usability problem and makes development more difficult for these platforms. As researchers from the EPICSAVE project stated, “there are no standards, yet, so VR systems must be very careful in teaching how to use them [...] such rough factors in our VR-prototype, which lead to “breaks in presence” experience and cognitive load (e.g.,communication and navigation in VE, wired head-sets)” [9].

5 Conclusion

Multi-user XR applications have a great potential in certain industries, these use-cases are very specific and it is yet to be seen how wide the adoption of the technology will be in other industries. The problems of accessibility and usability are greatly holding back the spread of XR, but it is usually the case with such novel technologies. Interestingly, the technological advancements in this field are driven by the consumer market and with the release of increasingly more affordable, highly integrated VR and AR solutions we will possibly see it appear in more and more industries.

References

[1] Virtual Reality Market Report. https://www.marketsandmarkets.com/market-reports/reality-applications-market-458.html.

[2] G.C. Burdea and P. Coiffet. *Virtual Reality Technology*. Wiley - IEEE. Wiley, 2003.

[3] Philipp Rauschnabel, Alexander Brem, and Young Ro. Augmented reality smart glasses: Definition, conceptual insights, and managerial importance. *Working Paper, The University of Michigan-Dearborn*, 07 2015.

[4] Yuichi Ohta and Hideyuki Tamura. *Mixed Reality: Merging Real and Virtual Worlds*. Springer Publishing Company, Incorporated, 1 edition, 2014.

[5] AR What Are All These Realities? VR, MR and XR 101. https://www.achao.design/inspire/wh are-all-these-realities-vr-mr-ar-xr-101.

[6] J. Thomas, R. Bashyal, S. Goldstein, and E. Suma. Muvr: A multi-user virtual reality platform. In *2014 IEEE Virtual Reality (VR)*, pages 115–116, 2014.

[7] Unity. https://unity.com.

[8] Jing Du, Yangming Shi, Zhengbo Zou, and Dong Zhao. Covr: Cloud-based multiuser virtual reality headset system for project communication of remote users. *Journal of Construction Engineering and Management*, 144, 12 2018.

[9] J. Schild, D. Lerner, S. Misztal, and T. Luiz. Epicsave — enhancing vocational training for paramedics with multi-user virtual reality. In *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–8, 2018.

[10] Shunichi Kasahara, Valentin Heun, Austin S. Lee, and Hiroshi Ishii. Second surface: Multi-user spatial collaboration system based on augmented reality. In *SIGGRAPH Asia 2012 Emerging Technologies*, SA '12, page 1–4, New York, NY, USA, 2012. Association for Computing Machinery.

[11] Vuforia. https://www.ptc.com/en/products/vuforia.

[12] S. Oh, H. So, and M. Gaydos. Hybrid augmented reality for participatory learning: The hidden efficacy of multi-user game-based simulation. *IEEE Transactions on Learning Technologies*, 11(1):115–127, 2018.

[13] Ingress by Niantic Labs. https://www.ingress.com/.

[14] The Oculus Quest 2. https://www.oculus.com/quest-2/.

Methods for coding e-cigarette use and vaping on Instagram: a systematic review

Walter Berggren
walter.berggren@aalto.fi

Tutor: Aqdas Malik

Abstract

With increasing e-cigarette use and vaping among the youth, there is research interest in the role of social media on nicotine use. This report reviews methods for coding e-cigarette use and vaping on Instagram. First, a literature search was conducted and 14 studies on the topic were chosen for further analysis. Secondly, data extraction was performed on the studies to summarize methodological information, coding methods, and the coded categories. The coding methods were found to be dividable in two categories: human-coding and machine-learning methods. Machine-learning methods use convolutional neural networks for feature extraction from images. Feature classification was done using both supervised and unsupervised machine learning methods. For captions, latent Dirichlet allocation, linear support vector machines, and logistic regression are used. Finally, the hand-coded and machine-learning methods are compared in terms of accuracy and performance. Hand-coding methods are found to code a broader range of categories than machine-learning methods with more nuanced distinctions, but are limited in the number of codable posts. Machine-learning approaches are found to be able to code a sufficiently broad set of categories by combining both image- and caption-based methods with satisfactory accuracy.

KEYWORDS: Instagram, e-cigarettes, vaping, review

1 Introduction

Instagram is one of the most popular social media platforms among the youth [3]. Due its popularity, there has been a growing interest in using Instagram posts for research purposes. Simultaneously, there is a growing trend of nicotine use among the youth in the form of novel tobacco products such as e-cigarettes [10]. One approach to study the growth is researching how the novel products are exposed to the youth on social media. This requires methods for systematically analyzing the contents of user posts. In data analysis, coding is used to assign labels to data for consolidating meaning. In the context of social media, coding can be used to consolidate meaning from user posts. This can be used to better understand e.g. sentiment towards novel tobacco products and how they are being exposed to users. Previous studies [16] have compared different methods for coding tobacco-related data on Twitter. The purpose of this literature review is to apply a similar approach and systematically review methods for coding tobacco-related data, but on Instagram instead of Twitter and limited to e-cigarette use and vaping.

2 Background

According to *The coding manual for qualitative researchers* ([19], p. 4), a code is "a word or short phrase that symbolically assigns a summative, salient, essence-capturing, and/or evocative attribute for a portion of language- based or visual data". For example, the transcript of an interview can be coded by assigning a topic to each sentence. These topics could then be used to identify patterns in the data based on characteristics such as similarity and frequency. Hence, coding is analysis and can be seen as a heuristic to discover new information. This is achieved by not only labeling existing data, but also enabling linking of the original data. Furthermore, the codes themselves can be synthesized to form broader categories that can be used not only to reduce an answer, but also to consolidate meaning. In other words, coding is a technique that can be used to help analyze and consolidate meaning from sets of data [19].

3 Methods

For reviewing and comparing different methods for coding Instagram posts related to novel tobacco products a systematic approach to literature search and data extraction is needed. The approach chosen is based on a similar article by Lienemann et al. [16] that compared methods for coding tobacco-related Twitter data. While focused on Twitter data, the proposed method is applicable to systematically reviewing coding methods for Instagram posts as well. The following sections describe how the literature search was conducted, how the study selection was done from the literature search, and how data was extracted from the selected studies.

3.1 Literature search

A literature search was conducted in two stages with the first one being in September 2020 and the second one in November 2020.

In September 2020, the e-article search functionality in Finna was used (a search tool provided by the National Library of Finland). The search terms were restricted to include only Instagram-related articles concerning coding of e-cigarette use and vaping. This was done by formulating the search query to include any article mentioning Instagram and a smoking-related term in the title. For selected smoking-related terms, the search terms used in the Twitter article by Lienemann et al. [16] were used as base because they were considered to be comprehensive. However, only a subset of the search terms were finally used to limit the search in this article to e-cigarette use and vaping. The search term *e-liquid** was also added to include the liquid used in e-cigarette products. The final search query was formulated as:

```
"Instagram" AND
  (tobacco OR nicotine OR cig* OR smok* OR vap* OR e-liquid*)
```

This search query returned 24 results.

In November 2020, a second search was conducted to verify the comprehensiveness of the first search. The verification search was performed on Web of Knowledge using the search query:

```
(TS=(Instagram AN
  (tobacco OR nicotine OR cig* OR e-cig* OR electronic*
    OR smok* OR vap* OR e-liquid*
)) AND LANGUAGE: (English) AND DOCUMENT TYPES: (Article)}
```

This search query returned 85 results.

3.2 Study selection

The results of the two literature searches were reviewed to verify that they concerned coding Instagram posts related to e-cigarette use and vaping.

Out of the 24 articles found in the first literature search, three were removed for concerning cigars, one for concerning water pipes, and three for being duplicates. Furthermore, five articles were removed for not coding posts (instead, they investigated perception by showing posts to volunteers).

Out of the 85 articles found in the second literature search, 18 results were removed for being duplicates from the first search, 57 for concerning unrelated topics, five for being out of scope (concerning water pipes, betel nets or cigarillos), and three for not coding posts. After removal, two additional articles were added to the set of studies to review.

3.3 Data extraction

The articles were read to extract methodological information. Methodological information was extracted on data collection (date collected, data collection rate, keyword/hashtag selection, data source), coding methods (coding method, number of posts coded, number of Instagram users, retrieval precision, retrieval recall, and coding agreement), and the coded categories (categories, based on image and/or caption). The extracted information is presented in tables 1 on page 5, 2 on page 6, and 3 on page 7. The tables contents are summarized in the following results section.

4 Results

Coding Instagram posts involves both fetching Instagram posts and the actual coding. First, the fetching methods used in the reviewed articles are presented. Secondly, the coding methods are described and their performance in terms of accuracy is summarized. Finally, the number of posts coded is compared across the different methods.

Article	Date collected	Data collection rate	Keyword selection	Data source
[14]	May 2017 and October 2017	100 posts per hour	#eliquid, #ejuice	Netlytic
[13]	May 2017, October 2017, March 2018, August 2018, and September 2018	100 posts per hour	#eliquid, #ejuice	Netlytic
[12]	October 2014	Sample from the period	#ecig, #evape	Iconosquare
[4]	2015-10-01–2015-12-31	All posts from the period	#ecig, #ejuice, #eqliquid, #vape, #vaping, #vapelife	Instagram API
[5]	2014-08-01–2015-07-31	Sample from the period	#cig, #cigarette, #cigarettes, #cigarillo, #cigarillos, #cigs, #ecig, #ecigs, #hookah, #nicotine, #rellos, #rillo, #rillos, #smokeclouds, #smokeselfie, #smokingselfie, #smokingtricks, #vape	Instagram API
[15]	"Span of five weeks"	10 posts twice a week during random times of the week	#vape, #vapelife, #vapor, #vapelyfe, #vapeporn, #ecig, #vaping, #cigs, #electroniccigarette	Statigram/Iconosquare
[7]	2018-03-01–2018-05-15	NS	#juul, #juuling, #juulvapor, #juulpod, #switchtojuul, #juulgang	NUVI
[17]	2018-06-08–2018-08-08	First 1000 posts per hour	#kandypens	Netlytic
[21]	2017, 2018, and 2019	100 posts per hour during 2 week sampling periods	2017 and 2018: #vape, #vapenation 2019: #vape, #ejuice, #eliquid, #vapecommunity, #vapefam, #vapelife, #vapelyfe, #vapenation, #vapeporn #TheRealCost	Instagram API
[11]	2019-06-01–2019-10-31	NS	#vaping	Instagram, scraped
[9]	2019-08-22–2019-09-12	All posts	#ejuice	Instagram, scraped
[6]	2018-03-01–2018-05-15, 2018-05-16–2018-11-16	Unspecified	#doit4juul, #girlswhojuul, #justjuul, #juuhit, #juul, #juul_university, #juulbabes, #juulbabies, #juulbong, #juulboys, #juulbreak, #juulcentral, #juulchin, #juulcompatible, #juulers, #juulfanatics, #juulfavorite, #juulforsale, #juulforyou, #juulgang, #juulgirls, #juulgirlz, #juulhigh, #juulhighschool, #juulhumor, #juuling, #juulinstock, #juuljob, #juulkit, #juullife, #juullovers, #juulluge, #juullyfe, #juulmangopods, #juulmoment, #juulnation, #juulpen, #juulpod, #juulpods, #juulrealm, #juuls, #juulsjuuls, #juulskins, #juulstarterkit, #juulstick, #juulvapar, #juulvape, #juulvapes, #juulvapor, #switchtojuul	NUVI
[1]	2017-11-03–2017-11-17	100 posts per hour	#ejuice, #eliquid	Netlytic
[20]	2012-02-03–2015-04-16	All posts	Four e-cigarette brands' (Blu, NJOY, Logic, and Metro) Instagram pages	Curalate

Figure 1. An overview of how Instagram data was collected in the reviewed articles. *NS: Not specified*.

4.1 Fetching Instagram posts

The reviewed articles fetch Instagram posts using Netlytic, Iconosquare/ Statigram, NUVI, Curalate, the Instagram application programming interface (API), and by scraping the web version of Instagram.

Netlytic is a tool for analyzing social network posts without programming knowledge that uses public APIs to retrieve data [18]. In the five reviewed articles that used Netlytic [14, 13, 17, 1], the post retrieval rate was limited to 100, 1000 or 10 000 posts per hour. However, support for Instagram was dropped by Netlytic in 2018 [18].

Iconosquare (previously Statigram) is an authorized third-party website that allows access to Instagram data [12]. This service was used by two

Article	Coding method	No. of posts coded	No. of Instagram users	Retrieval precision	Retrieval recall	Coding agreement
[14]	Hand-coded by researchers	May: 500 October: 500	NS	NR	NR	$\kappa = 0.81$
[13]	Hand-coded by researchers	Prewarning: 1,500 Postwarning: 1,000	NS	NR	NR	$\kappa = 0.85$
[12]	Hand-coded by researchers	85	NS	NR	NR	$\kappa = 0.77$
[4]	Hand-coded by researchers	2,208	NS	NR	NR	$\kappa = 0.91$
[5]	Hand-coded by researchers	5,721	NS	NR	NR	Krippendorff's $\alpha = 0.91$ within the team
[15]	Hand-coded by researchers	900	NS	NR	NR	NS
[7]	Sample hand-coded by researchers Depending on coding category: Logistic regression with L1 regularization; Linear support vector machine (SVM) with radial kernel, optimized with grid search; language-based filter	40,071	5,945	92%	92%	
[17]	Hand-coded by researchers	1,775	546	NR	NR	$\kappa = 0.85 - -0.96$
[21]	Image: Deep learning; convolutional neural networks Text: Tidytext	2017: 22,293 2018: 21,906 2019: 201,703 FDA campaign #TheRealCost: 46	NS	NR	NR	
[11]	Convolutional neural networks; K-means clustering; VGG-19	560,414	45,232	"Validation loss of 0.2575 and validation accuracy of 90.76%		
[9]	Hand-coded by researchers	1,936	NS	NR	NR	"70% to 90%"
[6]	Sample: Label if content was JUUL-related Classification: Linear support vector machine (SVM) Analysis: Semantic network analysis (SNA)	50,817	16,323	> 90%	> 90%	
[1]	Hand-coded by researchers	3481	NS	NR	NR	"70.6% to 89.9%"
[20]	Classification: Multinomial logistic regression Analysis: Discriminant function analysis (DFA)	5022	4	Blu: 91.40% NJOY: 61.63% Logic: 50.87% Metro: 45.55%		NR

Figure 2. An overview of the coding methods used in the reviewed articles. *NS: Not specified*. *NR: Not relevant*.

articles [15, 12] and data retrieval was limited to unspecified samples from a particular month and 10 posts twice a week during random times of the week.

NUVI is “a licensed syndicator of Instagram data” [6]. The data collection rate was unspecified in both two articles that used the service [6, 7].

Curalate is “a visual commerce platform that collects and compiles posts from brand-owned social media accounts” [20]. In the study that used the service [20] all posts from the time period were provided. However, the

Article	Coding method	Categories	Image and/or caption
[14]	Hand-coded by researchers	54 non-mutually exclusive coding categories. Ten topics: Descriptive metadata; user type; user location and language; image content; promotional practices and strategies; product information and flavors; marketing themes; health and product claims; politics; community and addiction hashtags	Image and caption
[13]	Hand-coded by researchers	Instagram user type; post language and user location; descriptive metadata; mentions of age restrictions for purchasing content; political statements and calls for advocacy about vaping; post visuals, promotional themes, and health claims ; the nicotine level of promoted e-liquids; the flavors; precense of FDA-mandated nicotine warning statement language on post images	Image and caption
[12]	Hand-coded by researchers	Post metadata/demographics; Healthpromotion and smoking cessation; Use and depiction of e-cigs; Shared identity/community; Critical of e-cigs	Image and caption
[4]	Hand-coded by researchers	Activity; Product; Advertisement; Text; Other	Image and caption
[5]	Hand-coded by researchers	Artificat visible; person number; brand name or logo; smoking normally; smoke play; presence of marijuana; anti-smoking; gender; age; ethnicity	Image and caption
[15]	Hand-coded by researchers	Customization; juice/flavors/ memes; marketing; celebrities; health benefits; anti-smoking; models; tricks; marijuana; social acceptance	Image and caption
[7]	Sample hand-coded by researchers Depending on coding category: Logistic regression with L1 regularization; Linear support vector machine (SVM) with radial kernel, optimized with grid search; language-based filter	Promotion; nicotine and addiction; lifestyle and youth-related appeals	Caption only
[17]	Hand-coded by researchers	Theme: User experience; product appearance; promotions; flavours; other Type of e-liquid solutions: Cannabis-related solutions; nicotine-related solutions; aromatherapy-related solutions; no solutions User profile: Average Instagram user; Vape vendor; KandyPens official account; Vaping enthusiast/advocate; Influencer; Other	Image and caption
[21]	Image: Deep learning; convolutional neural networks Text: Tidytext	Image: Man; Woman; Mod; Pod; E-juice; Other Caption: Promotions; Flavors; Devices; User experience	Image and caption
[11]	Convolutional neural networks; K-means clustering; VGG-19	E-liquid; e-liquids; e-cigarette; product package; persons; statement; miscellaneous	Image
[9]	Hand-coded by researchers	Cartoon; promotional content; cartoon as the company's logo; name of the company that used cartoons; Instagram user engagement data	Image and caption
[6]	Sample: Label if content was JUUL-related Classification: Linear support vector machine (SVM) Analysis: Semantic network analysis (SNA)	Major clusters: Youth and other tobacco and cannabis-related content; JUUL and other brand product characteristics; general vape community; promition of over-consumption	Caption
[1]	Hand-coded by researchers	Cartoon; logo; promo	Image and caption
[20]	Classification: Multinomial logistic regression Analysis: Discriminant function analysis (DFA)	Blu; NJOY; Logic; Metro	Caption

Figure 3. An overview of the coded categories in the reivewed articles.

results were limited to posts published by a brand's official account.

The Instagram API is the official programmatic way to access Instagram posts and is composed of three different APIs: a legacy API, the Instagram Basic Display API, and the Instagram Graph API. Which of these three APIs that is used in the three articles that use "the Instagram API" is

unspecified, but it is assumed to be the legacy API as the graph API was not released until January 2018 [8]. In the three articles that used the Instagram API [4, 5, 21], the data collection rate was either specified as all posts from a period, a sample from the period or a throttled rate of 100 posts per hour during two week sampling periods.

The final approach is scraping data from the web version of Instagram and this approach was used by two articles [11, 9]. In one of the articles the data collection rate was unspecified, but in the other all posts from the specified period were collected.

4.2 Hand-coding by researchers

Hand-coding involves researchers visually inspecting Instagrams post content and determining what categories posts belongs to based on the image and caption. In the reviewed articles, coding was usually done by either one or two researchers on different sets of posts. To validate consistency, a subset of the posts were typically double coded independently by two different researchers. By doing such double coding, the consistency can be quantified by calculating Cohen's kappa (κ). Cohen's kappa is a measurement that varies between 0.0 and 1.0, where a score of 1.0 indicates perfect agreement. The agreement in the reviewed articles varied between $\kappa = 0.77$ and $\kappa = 0.96$. If interpreting κ values of 0.61–0.80 as good and 0.81–1.0 as very good [2, 16], one of the articles specifying Cohen's kappa are considered good and four as very good.

In one article [5], hand-coding was split into multiple teams that coded different subsets of the posts. For measuring agreement within teams Krippendorff's alpha (α) was used and measured to be $\alpha = 0.91$, which was considered by the authors to be a high level of agreement.

4.3 Machine learning methods

Machine learning methods for coding Instagram posts can be applied on both the images and on their captions (text). The following sections compare the machine learning methods that have been used on images as well as on text.

Image: Convolutional neural networks

Convolutional neural networks (CNNs) can be used to extract features from images. The features can then be used to determine class of the image, i.e. the image is coded. Coding images using CNNs has been done

by Ketonen and Malik [11] and Vassey et al. [21].

Ketonen and Malik [11] used VGG-19, a particular CNN architecture, to predict a class label (a category) for each input image. The class labels were determined by extracting large set of features using VGG-19 pre-trained on ImageNet data, reducing the feature set using principle component analysis (PCA), and clustered using k-means clustering with $k = 7$. The clustering yielded seven distinct classes (e-liquid, e-liquids, e-cigarette, product package, persons, statement, miscellaneous) that were then predicted using the final CNN. The authors achieved validation loss of 0.2575 and validation accuracy of 90.76 %.

Vassey et al. [21] used Inception v3, a CNN trained by Google on the ImageNet dataset. The CNN was fine-tuned using the image results from querying Google Images and Instagram for the six identified classes (man, woman, mod, pod, e-juice, other). The fine-tuned model achieved validation loss of 0.32 and validation accuracy of 90 %.

Text: Latent Dirichlet allocation

Latent Dirichlet Allocation (LDA) is a method for finding patterns in collections of text and attempt to group the texts by topic. In the context of Instagram posts, the texts are the captions associated with a post. Vassey et al. [21] used captions by extracting the most frequently used keywords (ignoring stop words such as *the*) using R's Tidytext package and grouped the extracted keywords using LDA. The grouping resulted in 50 topics that were by hand identified as four different themes: promotions, flavors, devices, and user experience.

Text: Linear support vector machine

Linear support vector machines (SVMs) are supervised machine learning algorithms used for classification. Czaplicki et al. have used SVMs for classifying posts based on the captions in two different articles [7, 6]. In both articles the authors used SVMs for data cleaning by determining whether posts were related to the JUUL brand or not (a particular brand of e-cigarettes). The performance of the classification had a precision and recall of 0.92 and 0.92 in the first article [7] and "> 0.90" in the second article [6]. Furthermore, the first article [7] also used SVMs to determine whether posts contained youth content or not. For this, the classifier precision was 0.83 and the recall 0.70.

Text: Logistic regression

Logistic regression can be used for modeling the probability of a binary outcome given an input. Czaplicki et al. [7] modeled the two binary outcomes of whether a post contains promotional content or not and whether a post contains nicotine and addiction-related content or not. This was done by constructing two logistic regression models based on human-coded training data with L1 regularization to reduce overfitting. For promotional content, the model achieved a precision of 0.84 and a recall of 0.84. For nicotine and addiction-related content, the precision was 0.81 and recall 0.70.

Multinomial logistic regression extends the use case to more than two outcomes. Vandewater et al. [20] used multinomial logistic regression to predict by whom a post had been published from four different e-cigarette brands (Blu, NJOY, Logic, and Metro). The input to the model was based on a document-term matrix with the captions and the frequency of words within the captions. The model was found to correctly classify 91.40 % of Blu posts, 61.63 % of NJOY posts, 50.87 % of Logic posts, and 45.55 % of Metro posts.

5 Discussion

The results show differences in how Instagram posts are fetched and how posts are coded. The following sections discuss the performance of Instagram post fetching, how the resulting categories differ between the different coding methods, and their accuracy.

5.1 Instagram post fetching performance

Instagram post fetching performance varies greatly between the reviewed methods. The worst performing method is limited to 10 posts twice a week while the highest performing methods are able to fetch all posts from a given period. In the reviewed methods, fetching all posts was achieved both by using the Instagram API and by scraping the web version of Instagram. However, none of the methods utilizing a third-party service (Netlytic, Iconosquare/Statigram or NUVI) achieved this (Curalate did, but was limited to posts by a particular brand's official account). Hence, the results suggest that the desired approaches for fetching the highest number of user posts is by using the Instagram API or scraping the web version of

Instagram.

5.2 Coded categories

The types of coded categories differ depending on whether human-coding or machine learning is used. Within machine learning methods, the coded categories also differ depending on whether images or captions are used.

In general, the reviewed human-coding methods achieve more categories with a median of nine categories compared to four categories for machine learning. Human-coding is also able to recognize nuanced distinctions such as “smoking normally” vs “smoke play” and the presence of models or celebrities, nuances that the reviewed machine-learning methods do not demonstrate. On the other hand, human-coding is limited in the number of posts coded with 2500 posts being the median, while the reviewed machine-learning methods code a median of 50817 posts. The highest number of posts coded with machine-learning methods is 560414 compared to 5721 with hand-coding.

Within machine learning methods, image-based coding is able to distinguish between the presence of e-liquid(s), e-cigarettes, packaging, persons, statements, men, and women. Concerning these categories, the reviewed caption-based coding methods capture “statements” (similar to “promotion”), but are limited to “devices” more broadly and do not code gender at all. However, coding the categories “flavors”, “user experience”, “nicotine and addiction”, and age (“youth lifestyle”) is unique for caption-based methods.

Coding accuracy is high for both hand-coding and machine-learning methods. All hand-coding methods disclosing agreement in terms of Kohen’s kappa achieve scores that are generally considered good or very good [2]. The machine-learning methods disclose precision accuracy ranging between 83 and 91 %, the only study with significantly worse results is the method using multinomial logistic regression to estimate e-cigarette brands based on captions with as low as 45.55 % precision for the worst performing brand. This suggests that the reviewed method is not sufficient for coding for brands, but that the other reviewed methods for coding posts based on either images or captions are sufficiently accurate.

6 Usage, review limitations, and directions for further research

Researchers interested in studying e-cigarette use and vaping on Instagram can use these findings either to find previous studies on user behavior or as an aid to choosing an appropriate coding method for new studies. For the latter, the final table 3 listing the coded categories is the entry point for finding methods that cover the desired categories. Then, the second table 2 can be used to compare the methods used and their performance (both number of posts coded and precision). Finally, the first table 1 can be used to identify an approach to fetching Instagram posts and relevant key words to use in the search.

This study is limited to the search results found from the used search queries, which means that there may be additional studies on the topic that were not found. Furthermore, studies on other tobacco products were excluded from the review and further research could expand the scope to cover a broader range of tobacco products such as hookahs and cigars.

Performance comparison between different studies is also limited. This is due to the methods using varying categories as well as different data sets of Instagram posts to code posts. Further research could standardize a set of categories and Instagram posts and use the coding methods from the selected studies to better evaluate how their performance compares to each other.

7 Conclusion

Previous research on coding e-cigarette use and vaping on Instagram can be divided into hand-coded methods and machine-learning methods. Hand-coded methods rely on researchers assigning user posts to categories manually. Machine-learning methods use either images or captions. For images, convolutional neural networks are used to extract features from images and for feature classification both supervised and unsupervised machine learning methods are used. For captions, latent Dirichlet allocation, linear support vector machines, and logistic regression are used.

Hand-coded methods are able to code a slightly larger number of categories than machine-learning methods with more nuanced distinctions, but are limited in the number of posts they are able to code. Machine-learning approaches are able to code a broad set of categories by combining both image- and caption-based methods with satisfactory accuracy.

References

[1] Jon-Patrick Allem et al. “Return of cartoon to market e-cigarette-related products”. In: *Tobacco control* 28.5 (2019), pp. 555–557. DOI: 10.1136/tobaccocontrol-2018-054437.

[2] Douglas G Altman. *Practical statistics for medical research*. CRC press, 1990. DOI: 10.1002/sim.4780101015.

[3] Pew Research Center. *Social Media Fact Sheet*. Accessed Nov. 6, 2020. 2019. URL: https://www.pewresearch.org/internet/fact-sheet/social-media/.

[4] Kar-Hai Chu et al. “Vaping on Instagram: cloud chasing, hand checks and product placement”. In: *Tobacco control* 26.5 (2017), pp. 575–578. DOI: 10.1136/tobaccocontrol-2016-053052.

[5] Daniel K Cortese et al. “Smoking selfies: using instagram to explore young women’s smoking behaviors”. In: *Social Media+ Society* 4.3 (2018), p. 2056305118790762. DOI: 10.1177/2056305118790762.

[6] Lauren Czaplicki et al. “# toolittletoolate: JUUL-related content on Instagram before and after self-regulatory action”. In: *Plos one* 15.5 (2020), e0233419. DOI: 10.1371/journal.pone.0233419.

[7] Lauren Czaplicki et al. “Characterising JUUL-related posts on Instagram”. In: *Tobacco Control* 29.6 (2020), pp. 612–617. DOI: 10.1136/tobaccocontrol-2018-054824.

[8] Facebook for Developers. *Instagram Graph API Launches and Instagram API Platform Deprecation*. Accessed Nov. 6, 2020. Jan. 2018. URL: https://developers.facebook.com/blog/post/2018/01/30/instagram-graph-api-updates/.

[9] Allison Dormanesh, Matthew G Kirkpatrick, and Jon-Patrick Allem. “Content Analysis of Instagram Posts From 2019 With Cartoon-Based Marketing of e-Cigarette–Associated Products”. In: *JAMA pediatrics* (2020). DOI: 10.1001/jamapediatrics.2020.1987.

[10] Andrea S Gentzke et al. “Vital signs: tobacco product use among middle and high school students——United States, 2011–2018”. In: *Morbidity and Mortality Weekly Report* 68.6 (2019), p. 157. DOI: 10.15585/mmwr.mm6806e1.

[11] Vili Ketonen and Aqdas Malik. “Characterizing vaping posts on Instagram by using unsupervised machine learning”. In: *International Journal of Medical Informatics* 141 (2020), p. 104223. DOI: 10.1016/j.ijmedinf.2020.104223.

[12] Linnea I Laestadius, Megan M Wahl, and Young I Cho. “# Vapelife: An exploratory study of electronic cigarette use and promotion on Instagram”. In: *Substance Use & Misuse* 51.12 (2016), pp. 1669–1673. DOI: 10.1080/10826084.2016.1188958.

[13] Linnea I Laestadius et al. “Compliance with FDA nicotine warning statement provisions in e-liquid promotion posts on Instagram”. In: *Nicotine & Tobacco Research* (2020). DOI: 10.1093/ntr/ntaa092.

[14] Linnea I Laestadius et al. “From Apple to Werewolf: A content analysis of marketing for e-liquids on Instagram”. In: *Addictive behaviors* 91 (2019), pp. 119–127. DOI: 10.1016/j.addbeh.2018.09.008.

[15] Alexander S Lee et al. “A picture is worth a thousand words: electronic cigarette content on Instagram and Pinterest”. In: *Tobacco prevention & cessation* 3 (2017). DOI: 10.18332%2Ftpc%2F74709.

[16] Brianna A Lienemann et al. “Methods for Coding Tobacco-Related Twitter Data: A Systematic Review”. In: *Journal of Medical Internet Research* 19.3 (Mar. 2017), e91. DOI: 10.2196/jmir.7022.

[17] Anuja Majmundar et al. “Characterising KandyPens-related posts to Instagram: implications for nicotine and cannabis use”. In: *Tobacco control* 29.4 (2020), pp. 472–474. DOI: 10.1136/tobaccocontrol-2019-055006.

[18] Netlytic. *Data Source: Instagram*. Accessed Nov. 6, 2020. 2018. URL: https://netlytic.org/home/?page_id=254.

[19] Johnny Saldaña. *The coding manual for qualitative researchers*. SAGE, 2015. DOI: 10.1177/0092055X18760362.

[20] Elizabeth A Vandewater et al. “Whose post is it? Predicting E-cigarette brand from social media posts”. In: *Tobacco regulatory science* 4.2 (2018), pp. 30–43. DOI: 10.18001/TRS.4.2.3.

[21] Julia Vassey et al. “# Vape: measuring e-cigarette influence on Instagram with deep learning and text analysis”. In: *Frontiers in Communication* 4 (2020), p. 75. DOI: 10.3389/fcomm.2019.00075.

Capacity planning for vehicular fog computing

Joonas Rissanen
joonas.rissanen@aalto.fi

Tutor: Wencan Mao

Abstract

KEYWORDS: vehicular fog computing, fog computing, capacity planning

This review paper researches capacity planning methods used in the cloud, fog and edge computing and reviews their feasibility and limitations in vehicular fog computing.

The increased number of IoT applications has created a demand for more communication and computation capacity at the edge of the network. The IoT applications, such as vehicular applications, require low latency and high bandwidth in order to succeed in their latency-sensitive tasks. Vehicular fog computing is a fog computing architecture that uses mobile vehicles as carriers for fog nodes which has been enabled by the vehicle-to-vehicle and device-to-device communication technologies. Vehicular fog computing enables on-demand resource distribution cost-efficiently at the edge of the network.

The researched capacity planning methods have shown to be promising but with some limitations. The capacity planning method should consider changes in traffic between various areas at different times so resources could be distributed to areas according to their needs. Profiling requirements of different tasks were shown to be promising for vehicular fog computing because it had the potential to also bring improvements to the networking side while using 5G.

1 Introduction

The usage of Internet of things (IoT) has increased quickly in the last decade. IoT refers to a concept where various devices are connected to the Internet interacting with each other [3]. Along with the IoT, cloud computing services have become more and more popular. Cloud computing has provided on-demand services that IoT devices require. These services can be for example computation, storage and communication services. However more and more IoT applications require low latency but cloud computing cannot guarantee it since the distance between IoT devices and centralized data centers grows at the edge of the network which increases the latency and leads to worse quality of services (QoS). For example latency-sensitive tasks of self-driving cars require a lot of computing power and communication capacity in the interest of making safe decisions [14].

Fog computing can help to solve the above problem. Fog computing refers to an architecture where application services are hosted at the edge of the network as close as possible to the end-users reducing latency and optimizing bandwidth usage [12]. Compared to cloud computing's centralized architecture, fog computing uses a decentralized architecture that brings services to the edge where the data is generated and used [4].

Recently fog computing has been expanded to vehicular networks which is called vehicular fog computing. Vehicular fog computing refers to a fog computing architecture where fog nodes are installed on mobile connected vehicles [14]. The main strength of this architecture is the mobility of fog nodes. It enables on-demand resource delivery cost-efficiently at the edge of the network.

There are three decisions that must be made when designing fog computing architecture [13]. Where should the fog nodes be installed? How requests are routed from end-user to fog node? How many fog nodes should each data center hold? Vehicle-to-Vehicle (V2V) and Device-to-Device (D2D) communication technologies enable vehicles to serve as fog nodes which provides more possibilities [14]. But also the possibility of deploying fog nodes as static roadside units should not be neglected. Communication between end-users and vehicles is handled using V2V or D2D communication technologies as stated previously. The third question is

more complicated. Should the fog nodes be deployed on buses and taxis as proposed in work [14], to all vehicles or what kind of vehicles? How much capacity should each vehicle carry, and do they have homogeneous fog nodes or do some vehicles carry more powerful fog nodes.

Since the research in fog computing and vehicular fog computing is still at its infant age, this paper researches capacity planning methods in the cloud, fog and edge computing. Then it reviews their feasibility and limitations in vehicular fog computing.

This paper is organized as follows. Section 2 presents the architecture and the challenges of vehicular fog computing. Section 3 presents basic concepts used in capacity planning, related work for capacity planning methods used in cloud, edge and fog computing, and finally feasibility and limitations of presented methods in vehicular fog computing. Section 4 discusses observations of section 3 and finally, section 5 concludes the paper.

2 Vehicular fog computing

2.1 Architecture

Architecture for vehicular fog computing is composed from different layers; data generation layer, fog layer and cloud layer [5] [8]. Figure 1 shows a basic architecture for vehicular fog computing.

End users

End users can be anyone from vehicles to pedestrians to other road users. Generally, for vehicular fog computing, end users are smart vehicles. Smart vehicles generate data with their sensors (e.g. cameras, radars and GPS). The estimated amount of data collected by a smart vehicle is 25 GB/h in a single day [5]. Smart vehicles can process some of the collected data but their computing capacity is limited. In this case, they can offload their tasks to the fog layer for processing or other purposes which is enabled by the V2V and D2D communication technologies such as 802.11p [1].

The fog layer contains fog nodes that act as a middleware that processes the collected data and reports the processed data back to end-users and to the cloud server. Fog nodes can be deployed into static roadside units or mobile vehicles. Currently, much of the research has focused on how vehicles could be utilized as a carrier to deploy fog nodes. Y. Xiao et al.

[14] proposed a vehicular fog computing model where fog nodes are deployed on buses and taxis which allows the fog nodes to move along with the changing traffic. M. Sookhak et al. [11] present the vehicular fog computing architecture where computing power of parked cars could be utilized by surrounding end-users.

Cloud servers

The Cloud layer consists of centralized cloud servers. In some cases, fog nodes upload the data to cloud servers for additional processing or storing purposes. The Cloud layer also enables backing up important data uploaded by the fog layer.

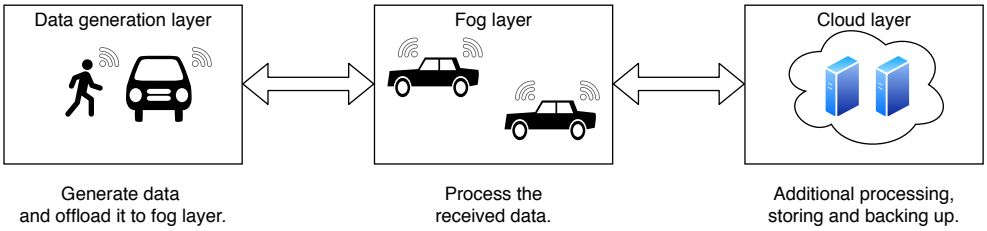


Figure 1. Architecture for vehicular fog computing [5].

2.2 Challenges

The traffic keeps changing over time and between different areas. It can be unpredictable at times due to car accidents or some other unpredictable events. This is a major challenge for capacity planning in vehicular fog computing because QoS drops instantly when there is not enough capacity at the edge [14]. But then also there should never be too much capacity since it increases the cost.

One challenge is managing computation capacity. The cost for computation should be minimized, the tasks should be managed and distributed among the nearby vehicular fog nodes [4]. The non-latency-sensitive applications can be deployed on the cloud but the latency-sensitive applications should be deployed on the vehicular fog nodes.

There are also challenges in managing communication capacity. The reliability of wireless and cellular networks is challenging especially when fog nodes and end-users are moving simultaneously. This causes changes in network topology which leads occasionally to uncertain connections between vehicles. [14] Also, wireless and cellular networks might not cover all the areas and their bandwidth has limited capacity which should al-

ways be taken into consideration.

3 Capacity planning

The goal of capacity planning is to optimize the number of vehicular fog nodes while maximizing the QoS and minimizing the cost of infrastructure. In addition, capacity planning is also used to find optimal specifications for nodes e.g. whether to use nodes with more computing power or storing capacity.

There are two types of latencies that must be taken into account when doing capacity planning for vehicular fog computing; the processing time of the fog node and the networking time from end-user to fog node. Processing time depends on factors such as the configuration of hardware, the queuing of the end-users, the requirements of vehicular application etc. The networking time depends on direct and indirect factors. Directly it depends on factors such as the offloading time, bandwidth strength etc. Indirect factors are for example velocity, the distance between the end-user and the fog node, density of users connected to the same fog node and data traffics [13].

3.1 Self-Adaptive Cloud Capacity Planning

Y. Jiang et al. [7] propose a self adaptive capacity planning method for cloud computing. They use historical data of requests to create provisioning and de-provisioning estimations. In provisioning estimation, an over-estimation led to increased cost due to idled resources, and an underestimation led to worse QoS. In de-provisioning estimation, an over-estimation led to worse QoS and an underestimation led to an increased amount of idled resources. For both estimations, they calculated the cost of idled resources and the cost of service-level agreement (SLA) penalty which happens when the QoS violates the predefined agreement. Through experiments, they demonstrated that their method can significantly reduce the maintenance cost of the cloud environment with their capacity planning method.

The cost of the SLA penalty might be too hard to estimate since there is no practical deployment. On the other, the cost of idled resources can be estimated as long as the historical data can be provided. However, this might also be challenging since vehicular applications are not yet in

mainstream use. The method does not either consider the networking side in capacity planning which could be a limiting factor. Still, this method could be quite feasible in the future when historical data can be provided and fog computing services become more popular.

3.2 Capacity Planning of Fog Computing Infrastructures under Probabilistic Delay Guarantees

I. Stypsanelli et al. [13] propose that capacity planning in fog computing can be presented as a Mixed Integer Linear Programming (MILP) problem. MILP problem refers to a problem where some of the variables are bound to be integers while others are allowed to be non-integers. They showed that by minimizing the probability where the processing time of the microdata center and the networking time of the request is greater than the maximum acceptable processing time, can be formulated as a MILP problem. The results of their experiment showed that notable cost savings can be obtained using this method.

This method is designed for fog computing hence it could be utilized in vehicular fog computing. But still, there is one major limitation. When it comes to vehicular fog computing, there should also be planning on how many vehicular fog nodes should be distributed into different urban areas. One of the strengths that vehicular fog computing provides, is mobile fog nodes which enables the service provider to control the number of fog nodes via vehicles between different areas. In order to use this strength fully, the predictions and models of future traffic should be made. With the help of the traffic models, the algorithm could be used separately for different areas in order to find the optimal capacity for vehicular fog nodes.

3.3 QoS-oriented capacity planning for edge computing

M. Noreikix et al. [9] propose a capacity planning method for edge computing where they use a knapsack algorithm in order to minimize the capacity. They research a hybrid edge cloud where the traffic is divided between edge nodes and the cloud. To determine resource requirements for applications, they profile different tasks and use benchmarking tools. For profiling, they determine what kind of resource demands various tasks have. For example, do the tasks have a need for CPU, GPU or both. They also profile demands for networking latency and bandwidth. Using this information and the knapsack algorithm, they determine what tasks

should be executed at the edge or in the cloud and the required capacity estimation for the system. The results show that executing the latency-sensitive tasks at the edge improved QoS and the non-latency-sensitive tasks could be executed in the cloud without reducing QoS.

This method could definitely work for vehicular fog computing. The profiling of tasks could yield to improvement in QoS because then the latency-sensitive tasks could be executed at the edge and the non-latency-sensitive tasks could be offloaded to the cloud. Especially the task offloading to the cloud could be beneficial as all the applications do not require minimal latency. Besides, the profiles of tasks could help to decide what kind of configuration of hardware is needed for fog nodes. Nevertheless, the presented method has the same limitation as the method presented in section 3.2 which means that the predictions and models could be very useful.

4 Discussion

A lot of research in fog computing has already focused on finding optimal locations for fog nodes at the edge of the network [15] [6]. For vehicular fog computing, it is not important to find exact locations for vehicular fog nodes but rather it should be focused on finding the optimal density of vehicular fog nodes in specific areas at different times of the day.

S. A. A. Shah et al. [10] present an overview of the 5G building blocks in the context of vehicular communication. According to the article, vehicles will have benefits from 5G at the application and system levels. Also, the usage of 5G with technologies, such as mobile edge computing, was shown to fix some of the deficiencies of 802.11p. However, the benefits of 5G could only be obtained as long as requirements for the vehicular applications were well defined.

5G shows a lot of promise in the context of vehicular fog computing. As cellular networks are moving towards 5G, it means that profiling the tasks of vehicular applications for capacity planning as in the work [9] could be very beneficial since benefits on the networking side can also be achieved. The data rate of 5G is also peaking at least 1Gb/s [2] which also benefits vehicular applications since they can generate gigabytes of data for processing.

Predictions and models for future traffic would definitely be helpful for capacity planning. The current research, especially works [13] [9] re-

viewed at sections 3.2 and 3.3 could very well be utilized for capacity planning in vehicular fog computing if the amount of end-users were well defined between various areas at different times of day. Although fog nodes could be installed on vehicles such as busses and taxis as in work [14] which could guarantee enough capacity during rush hours and also during more quiet hours. Nevertheless, it might lead to too much or too little capacity at the edge of the network if the fog nodes were installed on every bus and taxi. The models of future traffic might help to decide to what extent capacity is need because using only busses and taxis might lead to over or underestimation of capacity.

Service providers promise often good QoS around the clock but for example, during the night there might be very little or no traffic at the edge of the network. Then it might not be efficient to distribute vehicular fog nodes to those areas at those times because the chances are that there is very little or no traffic at all. In these situations, the option of deploying fog nodes to static roadside units in addition to mobile vehicles should be considered. This option could be an easy and efficient way to guarantee the minimum required capacity at the edge. Then vehicular fog nodes could be distributed to different areas on demand according to their needs.

5 Conclusion

This paper reviewed different capacity planning methods used in cloud, fog or edge computing and it researched their feasibility and limitations in vehicular fog computing. Vehicular fog computing is still a visionary concept and capacity planning for it is still at its early stages. Current research on capacity planning in cloud, fog and edge computing can be utilized for vehicular fog computing but nevertheless, they have their limitations. Ever-changing traffic makes it more difficult to utilize these methods because the capacity plan has to be optimized between different areas around the clock.

Current research for capacity planning in cloud, fog and edge computing has already shown promising methods. As for future work, there are many directions where to go. A capacity planning method that utilizes predictions and models for future traffic should be researched and also the option of deploying fog nodes into both static roadside units and vehicles should be researched as it might be a convenient way to guarantee minimum capacity at the edge of the network. Also profiling different

tasks for capacity planning appears to be promising since improvements to the networking side could be achieved, especially while using 5G.

References

[1] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments. *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pages 1–51, 2010.

[2] G. P. Fettweis. 5G and the future of IoT. In *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, pages 21–24, 2016.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[4] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen. Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873, 2016.

[5] C. Huang, R. Lu, and K. R. Choo. Vehicular fog computing: Architecture, use case, and security and forensic challenges. *IEEE Communications Magazine*, 55(11):105–111, 2017.

[6] M. Jia, J. Cao, and W. Liang. Optimal Cloudlet Placement and User to Cloudlet Allocation in Wireless Metropolitan Area Networks. *IEEE Transactions on Cloud Computing*, 5(4):725–737, 2017.

[7] Y. Jiang, C. Perng, T. Li, and R. Chang. Self-Adaptive Cloud Capacity Planning. In *2012 IEEE Ninth International Conference on Services Computing*, pages 73–80, 2012.

[8] T. Mekki, I. Jabri, A. Rachedi, and M. Ben Jemaa. Towards Multi-Access Edge Based Vehicular Fog Computing Architecture. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2018.

[9] M. Noreikis, Y. Xiao, and A. Ylä-Jaäiski. Qos-oriented capacity planning for edge computing. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, 2017.

[10] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally. 5G for Vehicular Communications. *IEEE Communications Magazine*, 56(1):111–117, 2018.

[11] Mehdi Sookhak, F. Yu, Ying He, Hamid Talebian, Nader Safa, Nan Zhao, Khurram Khan, and Neeraj Kumar. Fog Vehicular Computing: Augmentation of Fog Computing Using Vehicular Cloud Computing. *IEEE Vehicular Technology Magazine*, 12:55–64, 09 2017.

[12] I. Stojmenovic and S. Wen. The Fog computing paradigm: Scenarios and security issues. In *2014 Federated Conference on Computer Science and Information Systems*, pages 1–8, 2014.

[13] I. Stypsanelli, O. Brun, S. Medjiah, and B. J. Prabhu. Capacity Planning of Fog Computing Infrastructures under Probabilistic Delay Guarantees. In *2019 IEEE International Conference on Fog Computing (ICFC)*, pages 185–194, 2019.

[14] Y. Xiao and Chao Zhu. Vehicular fog computing: Vision and challenges. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 6–9, 2017.

[15] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo. Efficient Algorithms for Capacitated Cloudlet Placements. *IEEE Transactions on Parallel and Distributed Systems*, 27(10):2866–2880, 2016.

Advances in Attesting Run-time and Operational Behavior

Tianshi Xiang
tianshi.1.xiang@aalto.fi

Tutor: Thomas Nyman

Abstract

Run-time attestation is a security service which enables a trusted party (verifier) to authenticate information about the execution state of software running on a remote device (prover). Run-time attestation for embedded systems is a topic actively researched in recent years. Although different attestation schemes for resource constrained embedded systems have been proposed by researchers, there are few comparative research about the proposed solutions. This seminar paper conducts a literature survey of available solutions for run-time attestation and presents a comparison of their advantages, limitations, design and implementation technique as well as their performance.

KEYWORDS: remote attestation, embedded system security

1 Introduction

Embedded systems are ubiquitous in people's life. They can be anything from mobile phones to automotive systems, from medical devices to critical infrastructure, and even smart light bulbs. Several major cyber attacks [15, 13, 3], especially the Mirai botnet launched in 2016, made se-

curity concerns about embedded systems urgent.

Due to limited resources available to embedded systems, many security solutions feasible on general-purposed computers or data centers, are not suitable for embedded systems. According to the Internet Engineering Task Force (IETF), minimal resource required by Class-1 IoT devices are 10kB RAM and 100 kB Flash memory [4].

Remote attestation of embedded systems is a prominent security measure that enables a trusted party: the *verifier*, to verify if software on another device: the *prover*, is in a particular state.

Solutions for remote attestation can be based on either static features, such as measuring the program binary, or dynamic features, which describe the behavior of target program at run-time. Static features based attestation authenticates the integrity of the loaded program binary. Dynamic features based attestation can carry out verification of operational behavior based on the run-time state, such as control flow, number of loop iterations and so on.

The goal of this paper is to conduct a literature survey of available run-time attestation solutions for special-purpose and resource-constrained embedded systems. Solutions surveyed in this paper includes C-FLAT [2], LO-FAT [10], ATRIUM [22], LiteHAX [9] and OAT [21].

The paper is organized as follows: Section 2 describes background information as well as basic terminology and adversarial models. A survey of selected run-time attestation solutions is presented in Section 3. After that, Section 4, which is the main contribution of this paper, contains the comparison and evaluation of available solutions. Finally, Section 5 offers concluding remarks based on the survey.

2 Background

In this section, we present relevant background information and terminology related to embedded systems security.

2.1 The state of Embedded Device Security

Attacks such as Stuxnet discovered in 2010 and the Mirai botnet launched in 2016 put embedded systems security into the headlines of mass media. The landscape of embedded systems security are described by [24, 12].

Attackers are motivated by the data collected by embedded devices as

well as the combined computing and network power by massive numbers of vulnerable devices. With more and more embedded devices connected to the internet, the internet connectivity removes the physical access restrictions for adversaries. Research [8, 5] shows that vulnerabilities in firmware may lead to run-time exploitation.

2.2 Adversarial Methods

As mentioned in section 2.1, the connected embedded systems attract attackers. Here we introduce a few attack methods.

Return-oriented programming (ROP) [20], which is a type of code reuse attack. ROP corrupts memory during target program execution [19, 23]. ROP dynamically generates malicious code by chaining together code snippets of benign code without injecting any malicious instructions [23]. They modify the heap and stack of target program to divert execution flow. ROP attack can be utilized both in x86 architecture [20] and embedded processor [14].

Control-data attacks alter control flow of target program.

Non-control data attacks [6] do not alter control flow, instead, they corrupt application data such as user identity, configuration data and decision making data. These types of attacks corrupt program data, but not code pointers [18]. The attacked program can execute permissible but higher privileged control flow which is not intended for this program's run-time state. Chen et al. [6] showed in 2008 that these types of attacks are realistic.

Data-oriented programming (DOP) [11], is an attack to simulate expressive computations on the program memory, without exhibiting any illegitimate control flow [11]. In DOP, the adversary carefully corrupts non-control-data to build up sequences of operations without modifying the program's control flow [18].

2.3 Run-time Attestation

As shown by Figure 1, run-time attestation is typically implemented via a challenge and response protocol between a verifier and a prover [2]. The protocol works as following steps: 0) Verifier carries out one off-line measurement of the program to be attested and stores the result. Necessary features such as control flow information, value of critical variables, number of loops during the execution of program are measured. The measured

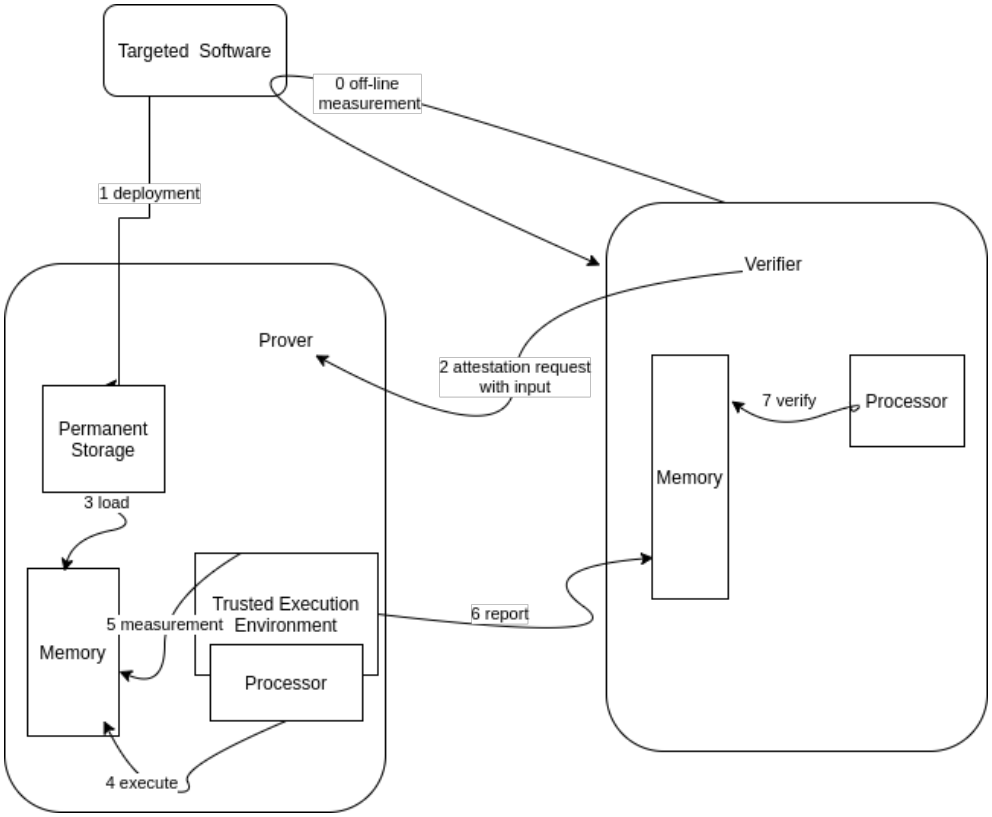


Figure 1. Run-time Attestation Protocol in General

features can be represented compactly by hashes. 1) Target software is deployed to the prover. 2) When attestation is requested, the verifier sends desired input and a nonce (for freshness) to the prover. 3) The prover loads target program into memory for execution. 4) The prover executes the program with received input, as well as potentially additional malicious input. 5) Corresponding features during the execution are measured by the prover. 6) Then the prover signs the measurement result represented by hash and the received nonce with it's private key as an attestation report. The report is sent back to verifier. 7) The verifier authenticates the report with prover's public key and nonce. The measurement value from received report is compared with off-line measurement for verification.

2.4 Standard Assumption

We have basic assumptions when discussing run-time attestation for embedded systems.

- Critical information such as private key of prover, attestation scheme code, etc are well protected inside trust anchor.

- We consider solutions for software-only attacks in our survey with an exception of ATRIUM [22].

2.5 Path Explosion

The hash measurements over all control-flow and data-flow events will result in a combinatorial explosion of the number of valid hash measurements, even if without considering the order of execution and iteration counts for loops [9].

3 Survey

This chapter summarises run-time attestation solutions surveyed in this survey paper. We elaborate the strengths and weaknesses of the selected solutions. We focus on comparison of run-time attestation solutions for resource-constrained embedded devices in this survey. So run-time attestation proposals based on *Trusted Platform Module* (TPM), which are too complex and expensive to deploy on embedded devices, are not included here.

Other commonly seen solutions such as *control-flow integrity* (CFI) [1], *code-pointer integrity* (CPI) [16], *code randomization* [7, 17], which can detect control flow attacks but can not provide any extra information. are not included in this survey neither.

Following sections describe the selected solutions for this literature survey.

3.1 C-FLAT

C-FLAT [2] is the abbreviation of Control-FLow Attestation for Embedded systems software.

This scheme measures execution path with a fine-grained control flow measurement. It is the first control-flow attestation scheme ever proposed and implemented [10]. Execution flow is measured by recording taken indirect branches. C-FLAT provides protections against certain level non-control data attacks which redirect the control flow to a high-privileged routine. These kinds of protections can not be provided by CFI.

C-FLAT also has limitations. 1) Software binary instrumentation is required by the scheme. This instrumentation requires compiler support

and leads to lack of legacy software support. 2) There is high performance overhead on prover devices. 3) This scheme is prone to CFG refactoring[1] (if without binary integrity check) 4) C-FLAT can not detect data only attack which does not modify control-flow [21] 5) Hash measured on prover device may not be verifiable at verifier due to program path explosion issue [21].

3.2 LO-FAT

LO-FAT [10] is the abbreviation of Low Overhead control Flow ATtestation in hardware.

This scheme is the first practical hardware-based approach to control-flow attestation [10]. Granularity of control-flow tracking (ie. max number of branches to track per loop) is configurable for a light weight version of LO-FAT.

Source and destination address pair (Src, Dst) of each executed branch instruction is measured during the target program execution at prover. Other information including auxiliary loop metadata about executed path in each loop, order of first occurrence of each executed path, iteration number per loop path, indirect branch target are measured as well to describe the run-time state of the target program.

Because recording of run-time information is performed by a separate hardware in parallel to the prover's main processor, the attestation performance is improved.

Heuristic in how to distinguishing backward branches is provided by LO-FAT: non-linking backwards branch is loop, because subroutine calls usually update link-register.

3.3 ATRIUM

ATRIUM [22] is the abbreviation of ATtestation ResIlient Under Memory. It is the first scheme to address physical attacks towards prover's memory without instruction set extension or code instrumentation [22]. This scheme can detect if the code in memory is replaced by adversary. The code replacement can be achieved for example by splicing in new memory with malicious code on memory bus. The detection is done by attesting executed instructions that extracted from the execution stage of processor pipeline.

[1] https://github.com/xoreaxeaxeax/REpsych

ATRIUM is inspired by the above surveyed 2 schemes with consideration about TOCTOU[2] problem, as one of the author for ATRIUM is also co-author for C-Flat and LO-FAT.

Executed instructions and control-flow path are measured by ATRIUM. Hash values of loop paths are calculated separately from that of non-loop code segments to avoid the path explosion issue.

Comparing with LO-FAT, ATRIUM reduces overhead by removing loop measurement information from on-chip memory to a content-addressable memory (CAM). Hash values are still stored at an on-chip dedicated memory.

With the conclusion reached, the authors of ATRIUM also agree with regards to LO-FAT that for obvious performance reasons, attestation based on recording every executed instruction is not practical. It is recommended to apply executed instruction attestation to pre-defined security-critical code regions.

3.4 LiteHAX

LiteHAX [9] is the abbreviation of LIghTwEight Hardware-assisted Attestation of program eXecution. It is the first run-time attestation scheme that verifies both control flow and data flow of target program for *Reduced Instruction Set Computer* (RISC) based embedded devices. It can detect attacks which do not modify control-flow of the target program on prover.

Data flow attestation is based on the intuition, that for RISC architecture, all known and reported DOP and non-control data attacks only corrupt memory *load* and *store* operation. Detection of data-flow attack is achieved by a short digest of memory access operations.

The representation of the prover's control flow is encoded by a bitstream and sent to verifier during the execution, which enables continuously monitoring of the target code execution on the prover. Instead of comparing measured run-time states with pre-calculated hash measurements, LiteHAX carries out attestation by performing symbolic execution and data-flow analysis constrained to activated segment of the control flow graph of the target program. Thus, execution path explosion problem is mitigated by restricting the attestation scope.

In the proof of concept implementation, dynamic symbolic execution is

[2]https://cwe.mitre.org/data/definitions/367.html

implemented on top of a Python framework angr[3]. CoreMark[4] was used for performance evaluation.

There is limitation of symbolic execution. Theoretically, symbolic execution may fail to resolve a symbolic expression which is required to generate data memory access [9].

3.5 OAT

OAT [21] is the abbreviation of Operation ATtester.

OAT introduced a new property, "Operation Execution Integrity (OEI)", which is a new security property for embedded devices. OEI covers the integrity of both control-flow and critical data-flow of the target program execution.

There is no need for verification engine at verifier to pre-compute the path. Verification engine uses a disassembler[5] to disassemble the target binary and carries out an abstract execution.

Control-flow integrity of target program is measured with the combination of hashes of back-ward edges in control-flow graph and the execution trace of 3 different types of forward-edges (direct jump, conditional branch and indirect transfers) in control-flow graph. Verifier carries out attestation by abstract execution of the binary code guided by forward-edge traces. Thus OAT reduces the amount of information needed for verification.

Data-flow integrity is measured by the define-use consistency of critical data, to trace if the value of critical variable at use is same as the value at previous define. Hence it reduces the overhead of checking every memory-write instruction. Beside, data integrity is only enforced for critical variables.

Obviously, binary instrumentation requires compiler support and the scheme does not support legacy software. Beside these two limitations, developers are required to define the scope of operations and to annotate critical variables.

---

[3] http://angr.io/

[4] https://github.com/eembc/coremark

[5] http://www.capstone-engine.org/

	C-FLAT	LO-FAT	ATRIUM	LiteHAX	OAT
ROP [a]	●	●	●	●	●
Non-control data [b]	●	●	●	●	●
Control-flow [c]	●	●	●	●	●
Path explosion [d]	○	●	●	●	●
Compiler support [e]	●	○	○	○	●
HW support [f]	○	●	●	●	○

Table 1. Comparison of conducted run-time attestation solutions

[a] symbol ● indicates the scheme is effective under return-oriented programming attack
[b] symbol ● indicates the scheme is effective under non-control data attack
[c] symbol ● indicates the scheme is effective under control-flow attack
[d] symbol ● indicates the scheme can limit the impact of path explosion problem
[e] symbol ● indicates the scheme requires compiler support for binary instrumentation
[f] symbol ● indicates the scheme requires custom hardware extension

4 Comparison

In this section, we present the result of our comparison of the surveyed run-time attestation. We aim to compare the chosen measurement techniques of each scheme, as well as their respective adversarial models of each scheme with each other.

The following aspects are compared: 1) Under which adversarial models do they remain effective, 2) how each scheme is designed and implemented, 3) and the underlying hardware platform for proof of concept implementation. This survey does not compare the performance and overhead of each solution.

Table 1 provides an overview of each attestation scheme. It shows comparison about 1) what kind of attack can each scheme detect, 2) does the target software application need instruction instrumentation at compile time, 3) and does the scheme solve the execution path explosion problem.

We can see from Table 1 that the surveyed scheme either needs compiler support for binary instrumentation or needs customized hardware extension support. Techniques for information measurements have been developed over time to limited the impact of combinatorial explosion of the state space.

Table 2 compares techniques how each scheme handles loop measure-

ment and the choice of hash algorithm. Hash algorithms are either black-2 family implementation or SHA-3 family implementation.

For loop measurement, C-FLAT treats loops as subroutines. LO-FAT hashes each loop path once and keeps a loop iteration counter for each unique loop. ATRIUM splits loop into separate segments from basic control flow. LiteHAX utilizes symbolic execution and data flow analyse constrained to the current active execution segment. OAT treats loop condition variables as critical data and assert it with data integrity.

	Loop Handling	Hash Algorithm
C-FLAT	as subroutine	Blake-2
LO-FAT	loop path (Src, Dst) + loop counter	SHA-3 512
ATRIUM	loop path (Src, Dst),+ loop counter + depth	Blake-2
LiteHAX	every executed instruction	SHA-3 512
OAT	hybrid scheme (loop variables)	Blake-2s[a]

Table 2. Comparison of Loop Information Measurement and Hash Algorithm Choice

[a] optimized for 8 - 32 bit platform

Table 3 compares what hardware platform are used for the proof of concept for each scheme, as well as the corresponding trust anchor. Prototype of C-FLAT and OAT are implemented on off-the-shelf hardware, while others use custom hardware extensions.

	Hardware in PoC	Trust Anchor
C-FLAT	Raspberry Pi 2 (ARM)	TrustZone A
LO-FAT	RISC-V Pulpino + Virtex-7 FPGA	Custom HW Extension
ATRIUM	RISC-V Pulpino + Virtex-7 FPGA	Custom HW Extension
LiteHAX	RISC-V Pulpino	Custom HW Extension
OAT	HiKey (Kirin 6220 ARM A53)	TrustZone A

Table 3. Hardware for Proof of Concept Implementation

5 Discussion

We conducted a literature survey on 5 prominent run-time attestation solutions for embedded systems.

From the above surveyed solutions, we conclude that measurement of execution states is the main challenge in run-time attestation. Different

techniques have been proposed, with corresponding advantages and limitations.

The measurement requires computation tasks. These tasks can be performed with off-line supports such as binary instrumentation by compilers or during run-time with additional hardware extensions support like others.

Attestation schemes also have to handle the execution path explosion problem during measurement of execution states. This is an inherent problem from target program execution. Different approaches has different scheme to limit the impact from path explosion.

Swarm attestation of large amount devices is a challenge which is not included in this survey.

References

[1] Martín Abadi, Mihai Budiu, Úlfar Erlingsson, and Jay Ligatti. Control-flow integrity principles, implementations, and applications. *ACM Trans. Inf. Syst. Secur.*, 13(1), November 2009.

[2] Tigist Abera, N. Asokan, Lucas Davi, Jan-Erik Ekberg, Thomas Nyman, Andrew Paverd, Ahmad-Reza Sadeghi, and Gene Tsudik. C-FLAT: Control-Flow Attestation for Embedded Systems Software. CCS '16, page 743–754. ACM, 2016.

[3] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. Understanding the Mirai Botnet. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1093–1110, Vancouver, BC, August 2017. USENIX Association.

[4] C. Bormann, M. Ersue, and A. Keranen. Terminology for Constrained Node Networks. Technical report, The Internet Engineering Task Force, Janaury 2014. https://tools.ietf.org/id/draft-ietf-lwig-terminology-05.html.

[5] Daming D. Chen, Manuel Egele, Maverick Woo, and David Brumley. Towards Automated Dynamic Analysis for Linux-based Embedded Firmware. *NDSS Symposium*, (February):21–24, 2016.

[6] Shuo Chen, Jun Xu, Emre C Sezer, Prachi Gauriar, and Ravishankar K Iyer. Non-Control-Hijacking Attacks are Realistic Threats. *Proceedings of the 14th USENIX Security Symposium*, pages 177–191, 2008.

[7] Frederick B. Cohen. Operating system protection through program evolution. *Computer and Security*, 12(6):565–584, October 1993.

[8] Andrei Costin, Jonas Zaddach, Aurélien Francillon, and Davide Balzarotti. A large-scale analysis of the security of embedded firmwares. *Proceedings of the 23rd USENIX Security Symposium*, pages 95–110, 2014.

[9] Ghada Dessouky, Tigist Abera, Ahmad Ibrahim, and Ahmad-Reza Sadeghi. LiteHAX: Lightweight Hardware-Assisted Attestation of Program Execution. In *Proceedings of the International Conference on Computer-Aided Design*, ICCAD '18. Association for Computing Machinery, 2018.

[10] Ghada Dessouky, Shaza Zeitouni, Thomas Nyman, Andrew Paverd, Lucas Davi, Patrick Koeberl, N. Asokan, and Ahmad-Reza Sadeghi. Lo-fat: Low-overhead control flow attestation in hardware. In *Proceedings of the 54th Annual Design Automation Conference 2017*, DAC '17. Association for Computing Machinery, 2017.

[11] Hong Hu, Shweta Shinde, Sendroiu Adrian, Zheng Leong Chua, Prateek Saxena, and Zhenkai Liang. Data-Oriented Programming: On the Expressiveness of Non-control Data Attacks. *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, pages 969–986, 2016.

[12] Sergey Iskhakov, Alexander Shelupanov, and Artur Mitsel. Internet of things: Security of embedded devices. *RPC 2018 - Proceedings of the 3rd Russian-Pacific Conference on Computer Technology and Applications*, pages 2018–2021, 2018.

[13] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. DDoS in the IoT: Mirai and Other Botnets. *Computer*, 50(7):80–84, 2017.

[14] Tim Kornau. Return Oriented Programming for the ARM Architecture. Master's thesis, Ruhr-Universität Bochum, 2009.

[15] D. Kushner. The real story of stuxnet. *IEEE Spectrum*, 50(3):48–53, 2013.

[16] Volodymyr Kuznetsov, Laszlo Szekeres, Mathias Payer, George Candea, R. Sekar, and Dawn Song. Code-pointer integrity. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 147–163. USENIX Association, October 2014.

[17] Per Larsen, Andrei Homescu, Stefan Brunthaler, and Michael Franz. SoK: Automated software diversity. *Proceedings - IEEE Symposium on Security and Privacy*, pages 276–291, 2014.

[18] Thomas Nyman. *Toward Hardware-assisted Run-time Protection*. PhD thesis, Aalto University, Oct 2020.

[19] Ryan Roemer, Erik Buchanan, Hovav Shacham, and Stefan Savage. Return-oriented programming: Systems, languages, and applications. *ACM Transactions on Information and System Security*, 15(1):1–34, 2012.

[20] Hovav Shacham. The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86). *Proceedings of the ACM Conference on Computer and Communications Security*, pages 552–561, 2007.

[21] Zhichuang Sun, Bo Feng, Long Lu, and Somesh Jha. OAT: Attesting Operation Integrity of Embedded Devices. *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1433–1449, 2020.

[22] O.Arias D.Sullivan A.Ibrahim Y.Jin S.Zeitouni, G.Dessouky and A-R.Sadeghi. ATRIUM: Runtime attestation resilient under memory attacks. *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, 2017-November:384–391, 2017.

[23] Laszlo Szekeres, Mathias Payer, Lenx Tao Wei, and R. Sekar. Eternal war in memory. *IEEE Security and Privacy*, 12(3):45–53, 2014.

[24] John Viega and Hugh Thompson. The state of embedded-device security (Spoiler Alert: It's Bad). *IEEE Security and Privacy*, 10(5):68–70, 2012.

Usability Analysis of Devices in an IoT ecosystem

Sayed Farhan Amjad
sayedfarhan.amjad@aalto.fi

Tutor: Amel Bourdoucen

Abstract

Technology is penetrating everyday lives at an accelerated rate, it is essential to understand how the modern smart devices can provide benefits to the users while ensuring their safety from the inherent security challenges associated with the internet. This paper will discuss the fundamentals of IoT, how IoT can contribute in improving lives, and a brief analysis on the negative aspects of increased exposure to the internet that can lead to potential harm.

KEYWORDS: IoT, Smart Homes, Intelligent Systems, Cloud, Privacy, Low Latency Networks, Anonymity

1 Introduction

The Internet of Things (IoT) is a recent communication paradigm that envisions a future in which the objects of everyday life will be equipped with microcontrollers, transceivers for digital communication. These digital objects will be able to communicate among themselves as well as with the users, becoming an integral part of the Internet [2]. People communicate with the medium of internet using their personal computers and

smartphones but it is about to change because of IoT. Internet is going to become much more immersive and pervasive. Transformation is happening in many key industry and service sectors because of the technological advancements IoT is bringing to the market such as, Smart Governance, Smart Mobility, Smart Utilities, Smart Buildings, and Smart Environment [1].

In recent decades, consumer electronics became popular in almost every household, hospital, educational institute, and workspace all around the globe. Most of the users will lose if not already the track of the number of items they own containing a microprocessor and integrated circuits [3].

In the near future, an average person will have hundreds of devices capable of collecting information, communicating over the internet, and they will become a part of the routine life. At present, it is difficult to imagine living without smartphones as users have become too dependent on the easy to use and valuable services. In the future, these IoT sensors and devices will become an extension of the five human senses making it hard to move away from the benefits provided by them.

The Usability and fast adaptation of these devices are dependent on the user interface and user experience that these devices offer. Fast response time and correct information are essential for the system to seem realistic. We will discuss some of the major use cases and potential risks involved with these technologies is further detail.

First section discusses the internet of things, Section 2 deals with the different use cases in benefits of these smart devices. Section 3 and 4 are related to the potential safety risks and how can these risks be avoided by following the best practices. Finally, the last section will conclude the paper based of the advantages and disadvantages analyzed in the previous sections.

2 Use cases of IoT

IoT is penetrating almost every field of human life just like the transformation caused by personal computers in the past few decades. We will discuss some key fields that are transforming by microprocessors.

2.1 Entertainment

Drones, Smart Bulbs, connected speakers, and smart home entertainment systems are a few examples of the contribution of IoT in this industry. Data collected by the devices and sensors help users in getting more personalized interaction with the content and companies can produce and target directly to the personal preferences. When interacting with smart TV with the help of voice commands, rewinding, fast-forwarding, muting, paying attention to the content by not talking to one another, the TV can analyze and extrapolate the personal preferences without the need of an explicitly input [18]. As the gap between a person and their digital self closes, users will be offered more personalized services by the service provides, and as a result, customer satisfaction will improve.

2.2 Farming and Agriculture

The concept of drip irrigation [19] is continuously gaining popularity as the swear consequences of the climate crisis are starting to show up. Moisture sensors in the soil and software-controlled watering systems and helping us achieve maximum yield with the least amount of water from agricultural lands that are prone to swear droughts. Drowns are used to spray the fields with pesticides and technology is also playing a great role in quality control and effect transportation of food items with short shelf life.

2.3 Medical and Health

Wearable devices available in the market can measure blood oxygen level, hearts electrocardiogram, heart rate, SpO_2 level, stress level. In the future, these devices will have considerable more capabilities. A watch on the wrist with GPS and cellular connectivity can call emergency services in case an accident or the person losing consciousness.

Cardiac arrest (is a sudden loss of blood flow) is one of the prime factors for a breathing disorder [7]. Mild cardiac arrest can be identified using an echocardiogram [7]. Wearable devices equipped with necessary sensors can detect such conditions in advance and inform the emergency contacts and nearby medical services to save lives.

The contagious disease such as corona virus spread when infected asymptomatic person is coming in proximity of healthy people; as a result the

virus is transferred without any detection. In order to track and notify people about the possible exposure to virus, smartphones and wearable device equipped with low energy wireless radio are used to keep track of people who are coming in contact with one another, and later the same information is used to notify for possible exposure. As a result, a seemingly difficult problem of tracking the invisible virus is resolved by a small piece of software installed in the existing devices.

Assisted living is a concept where technology helps people in the trivial tasks of everyday live specially in senior care. Elderly can communicate with their loved ones with a click of a button. Caretakers can monitor and respond to their needs instantaneously. Medication management is done digitally, and it eliminates the possibility for human error there where it is critical for a patient to strictly follow the prescribed medicine. The human body has multiple different physiological signs that can be measured: from electrical signs to biochemical, human bio-signals are possible to be extracted and be used to better understand the bodily health status and reaction to external factors [8]. Passively monitoring these vital signs with the help of wearable devices can save lives of critical patients.

2.4 Security

Drone technology, security cameras, and facial recognition technology help security agencies fight crime and make the neighborhoods safer. Small inexpensive quadcopter drones can fly loge distances and can be used in rescue operations.

Digital forensics is the process of identification, extraction, and documentation of computer evidence which can be used by security agencies to track down criminals and help the process of justice [22]. The devices are tracking activities and logging them with time stamps that can be used to reconstruct the timeline of different events, serving as evidence, or for validating the statements made my the witnesses in the court of law.

2.5 Energy

The energy demand is growing exponentially, and the world is in a period of transitioning from the sources of energy generation that burn petrochemicals to a better reusable and sustainable future. It is necessary to improve the efficiency of the energy consumption processes and tech-

nology is playing an essential role in achieving that. Governments have recognized a need for modernizing the electric energy system and establishing such Smart Grids around the world [4]. Tools such as Autobidder uses advanced machine learning algorithms to decentralize the electric grid.

Sensors are used on the roads to detect traffic and accordingly turn the road lights on when needed. In the houses, the smart thermostats controlled by artificial intelligence knows when owners will arrive home and efficiently set the temperature just in time and save electricity.

2.6 Transportation

The transportation industry is also benefiting heavily from modern devices. Data shared by the users driving all over the globe help us see real-time traffic information that is crucial for planning our routes and reaching out destinations on time. Traffic police can monitor traffic from the control room and redirect drivers automatically. Cameras and sensors can detect traffic rule violations and penalize the drivers electronically. Traffic lights are controlled according to the traffic on the road, digital road signs and smart navigation devices provide warning and suggestions in advance to make driving safer.

In the future, people will be driven by autonomous vehicles that make use of advanced artificial intelligence. There are various semi-autonomous features already introduced in modern cars such as, lane-keeping, automatic braking, and adaptive cruise control are based on such systems. It is predicted that most companies will launch fully autonomous vehicles in the coming years. The future of autonomous vehicles is an ambitious era of safe and comfortable transportation [6].

3 Potential risks of IoT

Given the functional requirements of privacy, maximal anonymity might not be practically useful for individuals and could circumvent accountability. Privacy is a universal requirement that enables individuals to maintain different types of interactions, for some individuals the consequence of a breach of privacy protection could be psychological or physical harm, including death [5].

3.1 Data Breach and Hacking

With increasing amounts of data being generated and stored in a variety of environments, the probability of a breach in the security of our databases is also increasing. Data will become increasingly more valuable. Basic personal information that is usually termed as useless in the hands of criminals can be life threatening.

3.2 IoT Botnets

IoT botnets are a group of hacked computers, smart appliances and Internet-connected devices that have been co-opted for illicit purposes [13]. Hackers gain access to these devices and wait for the right time to plan and execute a target attack or sell the access on dark web. A DDos attack happened in 2016 that took many of the top websites down, including Twitter and Pinterest. The attackers used more than twenty-five thousand CCTV cameras from ninety-five different countries to generate more than fifty thousand web requests per second, as a result the servers were overloaded, and it took hours for the services to get restored [11].

Hackers can create artificial spikes on the electric grid by switching on the high energy appliances at once and cause widespread blackouts that can follow by large scale vandalism or target attacks on the valuable infrastructure when the security and surveillance systems are shutdown because of the break in electricity supply.

3.3 False Digital Evidence

Digital devices log events from the begining of the day to the time the person goes to bed, every single activity is recorded. The digital alarm device knows when the wake up alarm was set, The smart bulb in the washroom, every time the smart fridge is opened, the smoke detector in the kitchen, smart door lock, smart car, social media feeds, check-ins, use of digital payment methods, digital communications, and all the scheduled appointments in the digital calendar, every single thing is generating a digital trails. If a malicious entity wants to plant a false evidence against an innocent person, altering the logs and adding false information can turn that person into a suspect of a serious crime.

3.4 Wardriving

It is a process of geotagging different wireless access points that can later be used to identify the location of any digital device connected with them [10]. The location can later be used for targeted attacks. Suppose there is as unsecure IoT device in a network and someone can remotely gain access. Unauthorized access and location data from publicly available achieves obtained from wardriving can be combined. The attacker can exploit such vulnerabilities possibly deactivate the security system and unlock the smart door locks and steel valuable items without any alarm going off.

3.5 Tracking and Spying

The technology used to track the exposure to Corona Virus can also be used by malicious entities to trace anyones movements and social connections. A small low energy Bluetooth device can log the unique media access code address of any other Bluetooth enabled device that comes in its vicinity. With a little correlation analysis and analytics with the information form other digital service, someone can extract movements and social connections of the subject and start to exploit in numerous ways.

The cameras, microphones, and other very sensitive sensor suite in the electronic devices can allow someone to track the user's moments from a distance. The private communications, banking details, personal data, and preferences can be stolen and processed without victim's knowledge. Such detailed information can be used to generate behavioral models that can predict a persons future decisions, and it can also be used in a targeted campaign to brainwash masses use them to nefarious purposes with them realizing that they are being used. Cambridge Analytica was a company that specialized in changing the opinions of general public using volumes of data and running targeted advertisement [17].

3.6 Safety of Children

The toys for little children equipped with wireless radios antennas that let them connect with the apps on the smartphones and sometimes directly to the cloud services. Lack of awareness about cyber security and unwillingness of manufacturers to invest in enhancing the security of these relatively inexpensive toys leave out children vulnerable. Several years

ago, a smart doll named Cayla was banned in Germany because it of its potential of being used as a spying tool, as it had a built-in microphone and the ability to transmit data wirelessly [9].

3.7 Ransomware

Cyber criminals can hold the victim's important personal or business data by encrypting it, locking users out of their own devices and charge you ransom for returning the data back. Sometimes business must pay millions for dollars to regain access to their data back from these criminals [15]. Attackers would ask for payments in crypto currencies resulting in minimizing the possibility of law enforcement agencies tracking them.

It is always a better to invest in enhancing the security infrastructure of business, hospitals or governmental organizations then to wait for a security breach. A ransomware attack can be triggered with a single phishing email, a pop-up or a message with embedded code.

3.8 Privacy

One of the main concerns regarding the adaptation of modern technology is the amount of data being collected by large corporations about the personalities, habits, likes, and dislikes of their consumers. Every internet user has a digital footprint that is going to live forever on cloud servers somewhere in the world. These companies know more about us then we know ourselves and it is an alarming situation. The targeted ads that naive internet users cannot resist, personalized content recommendation encouraging to keep consuming content have damaging effects on human brains. The world has become a global village and human beings are becoming increasingly isolated in their digital bubbles.

The privacy should be everyones right and companies should be regulated with sensible privacy laws. Sensible privacy refers to the combination of privacy tradeoffs between privacy and usability and privacy and accountability [5].

Identity theft, damaging reputation, targeted attacks, black mailing are few of the examples of the potential attacks after the invasion of someones privacy.

3.9 Governmental control

In some parts of the world, the tyrannical governments are controlling every aspect of the lives of their citizens. The idea of freedom is lost. You cannot communicate with anyone without governmental agencies listening and analyzing to the conversations [21]. Data generated for IoT devices can be misused by the corrupt elements as a result the user suffers.

4 How to be safe

Information and cyber security should become a part of education from an early age because in the future our personal information is going to be our most valuable assert and as the value of information grows the probability of attacks also grows in proportion. Human beings are the weakest link in any cyber-attacks [14]. A vast majority of internet users lack digital health training and are unwittingly exposing themselves to easily avoidable risks [14].

Kids are spending time on the internet and being exposed to all kinds of treats. They are the naïve targets of attackers would encourage them to click on links to get free games of win prizes. It is reasonable to teach them about cyber security at schools along with other essential subjects.

4.1 Use UpToDate software

The old devices with no security patches are the easiest target of attach. Regularly updating to the latest software version. Avoid using devices whose support period as passes and they do not receive security updates from the manufacturer.

4.2 Avoid buying unsecure devices

If an IoT device is inherently unsecure by design or the company making the device is not committed to providing security of its customers in order to save cost or any other reason, such products should be avoided.

4.3 Use separate network

Sometimes the data in a small inexpensive device is not so valuable so the manufacturers do not invest as heavily on the security side as they

should. But the vulnerability in that particular device can compromise the whole network providing an gateway to important sensitive personal data moving on the network, so it is a good practice to use a separate network for your IoT device from other more important gadgets.

4.4 Periodic Backups

It is a good practice to take periodic backups of your critical data and software, which not only help you restore operations after a cyber-attack about also protect you from other unexpected events such as hardware failures or natural disasters.

5 Conclusion

The privacy is as important as usability of IoT devices. It is important for governmental organizations to standardize frameworks that ensure safety and security of internet users. Unsecure devices should not be allowed to enter the market. Although the usability of these devices suffers when privacy is insured but it is necessary to find a balance between the two.

Once the security and privacy concerns are addressed the benefits of integrating everyday lives with smart devices that can monitor a person's health, wellbeing, provide security, and automate the trivial daily tasks will out way the short comings and will lay a foundation for the widespread adaptation where life without these devices will become undesirable.

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities," in IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22-32, 2014.

[2] L. Atzori, A. Iera and G. Morabito, "The internet of things: A survey", Comput. Netw., vol. 54, no. 15, pp. 2787-2805, 2010.

[3] Frank Stajano, Ross Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks University of Cambridge Computer Laboratory, Cambridge, UK

[4] Anderson, Roger & Ghafurian, Reza & Gharavi, Hamid. (2018). Smart Grid The Future of the Electric Energy System.

[5] Budi Arief, Kovila P. L. Coopamootoo, Martin Emms, Aad van Moorsel. Sensible Privacy: How We Can Protect Domestic Violence Survivors Without Facilitating Misuse

[6] Bimbraw, Keshav. (2015). Autonomous Cars: Past, Present and Future - A Review of the Developments in the Last Century, the Present Scenario and the Expected Future of Autonomous Vehicle Technology. ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings. 1. 191-198.

[7] Balashanmugam, Dr.Thiyaneswaran. (2020). Breathing Level Monitoring and Alerting by Using Embedded IoT. Journal of Green Engineering. 10. 29862994.

[8] Dias D, Paulo Silva Cunha J. Wearable Health Devices-Vital Sign Monitoring, Systems and Technologies. Sensors (Basel). 2018;18(8):2414. 2018.

[9] https://www.cnet.com/news/parents-told-to-destroy-connected-dolls-over-hacking-fears/

[10] https://en.wikipedia.org/wiki/Wardriving

[11] https://www.csoonline.com/article/3089298/iot-botnet-25-513-cctv-cameras-used-in-crushing-ddos-attacks.html

[12] Beltrán-García P., Aguirre-Anaya E., Escamilla-Ambrosio P.J., Acosta-Bermejo R. (2019) IoT Botnets. In: Mata-Rivera M., Zagal-Flores R., Barría-Huidobro C. (eds) Telematics and Computing. WITCOM 2019. Communications in Computer and Information Science, vol 1053. Springer, Cham.

[13] https://internetofthingsagenda.techtarget.com/definition/IoT-botnet-Internet-of-Things-botnet

[14] https://www.stormshield.com/news/should-we-be-teaching-cybersecurity-in-school

[15] https://www.npr.org/2020/10/29/928979988/u-s-hospitals-targeted-in-rising-wave-of-ransomware-attacks-federal-agencies-say

[16] Shen, Young-Young & Boppana, Abhishektha & Arquilla, Katya & Anderson, Allison. (2018). Wearable sensor suit system for quantifying human-spacesuit interactions. 1-13.

[17] https://www.theguardian.com/politics/2017/mar/04/nigel-oakes-cambridge-analytica-what-role-brexit-trump

[18] Kang, H.-S.; Park, M.-S., Kim, S.-J. Study on Smart TV Forensics. J. Korea Inst. Inf. Secur. Cryptol. 2014.

[19] https://en.wikipedia.org/wiki/Drip_irrigation

[20] Ali Montazar, Michael Cahn, and Alexander Putman Research Advances in Adopting Drip Irrigation for California Organic Spinach: Preliminary Findings, 2019.

[21] Strittmatter, K., We Have Been Harmonized: Life in China's Surveillance State, 2020.

[22] https://www.guru99.com/digital-forensics.html

Distributed and parallel rendering for gaming and XR

Reetu Kontio
reetu.kontio@aalto.fi

Tutor: Matti Siekkinen

Abstract

KEYWORDS: GPU, graphics, parallel rendering, distributed rendering, remote rendering, cloud gaming

1 Introduction

This paper is a literature review on the topic of distributed and parallel rendering. The research is focused particularly on rendering in games and mixed reality (XR). Furthermore, cloud gaming is discussed, as it is in close relationship with the before mentioned topics. As these topics focus heavily on real-time rendering, 3D modeling and other non-real-time rendering is left out of the scope of this paper.

The main solutions discussed here consist of different approaches to implementing rendering pipelines for distributed remote systems. To elaborate, the possibilities of asynchronous lighting processing and megatexture processing are examples of methods discussed in this review. These methods are evaluated and compared, and the paper discusses the implications and possibilities of them.

This paper also provides examples and presents discussion on existing

solutions and commercial services utilizing parallel and distributed rendering. Such examples are Google Stadia, GeForce SLI and GeForce Now by NVidia, AMD Crossfire and Vulkan by Khronos Group. Two of these, Stadia and GeForce Now, are discussed more in detail. The main research method used in this paper is reviewing and comparing existing solutions and research.

2 Background

As user expectations on high fidelity graphics are constantly increasing, the hardware requirements of devices are also increasingly expensive. User experience is diminished when they are required to run applications with reduced graphical fidelity or if they are forced to experience input lag. Furthermore, users with mobile devices or low-end hardware may not be able to utilize the applications with high-end graphical demand at all. [12, p. 10] [13, p. 255]

One approach on solving these challenges is to utilize remote parallel rendering. The most common method is to divide the tasks required for the application and sending the calculation-heavy parts to a server, where they are again divided by multiple parallel computing units, Graphic Processing Units (GPUs) in this case. Utilizing this method enables the client to focus on the tasks most crucial to user responsiveness. [13, p. 255]

The next subsections provide basic background for both parallel rendering and remote rendering individually and the reasons and problems behind these topics are discussed.

2.1 Parallel rendering

Graphical computing is very calculation intense. For example, a virtual reality application requires substantial amounts of floating point operations executed frequently [14, pp. 1-2]. Newest top-of-the-line graphic cards today are capable of performing tens of trillions of floating point operations per second (Teraflops, TFLOPS). However, this computing power may always be surpassed by adding graphics complexity. This is why demanding applications may utilize rendering clusters of multiple GPUs.

Parallelism in rendering may occur on different levels simultaneously. A single GPU utilizes parallelism between its cores. However, clustering multiple GPUs and utilizing them in a single application further en-

hances the performance. In a cluster of GPUs, the rendering task is split between the GPUs and then synchronized after each individual GPU has completed its task [9, p. 2].

One method of utilizing GPU parallelism is utilizing more than one GPU in one computer. There exists technologies such as NVidia SLI [10] and AMD Crossfire [4] which may be utilized in multi-GPU system to parallelize rendering. [18, p. 114].

2.2 Remote rendering

Regardless of the current state of top-of-the-line GPUs, most of the consumers are utilizing substantially more limited graphical processing capabilities. Examples of this are mobile and laptop users. Remote rendering might offer these users a solution for using high fidelity graphics applications, as it does not require the clients to possess a powerful GPU.

In a system utilizing remote rendering, a server performs the rendering and conveys the output to another device displaying it to the user. When the connection between the server and client device is interactive, the user is able to feed input into the client device, which in turn sends it to the server. The server then proceeds to render the output based on the input and then sends it back to the client device to be displayed to the user. [15, p. 1] Main hindrance in this setting is latency. The user experience is greatly diminished if the input latency rises above a certain level. [7, p. 1]

3 Parallel remote rendering

When rendering graphics in a parallelized remote system, the different parts of the pipeline may be distinguished into two parts: front end and back end. Front end contains the application and the scene to be rendered. The back end contains a master computer and a number of slave computers. [9, p. 2] In this communication between these two parts, the sorting of geometry primitives may be performed in different stages of the process [9, p. 2] [14, p. 2]. This section explains the functions of these parts and discusses different approaches to the sorting of geometry primitives.

3.1 Geometry primitive sorting

Geometry primitives are the simplest geometric units the system may process. When rendering, the system is required to sort the primitives and calculate their effects regarding the output image [14, p. 2]. This sorting may occur in different stages during the process. Three methods of sorting discussed here are: sort-first, sort-middle and sort-last [14, p. 2].

Sort-first approach distributes the primitives for processing units during the geometry processing [14, p. 3]. The color buffer of the image is divided into sections and distributed arbitrarily for the processing units [9, p. 2] [14, p. 3]. The basic concept is simple, but problems occur, if a primitive is located between the regions assigned to different processing units. In this case, these primitives are redistributed. However, this causes overhead. [14, p. 3] After each slave processing unit has completed their rendering task, they return their color buffers to the master processing unit, which then combines them resulting in the final image. [9, p. 2]

Sort-middle approach conducts the distribution of primitives in the middle of the pipeline, after geometry processing, but before rasterization. The parallel processing units then rasterize the regions of the output image dedicated to them. [14, p. 3] This method allows the passing of light rays, when using ray tracing techniques, enabling increased graphical fidelity [3].

Sort-last approach distributes the primitives in the end of the rendering pipeline. The rasterized image is divided in regions and distributed to the processing units, where the pixels are computed from the samples. [14, pp. 3-4] [17, p. 458] This approach requires the system to convey the depth buffer in addition to the color buffer. The depth buffer is used to determine the visibility of the final pixels. This requires more bandwidth between the processing units. [9, p. 3] [14, p. 4]

3.2 Front end

The application which contains the scene to be rendered resides in the front end of the distributed remote rendering pipeline [9, p. 2]. A relatively simple solution might be to send all the primitives to the rendering units. However, this increases the bandwidth usage, as the more complex the scene is, the more bandwidth is required. [9, p. 5]

Alternative approach to this task is to cache all the primitives of a respective processing unit in the rendering slave itself. Using this method, the front end requires only to send the primitives that are altered. This requires the system to transmit the whole scene data only once, after which the bandwidth requirements are substantially lower. [9, p. 5] However, a drawback of this solution is the increased complexity.

Front end using sort-first method of distribution may be sub-optimal, as changes in the camera position or rotation require the sending of the whole scene to the back end. However, in sort-last approach, the scene may be divided in a manner that is view-independent. This reduces amount of the sent data. [9, p. 6] As later discussed in this paper, view-independent calculations enable more efficient ways of parallerization, especially in scenarios where there are multiple clients in the same scene [8, p. 9].

3.3 Back end

The back end contains multiple processing units. They may be split into one master combiner and chosen number of slave renderers [9, p. 2]. Furthermore, before master combiner, there might be additional sub-combiners [9, p. 3]. The slave processing units render their assigned regions of the image and send the result to master combiner. The master combiner combines the results and produces a full image. [9, p. 2] To reduce bandwidth issues in the backend, sub-combiners may perform sub-image combinations before the master combiner [9, p. 3].

As discussed before, the sort-first approach requires less bandwidth but causes more overhead. Furthermore, as sort-first approach is not view-independent, which requires a substantial amount of data to be moved, if the camera view in the scene changes [3]. The sort-last approach reduces the overhead but requires the depth buffer to be transmitted between the processing units, which effectively doubles the bandwidth requirements as both color and depth buffer is required [9, p. 3]. Using cascading sub-combiners reduces bandwidth problems on the back end side, but increases latency [9, p. 4]. However, sort-last as is, prevents the processing of global illumination (GI) [3]. This paper discusses GI more in detail later on in the text.

4 Distributd rendering solutions

This section discusses different methods considering distributed and remote rendering. A number of relevant approaches and solutions are presented and their advantages and disadvantages are discussed.

4.1 Distributed global illumination using regular grid

Global illumination (GI) describes the movement of light in a scene, between different objects and surfaces. Magro et al. present a solution to remotely calculate global illumination using Regular Grid Global Illumination (ReGGi). This solution offers high-fidelity illumination to any device, regardless of the hardware [12, p. 10]. It may be utilized for example in wireless VR headsets, due to small hardware and bandwidth requirements [12, p. 11].

The design of ReGGi is focused on distributed environments, such as a rendering server. The server calculates the illumination data and transfers it to the clients, which are responsible for rasterization. The server divides the scene into a regular grid and every cell receives up to two light samples. The cell illumination is then calculated and sent to the client. The client stores a snapshot of the grid and updates it asynchronously when the individual cell data is received from the server [12, p. 13]. Using this method, the calculation heavy illumination processing occur on the more powerful server and the limited hardware of the client only has to rasterize the result.

4.2 Asynchronous distributed lighting calculation

The two approaches presented here are partially similar to before mentioned ReGGi, but they are discussed in this separate section, due to their similarity among themselves. Direct lighting is relatively light computationally and thus may occur in a client device [5, p. 1829] [8, p. 5]. However, indirect lighting requires substantially more computational resources. Remote Asynchronous Indirect Lighting (RAIL) is an approach to asynchronous distributed rendering. The main purpose of RAIL is to decouple the calculation of direct and indirect lighting in dynamic scenes. [5, p. 1829] Second approach to distributing lighting calculations is CloudLight framework, which also utilizes asynchronous lighting processing. It includes three different pipelines for different client-server scenarios. [8,

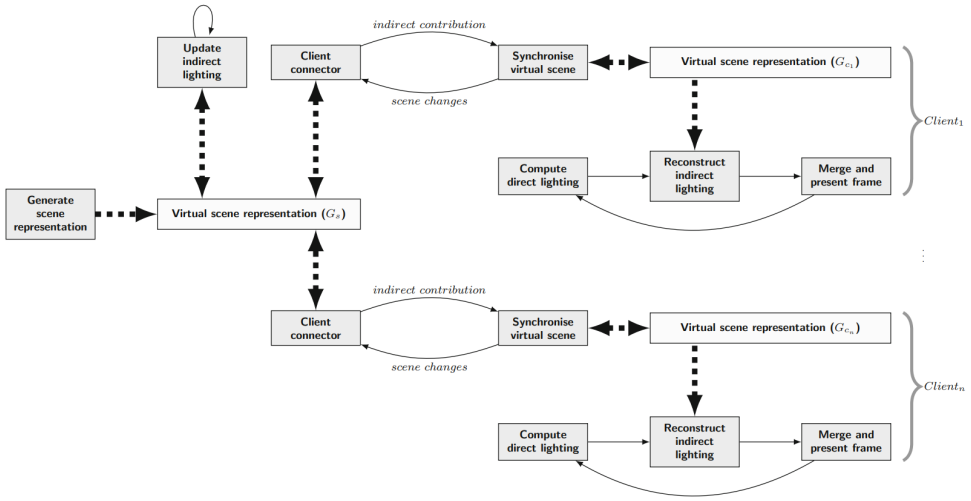


Figure 1. The architecture of the RAIL [5, p. 1830]

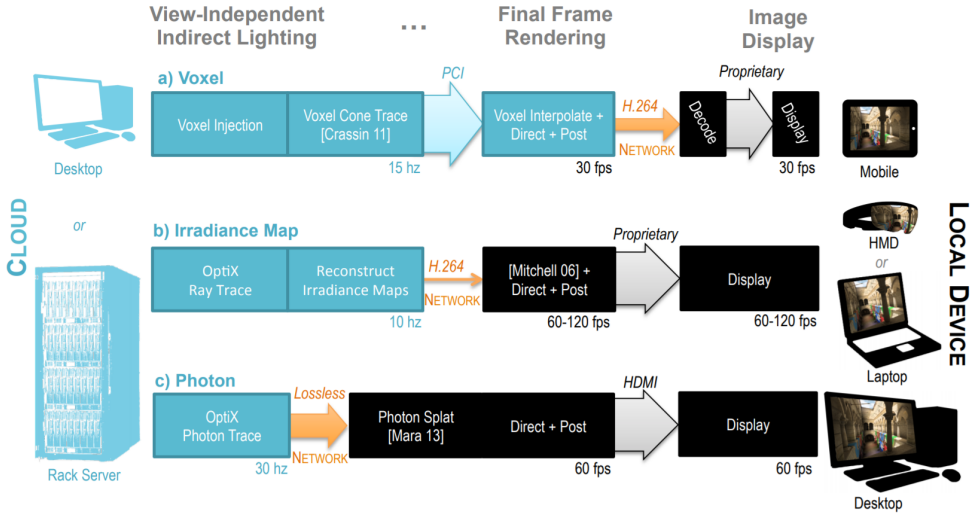


Figure 2. The three different CloudLight pipelines [8, p. 1]

p. 1]

In both CloudLight and RAIL, the server utilizes different tracing techniques, such as path, ray or cone tracing, to calculate the indirect lighting of the scene. [5, p. 1830] [8, p. 6] The results are then sent to the client, which has calculated the direct lighting. Client then proceeds to reconstruct the indirect lighting and merge it to the direct lighting. After that, the client may continue calculating the direct lighting. (Figure 1) However, in CloudLight approach, also the direct lighting may be calculated in server, when the clients are low-end devices, such as mobile phones. (Figure 2)

RAIL reduces latency, as the frame updates are not connected directly

to network performance. It also enables devices to reach higher fidelity visuals, as the graphics rendering is not entirely depended on the client hardware. Furthermore, computation costs may be reduced by sharing the view-independent indirect lighting information among other clients in the same scene. [5, p. 1829]

The testing with CloudLight revealed that latency in indirect illumination is in fact not a substantial problem. As the updating of lighting is performed asynchronously, it leaves the game play mostly unaffected. As long as half a second latency might be acceptable visually when rendering indirect lights [8, p. 2].

4.3 Megatexture processing

One bottleneck for graphics processing is the limitations of Video RAM (VRAM). Applications or games utilizing large textures are limited to the size of the VRAM of the graphics processing unit. Magro et al. approach this problem by circumventing the VRAM limitations by using Device-Agnostic Radiance Megatextures (DARM). Furthermore, this solution does not require high-end hardware on the client device, as a server performing the more calculation heavy operations. [13, p. 256]

Megatextures are large textures containing the whole scene geometry. DARM utilizes remote server for rendering and similarly to the before mentioned global illumination solution, the client rasterizes the result received from the server (Figure 3). The server utilizes two threads: a rendering thread and streaming thread. The updating of the megatexture occurs on the rendering thread and the streaming thread is responsible of conveying the data from the megatexture to the client. [13, p. 257] The server creates a megatexture out of the scene geometry, dividing the intdividual objects into non-overlapping patches that are set to the megatexture. The connection between patches are stored into connectivity graph. The server utilizes chosen rendering technique to render the megatexture. This technique may be for example rasterization, ray tracing or path tracing. Any calculation heavy shading, lighting information or other data are baked into the megatexture [13, p. 258] To preserve bandwidth, only modifications to the megatexture are sent to the client. The client renders the received megatexture on the scene mesh. [13, p. 259]

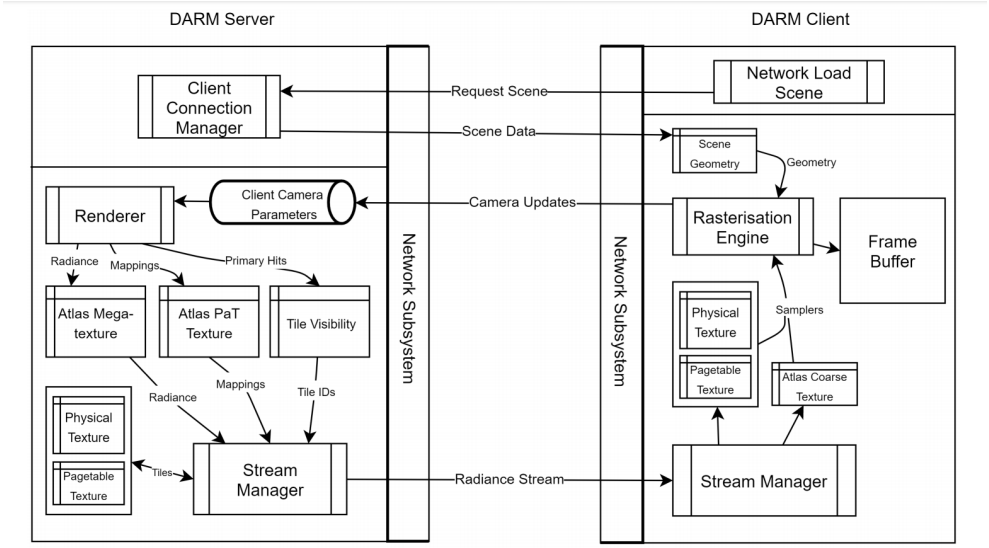


Figure 3. The architecture of the DARM server-client system [13, p. 257]

4.4 Distributed Framebuffer

Distributed Framebuffer (DFB) is a framework for applications using distributed rendering. In this framework, the image is divided into tiles and a dependency tree is established between them. If an operation for tile requires imforrmation from other tile, DFB uses the dependency tree to route the required information. [17, p. 460]

However, this implementation supports only local lighting effects. Therefore, global illumination is not a possibility for DFB as is. [17, p. 466] Yet, there exists approaches on sending global illumination rays between nodes, utilizing ray passing in sort-middle approach, as mentioned before [3]. Those implementations might offer ways of integrating global lighting effects into the DFB, but it remains unknown how efficient this implementation is [17, p. 466].

5 Case study: cloud gaming platforms

Cloud gaming is a relative new form of virtual entertainment, but it already has potential of gaining a substantial market share in the future [7, p. 1]. A number of services are already available for consumers. This section discusses two cloud gaming platforms: Google Stadia [1] and NVIDIA GeForce Now [2]. The bandwidth and latency compensation strategies of these platforms are presented and analyzed.

5.1 Google Stadia

One of the most recent major entries in the cloud gaming market is Stadia by Google Inc. It is a cloud gaming platform that provides both single time purchase games and subscription based services. [6, p. 1]

Google claims that it is using solutions named "negative latency" that reduce the latency over network. However, Google has not published any official details covering this statement, but there exists speculation over what methods they might have used. [6, p. 4] One possible method of reducing latency is future state prediction [6, p. 4] [11, p. 1]. The system uses Markov chains to predict user input and thus makes it faster to respond user input, if predicted correctly. In the case of contradiction between the user input and the prediction, the system may use error compensation or rollbacking. [11, pp. 2, 12] The Results implied increased user experience [11, pp. 12-13], thus Google might be implementing similar features to Stadia.

5.2 NVIDIA GeForce Now

NVIDIA is one of the largest graphics card manufacturers in the world. They have developed their own cloud gaming approach, GeForce Now (GFN) [16, p. 1]. GeForce Now implements bandwidth estimation and adaptation algorithms that compensate for the fluctuations in network connection. The adaptation algorithm prioritizes frame rate over resolution. Due to this, in the case of increased latency, GFN attempts to keep the frame rate (FPS) of the game stable by lowering the resolution. When recovering from latency increase, this same priority is observable: frame rate increases before resolution. [16, p. 3]

Additionally, GFN implements an error correction solution in the case of packet loss. Gameplay testing with atrificial packet loss displayed that even at packet loss rate of 10%, the gameplay was fluid. [16, p. 4]

6 Conclusion

This paper discussed certain approaches on distributed remote rendering. The general trend implies that there are three key factors: latency (and by extension client side frames per second (FPS)), bandwidth and graphics quality. In most of the cases, there is a trade-off between at least two

of these variables. An example of this was using sort-last approach, to decrease latency at the cost of bandwidth. Furthermore, certain methods described are applicable only to a certain feature of graphical processing, for example dividing image to tiles reduces the applicability of global illumination processing. Moreover, splitting graphics processing into task performed locally and externally, is a method that offered performance benefits. To give an example, increase in performance was discovered when using the division between direct and indirect light calculations for client and server side respectively.

As the case studies over cloud gaming platforms imply, there is also methods of compensating latency and bandwidth issues by other means than optimizing graphical calculations. User input prediction, network estimation and adaptation algorithms and error correction have also displayed value in achieving enhanced user experience in cloud gaming.

As the paper focused mainly on real-time rendering, a portion of potential material might have been left out of the scope of this paper. For example, some medical rendering or 3D-modeling methods might provide efficient approaches also to real-time rendering. This area of study was left out of the scope of this review, but further investigation might provide useful insight.

This paper covers only a portion of all the possible solutions in distributed rendering. However, even from this narrow literature review, potential solutions for seamless cloud rendering is visible. The techniques and methods described in this paper are proven to provide increase in performance individually. The next step is to explore methods of combining these different approaches, to achieve maximum efficiency for remote rendering.

7 Future sights

As mobile devices and by extension mobile applications continue to grow their hold on the computing hardware market, it is reasonable to believe that cloud computing in real-time rendering applications will also be used increasingly. As for today, gaming in cloud services is already a rising trend, but has not yet reached the point, where all of the users may experience the same quality than with own desktop PC or console.

As the paper by Crassin et al. suggests, a substantial amount of non-graphical applications today are using peer-to-peer techniques to send

data [8, p. 22]. Could it be plausible to implement a remote distributed peer-to-peer solution, that might not even require a hardware-heavy server, but rather utilizes the hardware of the users as a distributed rendering system?

References

[1] Google stadia. https://stadia.google.com/. Accessed 29.11.2020.

[2] Nvidia geforce now. https://www.nvidia.com/geforce-now/. Accessed 29.11.2020.

[3] G. Abram, P. Navrátil, P. Grossett, D. Rogers, and J. Ahrens. Galaxy: Asynchronous ray tracing for large high-fidelity visualization. https://player.vimeo.com/video/304815977, 2018. Accessed 28.11.2020.

[4] ATI. Ati crossfire™ technology white paper, January 2008.

[5] Keith Bugeja, Kurt Debattista, and Sandro Spina. An asynchronous method for cloud-based rendering. *The Visual Computer*, 35:1827–1840, December 2019.

[6] Marc Carrascosa and Boris Bellalta. Cloud-gaming:analysis of google stadia traffic, 2020.

[7] S. Choy, B. Wong, G. Simon, and C. Rosenberg. The brewing storm in cloud gaming: A measurement study on cloud to end-user latency. In *2012 11th Annual Workshop on Network and Systems Support for Games (NetGames)*, pages 1–6, 2012.

[8] Cyril Crassin, David Luebke, Michael Mara, Morgan McGuire, Brent Oster, Peter Shirley, Peter-Pike Sloa, and Chris Wyman. CloudLight: A System for Amortizing Indirect Lighting in Real-Time Rendering. *Journal of Computer Graphics Techniques*, 4:1–27, April 2015.

[9] André Hinkenjann, Fraunhofer Imk, Matthias Bues, and Tobias Olry. Mixed-Mode Parallel Real-Time Rendering on Commodity Hardware. In *In proceedings of 5th Symposium on Virtual Reality*, pages 1–11, 2002.

[10] Johnson, Philip B. (Campbell, CA, US). Connecting graphics adapters for scalable performance, January 2009.

[11] Kyungmin Lee, David Chu, Eduardo Cuervo, Johannes Kopf, Yury Degtyarev, Sergey Grizan, Alec Wolman, and Jason Flinn. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. MobiSys '15, page 151–165, New York, NY, USA, 2015. Association for Computing Machinery.

[12] Mark Magro, Keith Bugeja, Sandro Spina, and Kurt Debattista. Cloud-Based Dynamic GI for Shared VR Experiences. In *IEEE Computer Graphics and Applications*, pages 10–25, September 2020.

[13] Mark Magro, Keith Bugeja, Sandro Spina, Kevin Napoli, and Adrian De Barro. Atlas shrugged: Device-agnostic radiance megatextures. In *VISIGRAPP 2020 - Proceedings of the 15th International Joint Conference on*

Computer Vision, Imaging and Computer Graphics Theory and Applications, volume 1, pages 255–262, February 2020.

[14] Steven Molnar, Michael Cox, David Ellsworth, and Henry Fuchs. A Sorting Classification of Parallel Rendering. In *IEEE Computer Graphics and Applications*, volume 14, pages 23–32, July 1994.

[15] Shi Shu and Hsu Cheng-Hsin. A Survey of Interactive Remote Rendering Systems. In *ACM Comput. Surv*, volume 47, pages 1–29, May 2015.

[16] M. Suznjevic, I. Slivar, and L. Skorin-Kapov. Analysis and qoe evaluation of cloud gaming service adaptation under different network conditions: The case of nvidia geforce now. In *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, 2016.

[17] Will Usher, Ingo Wald, Jefferson Amstutz, Johannes Günther, Carson Brownlee, and Valerio Pascucci. Scalable Ray Tracing Using the Distributed FrameBuffer. *Computer Graphics Forum*, 38:455–466, June 2019.

[18] Evangelos Zotos and Rainer Herpers. Interactive Distributed Rendering of 3D Scenes on multiple Xbox 360 Systems and Personal Computers. In *International Conference on Cyberworlds*, pages 114–121, 2012.

Usability and Security Tradeoffs in QR Code Usage

Vili Moisio
vili.moisio@aalto.fi

Tutor: Amel Bourdoucen

Abstract

Quick Response (QR) codes are two-dimensional bar codes that can be read and generated with ease using smartphones. For this reason, they have entered the public consciousness and are used in various contexts in our everyday life. Despite their popularity and long history, they are known to be vulnerable to security threats due to a lack of standardized countermeasures. These concerns have generally been accepted in order to keep the user experience simple and flexible. This paper is a literature review of recently published proposals and experimental solutions to further augment the technology. The central criteria with regards to the user experience of QR codes are laid out and inspected in detail followed by the biggest security challenges. Subsequently, the featured schemes are evaluated by their capability to make the use of QR codes safer for the user while maintaining a desirable extent of usability.

KEYWORDS: usability, security, QR code, bar code, mobile

1 Introduction

As the world becomes increasingly more digital and wireless, the need for fast and seamless transmission of information increases. In the ideal case, the process should only require hardware that is readily available and simple to operate while only demanding minimal input from the user. One such method is the Quick Response code (QR code) that was initially conceived by and for the Japanese automotive industry in 1994 [2].

The QR code [21] uses square grids with two-dimensional bar codes consisting of black squares on a white background to convey the data in a form that is readable by many devices with visual scanning capabilities, including modern smartphones that are typically equipped with cameras. In a typical case, the code contains an encoding of text data or a URL for quick access via a web browser.

The use of QR codes is cost-effective in addition to being easy both to generate and distribute, which has made them common in contexts such as marketing and making purchases via mobile devices. Despite their numerous advantages for everyday application, their utilization comes with its own set of weaknesses. In particular, the risks regarding security are often overlooked in the context of QR codes [16]. This paper examines the potentially susceptible aspects of the technology and to what extent can they be remedied without significantly dampening the user experience.

The structure of this paper is outlined in the following manner. Section 2 establishes the underlying technical specifications and principles of QR codes. Section 3 inspects the usability criteria in the context of typical present-day use cases, whereas Section 4 presents the security problems and methods to thwart them in contrast. These two aspects are combined in Section 5 in the form of tradeoff analysis and proposed solutions. Section 6 concludes the topic with finishing statements.

2 Technology

The QR code [1] can be generated according to several configurations with a different total number of filled and empty square-shaped cells, which are also referred to as modules. These configurations range from 21x21 modules (Version 1) to 177x177 modules (Version 40). Furthermore, the more compact Micro QR code can comprise from 11x11 squares up to 17x17 in increments of two per dimension for a total of 4 versions (M1 – M4).

Figure 1. Composition of a QR code [13]

In addition to the size of the matrix, another contributing factor to the amount of data that can be carried by one code is the encoding used. QR codes support four different character encoding sets: numeric, alphanumeric, byte and kanji (i.e. characters of Chinese origin in the Japanese language) data [1].

As specified in detail in [21], a Reed-Solomon error correction algorithm with four levels of varying intensity is utilized in QR codes to recover damaged or unrecognized data. The maximum capacity of the data carried by the matrix becomes smaller for each increase in the level of correction. The second to lowest degree is the most commonly used form of correction, but it may be profitable to sacrifice character count in favor of more reliability in more harsh conditions.

2.1 Encoding

The aforementioned components make up for the majority of any given QR code. In accordance with the encoding chosen, the central data area of the square contains the data as white and black symbols indicating the binary symbols 0 and 1 respectively. These cells and the characteristics mentioned in the following paragraph are pinpointed in Figure 1.

For the purpose of locating and interpreting the QR code in a stable manner, the square contains standardized patterns at specific points of the matrix [5, 16]. Finder patterns at three corners of a matrix allow its size, position and orientation to be deduced correctly. Larger versions of QR codes include lone isolated modules as alignment patterns to combat warping of the image and timing patterns to pinpoint coordinates of each module. Timing patterns are characterized as single horizontal and

vertical lines with alternating modules of black and white that connect the finder patterns to each other. To distinguish the QR code from background noise, the matrix is enclosed in a white margin on all sides. This part is called the quiet zone and is often equal in depth to the size of four modules.

2.2 Decoding

After being read by a camera, a common method of deciphering the image used by decoders is to calculate the luminance values of each module [9]. The binarization process uses a non-standard threshold value to differentiate between black and white modules, as there will always be situational variables based on the specific circumstances and the hardware used, often providing vastly different results. Furthermore, the calculation can be done on a global scale including the entire graph or with local windows, the latter of which has generally been proven to be more reliable. A popular open source library called ZXing uses a hybrid technique in which a set of windows are overlapped in order to measure the average luminance.

3 Usability

Given the growth in the diversity of applications reliant on QR codes intended for various industrial and leisure purposes alike, the technology must accommodate for many different situations while keeping the user experience at an acceptable level [12]. The main problem that arises may change depending on the task at hand – as such, the shortcomings must be examined at a case-by-case base. In any event, the speed at which the code is recognized and decoded in tandem with the overall success rate are generally considered [8, 12, 21] adequate metrics for measuring usability.

This section presents common challenges encountered when scanning QR codes and what should be taken into account when generating them in some of the most prominent areas where they are utilized.

3.1 Distance

As supported by the experiment in [8], the proximity of the reader to the QR code that is being read proves to have a significant effect on the success rate of a scan attempt. The effect is further compounded when the data density of the given matrix increases, with the discrepancy being

especially large between matrices of different sizes that contain smaller amounts of information. A logical counter-measure to deal with situations where a greater distance must be achieved is to enlarge the physical dimensions of the QR code upon deployment.

3.2 Geometric distortion

Minimizing the distance to a QR code brings up another issue in that even small lateral movements cause the angle to deviate further from perpendicular. While the existence of alignment patterns mitigates this to an extent [21], the difficulties that arise from such distortions are not limited to the scanning direction being suboptimal. In fact, there are times in which it is desirable to display the QR code on a curved or otherwise uneven surface. For achieving such purposes, approaches such as the one employing a form of shadow manipulation for matrices on non-planar surfaces in [20] have been studied.

3.3 Particularity

Some use cases, such as the one described in [17] may require several QR codes to be placed near one another. In such instances the adjacent matrices may interfere with each other, as typical scanners are not equipped with the capabilities to pick and choose at one's discretion but rather decode the first one that is recognized as valid. A digital zoom feature in some smartphone applications can be used to tackle this problem, but a marker that singles out the specific QR code is ideal.

3.4 Identifiability

One apparent drawback of QR codes is that their contents are unintelligible to the human eye. As a monochromatic bar code, the characteristic look lacks the attractive and captivating qualities that are deemed important in public relations and commercial contexts. Research has been conducted [9, 18] for the purpose of incorporating visually consequential colored imagery into a standard QR code to make them more appealing and distinguishable. However, it is crucial to understand that this is merely a decoration and does not guarantee the actual contents of any given matrix.

4 Security

The QR code presents appealing propositions due to the high availability of readers and fast, simple operation. On the other hand, the ease of use combined with the lack of a standardized security or privacy measures makes it a potential target for malicious attacks [8]. Security becomes an increasingly important aspect for use cases where confidentiality is assumed, such as when the identity of a person has to be asserted or when money is transferred – both of which QR codes have been used for [16].

Some of the most prominent security loopholes and forms of misuse of QR codes are described below followed by concepts that could help minimize their impact.

4.1 Tampering

An existing QR code can be turned into an attack vector by superimposing a set of black squares on top of it in such a way that the encoded data changes [23]. If the bad actor is aware of the the inner workings of the technology, virtually any white-to-black modification can be carried out using something as simple as a pen. The attacker may also replace the entire QR code to achieve similar results in cases where it is printed on a loose sheet [16].

This method has been used for example to alter an URL in order to lead the user to a website that differs from what the destination was assumed to be [15]. The site may masquerade as a trusted entity and try to get the user to disclose some confidential information for example by prompting for credentials as a form of phishing attack. Likewise, QR codes attached to advertisements that are meant to offer goods or services can be tampered with to display website similar to the one expected, but having the payment directed to the carrier of an attack, thus leading to a fraudulent sale and monetary loss.

4.2 Malicious QR codes

Rather than modifying a point of interest, the fast and cheap generation and proliferation of QR codes also allows attackers to tailor their own matrix from scratch, therefore enabling for more bespoke data content. Spreading unassuming QR codes on the internet or by other means even

without explicit context can prove surprisingly effective, as the mere curiosity of people can turn them into potential victims [6]. These types of attacks have the possibility of including malware that can exploit vulnerabilities in a given reader software to hijack the device and take use of its various functions [15].

4.3 Combative measures

To prevent unsuspecting users from visiting malicious websites via QR codes, the most self-explanatory method is to make the target URL known before navigation can occur [14]. However, that alone can not be deemed a sufficient solution as it assumes that the person scanning is capable of recognizing harmful links on their own. The prominence of URL shortening services and domain names that closely resemble legitimate ones by including subdomain prefixes or special characters make this type of attack even more effective. Thus, preventive measures that blacklist or detect dubious sites could be incorporated to the browser or reader software themselves.

The paper in [3] proposes a public key authentication method using the secure RSA algorithm as a form of encryption for QR codes. Therefore, following the specification each matrix is signed and encrypted before generation and can only be read by a user possessing the matching private key. In addition to the validation feature, documents shared in the schema can be verified using a digital signature. The system is intended to allow the sharing of sensitive data while retaining the convenience of the ubiquitous nature of QR code scanners.

A novel approach to the encrypting of QR codes by camouflaging the matrix called mQRCode is presented in [19]. The technique is based on the distractions created by the Moiré phenomenon and as such the matrix is concealed due to the innate properties of the display projecting the bar code and the camera lens rather than encrypted messages or passwords. During the scanning process, the QR code is introduced to the scanner at an exceedingly limited distance and viewing angle in order to expose the data. The intended use for this kind of security measure that relies on optical cryptography is mainly to prevent an outside force of interrupting a mobile payment QR code, which are characteristically only visible for a short period of time before it is scanned by the intended recipient.

5 Evaluation of usability–security tradeoffs

The concepts of usability and security maintain a close relationship that, depending on the circumstances, can often be considered to be a conflict of interest to a varying degree: any modifications to improve one of these two aspects in a given system will in most cases cause the other to suffer. As the user experience is significantly more visible and widely understood by the general population, it is only natural for security concerns to attain a much smaller degree of importance – at least until the protection is compromised and the damage is already done. QR codes exemplify this phenomenon quite well, as despite their relatively long existence and the lack of security being acknowledged by professionals, the process of improving security for mobile technologies in general has experienced a considerable level of friction [14].

This section analyzes possible solutions to improve the lacking aspects of the QR code standard. More specifically, we are interested in the large-scale implications with both security and usability viewpoints in mind.

5.1 Encryption

Incorporating additional security features such as a encryption would go a long way towards making the application of QR codes safer. To this end, cryptography systems based on symmetric [10] and private-public key pairs [3] have been proposed. The primary aim of both solutions is to provide a way to share sensitive information without the possibility of outsider interference, which would expand the pool possible use cases. In addition, the symmetric key method doubles as a means to prevent modified, malicious QR codes from being read, as the encryption prevents tampering by anyone who does not know the password.

There are numerous impediments to integrating such cryptographic methods to the current, widespread implementation of QR codes. Generating encryption keys can be computationally expensive and unavoidably introduces some processing time to each bar code creation and scan attempt. This drawback is aggravated even more by the need to prompt users for password input, which complicates the experience in addition to causing the process to take longer.

We also note that encryption inevitably creates overhead that further curtails the already limited data cap of QR codes. If the proposed method were to be used for transferring large records of identifiable information,

it could well require multiple matrices to be generated and consequently read with each event enquiring for credentials separately. On the other hand, a comprehensive survey conducted on cryptographic QR codes describes their evaluation of the user experience as "reasonable" [8] while admitting that situations where many of the aforementioned obstacles do become a significant problem.

5.2 Trusted third parties

The use of QR codes could also be made more secure via the help of an external authentication layer [3]. An authority that can certificate a QR code could turn their use into a legitimate method of identification, as currently there is no guarantee on the validity of the decoded contents. However, the current implementation has proven adequate for closed environments such as hospitals where the risk of outsider interference is small when evaluated against the benefits of the fast operation, but fails to provide enough protection on a larger scale and in the context of public spaces. As an applicable real-world example of a highly critical use case in the form of an authenticated online voting system is presented in [7].

Nevertheless, the validity of such schemes has also been contested by highlighting the impracticality of having to rely on signatures and trusted third parties [17, 23]. In the current times, internet infrastructure and mobile end devices are arguably even more omnipresent than the bar code scanners themselves, but necessitating online confirmation creates additional costs and another potential hurdle to be cleared in the case of a downtime or other disruption in service. Furthermore, if a specific application is required to operate such an interface, the entire process is likely to be cumbersome enough that the immediate and distinguished benefits of QR codes are essentially nullified.

5.3 Obfuscation

A more human-centric approach to achieving an increase in security that relies on the perception and habits of the users is to conceal QR codes in various ways. If the aim of the attacker is to make minor adjustments to the contents of a bar code and lead victims to a falsified destination, injecting QR codes on surfaces other than flat, two-dimensional print media and screens [20] turns tampering into a far more complicated task. Alternatively, the matrix can be disguised so that it cannot be observed by the

naked eye and requires a specific procedure to be scanned [19]. The previously introduced e-voting scheme [7] uses an additional layer of visual cryptography by means of splitting the encoded data into two components, both of which are required to obtain the resultant QR code.

While the aforementioned methods provide interesting results with potential for certain situations, we maintain that their large-scale applications are limited. The traditional QR code specification has been criticized for its lack of immediate visual feedback [9] and any additional masking is sure to cause confusion and ire to the inexperienced user. Moreover, using camouflaging in this manner can be though of as security by obscurity and thus cannot be relied upon as the lone form of protection.

5.4 Regulation

When visiting an open mobile marketplace such as the Google Play digital store, a quick search in the application section reveals numerous third-party alternatives for a QR code reader. There are no immediately perceptible differences between any of them and we can observe that the vast majority of the applications have received positive ratings. Be that as it may, a thorough survey [22] suggests that a significant portion of available software capable of parsing QR codes does not employ security measures to a sufficient degree. Most notably, there is a large discrepancy in how a decoded URL is displayed, if at all, and the permissions requested do not always correspond to any specific actions of the software.

A quick test on Android 9 and 10 reveals that the default camera software recognizes and scans QR codes, although the older version requires Google Lens to be active. This feature is not advertised in any meaningful way, so an user who is not aware of its inclusion might resort to downloading a dedicated application and subject themselves to potential security risk. The same problem technically exists on iOS but is mitigated considerably, as the review process [4] is much more thorough on the platform. However, as freedom is a key asset of the Android ecosystem [11] and the core user base is built upon this very foundation, there can never be absolute guarantee of all available applications being up to the standards despite their best efforts. Ultimately, the burden then falls on the platform holder to raise awareness of the features it provides and additionally to implement system-level security measures that utilize cutting edge technology such as machine learning.

6 Conclusion

In this paper, we inspect potential means of improving the technology of QR codes. The main focus is placed on the aspects of usability and security. In particular, the point of interest is how changes in one of the key areas affects the other and whether or not a well-balanced solution can be proposed at this time. Innovations that would enable the technology to be applied in new ways were also discussed.

We find that despite several studies having touched upon the subject, a sufficiently versatile solution is proving hard to develop. The topic of QR codes is still ongoing research, but the extent and long-term impacts of any experimentation conducted have been minimal. The viability of making significant modifications to an already established standard is a challenging endeavor on the face of it, and even more so for a technology that is as known and widely used by the general populace as QR codes are. It seems rather likely that QR codes continue to fill their purpose as they currently do and a competing technology fills the needs that remain unaccounted for.

References

[1] Information Technology – Automatic Identification and Data Capture Techniques – QR Code Bar Code Symbology Specification. Standard ISO/IEC 18004:2015(E), March 2015.

[2] Denso ADC. QR Code Essentials, 2012. Accessed on: November 27, 2020. [Online]. Available: https://delivr.com/resources/files/1058/DENSO_ADC_QR_Code_White_Paper.pdf.

[3] S. Ahamed and H. A. Mustafa. A Secure QR Code System for Sharing Personal Confidential Information. In *International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering*, July 2019.

[4] Apple. App Review. Accessed on: November 27, 2020. [Online]. Available: https://developer.apple.com/app-store/review.

[5] E. Chen and L. Lei. Research and Realization of Encoding on QR Code. In *2012 5th International Congress on Image and Signal Processing*, pages 1299–1302, October 2012.

[6] R. Dudheria. Evaluating Features and Effectiveness of Secure QR Code Scanners. In *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pages 40–49. October 2017.

[7] S. Falkner, P. Kieseberg, D. E. Simos, C. Traxler, and E. Weippl. E-voting Authentication with QR-codes. In *Human Aspects of Information Security, Privacy, and Trust*, volume 24, pages 149–159, June 2014.

[8] R. Focardi, F. L. Luccio, and H. A. M. Wahsheh. Usable Cryptographic QR Codes. In *2018 IEEE International Conference on Industrial Technology*, pages 1664–1669, February 2018.

[9] G. J. Garateguy, G. R. Arce, D. L. Lau, and O. P. Villarreal. QR Images: Optimized Image Embedding in QR Codes. *IEEE Transactions on Image Processing*, 23(7):2842–2853, July 2014.

[10] N. Goel, A. Sharma, and S. Goswami. A Way to Secure a QR Code: SQR. In *International Conference on Computing, Communication and Automation*, May 2017.

[11] Google. Secure an Android Device. Accessed on: November 27, 2020. [Online]. Available: https://source.android.com/security.

[12] M. Katona, P. Bodnár, and L. G. Nyúl. Distance Transform and Template Matching Based Methods for Localization of Barcodes and QR Codes. volume 17, number 1, pages 161–179, January 2020.

[13] KeepAutomation. QR Code Introduction. Accessed on: November 20, 2020. [Online image]. Available: http://www.keepautomation.com/qrcode/.

[14] A. Kharraz, E. Kirda, W. Robertson, D. Balzarotti, and A. Francillon. Optical Delusions: A Study of Malicious QR Codes in the Wild. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 192–203, September 2014.

[15] P. Kieseberg et al. Malicious Pixels Using QR Codes as Attack Vector. In Ismael Khalil and Teddy Mantoro, editors, *Trustworthy Ubiquitous Computing*, chapter 2, pages 21–38. Atlantis Press, Paris, 2012.

[16] K. Krombholz, P. Fruehwirt, P. Kieseberg, I. Kapsalis, M. Huber, and E. Weippl. QR Code Security: A Survey of Attacks and Challenges for Usable Security. In *Human Aspects of Information Security, Privacy, and Trust*, volume 24, pages 79–90, June 2014.

[17] S. Latvala, M. Sethi, and T. Aura. Evaluation of Out–of–Band Channels for IoT Security. *SN Computer Science*, 1(1):18:1–18:17, January 2020.

[18] K. T. Lay and T. C. Chen. Visual QR Codes with Lossless Picture Embedding. In *2018 3rd International Conference on Intelligent Green Building and Smart Grid*, pages 50–53, April 2018.

[19] H. Pan, Y. C. Chen, L. Yang, G. Xue, C. W. You, and X. Ji. mQRCode: Secure QR Code Using Nonlinearity of Spatial Frequency in Light. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 27:1–27:17, October 2019.

[20] H. Peng, L. Lua, A. Sharf, and B. Chen. Fabricating QR Codes on 3D Objects Using Self-Shadows. *Computer-Aided Design*, 114:91–100, September 2019.

[21] S. Tiwari. An Introduction to QR Code Technology. In *2016 International Conference on Information Technology*, pages 39–44, December 2016.

[22] H. A. M. Wahsheh and F. L. Luccio. Security and Privacy of QR Code Applications: A Comprehensive Study, General Guidelines and Solutions. *Information*, 11(4):217:1–217:23, April 2020.

[23] K. S. C. Young, K. L. Chiew, and C. Lin Tan. A Survey of the QR Code Phishing: The Currentattacks and Countermeasure. In *2019 7th International Conference on Smart Computing & Communications*, June 2019.