Individual Diploma Thesis

MACHINE LEARNING IN 5G SECURITY

Sofroniou AnnaMaria

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

May 2023

UNIVERSITY OF CYPRUS
DEPARTMENT OF COMPUTER SCIENCE

MACHINE LEARNING IN 5G SECURITY

Sofroniou AnnaMaria

Supervisor
Dr. Vasos Vassiliou

The Individual Diploma Thesis was submitted for partial fulfilment of the requirements for obtaining the degree of Computer Science of the Department of Computer Science of the University of Cyprus

May 2023

Acknowledgements

I would like to sincerely thank my supervisor Dr. Vasos Vassiliou, who offered me the opportunity to work on such an interesting topic for my thesis, and for the guidance he provided.

Also, I would like to thank my family and friends for their support and encouragement during my studies.

Abstract

This individual thesis project contains information about machine learning techniques in 5G security. The fifth generation of cellular networks is the most advanced cellular technology deployed so far. 5G brings innovations in comparison to the previous generations like massive MIMO, the mmWave, Multi-access Edge Computing and Network slicing, while providing high data rates, low latency, and high bandwidth. The security of the fifth generation has been evaluated by many researchers. While its security has improved over the other previous generations, still has a few vulnerabilities that have been brought to broad attention. These vulnerabilities are presented in the thesis.

Machine learning is considered a very powerful to fortify the security of 5G. In this thesis are presented some machine learning approaches, for example, federated learning for HetNets, and for network slicing. Another machine learning solution presented is about reinforced learning for multi-access edge computing.

Furthermore, machine learning is a double-edged sword for security, since on the one hand it enhances it, but on the other side, machine learning models introduce their own vulnerabilities in the system. Vulnerabilities of machine learning models are included in this thesis too.

Finally, conclusions about the security evaluation of the implication of machine learning in 5G are presented, as well as future works about machine learning models in wireless communications.

Contents

Chapter 1

Background information about cellular networks

1.1 General introduction

The innovative 5G technology took the world by storm this decade. The rapid spread of the fifth generation can be justified by the high bandwidth, high data rates and low latency it delivers. The new promising technologies introduced revolutionized the cellular networks technologies and paved the way for the upcoming generations. Massive MIMO and the mmWave enhanced the performance of the network by efficiently using resources, multi-access edge computing reduced latency, computational and energy requirements for end devices, while network slicing provided customization and isolation for network slices.

The security of a such widely used cellular network was evaluated, and as expected there are some vulnerabilities presented just like in every other system. The vulnerabilities violate the CIA security triad, so researchers proposed multiple solutions to prevent that. Multiple solutions proposed use machine learning approaches to mitigate the security challenges introduced.

Machine learning is commonly used for security purposes, but unfortunately, machine learning can also have vulnerabilities, so the networks implementing such solutions

inherit these vulnerabilities too. An adversary can exploit these vulnerabilities to attack the network, compromising its privacy and confidentiality.

1.2 Motive

There are several motives for conducting a literature review on machine learning in 5G security. To begin with, the fifth generation is already deployed and available to the public, meaning its threat surface should be addressed, with the motive being to stay on top of the latest advancements that can tackle these challenges. One of the emerging techniques to tackle security concerns are machine learning models, so another motive is to what machine learning models work first of all in the context of not only 5G networks, but also to enhance its security. Machine learning approaches were reviewed and presented in the literature review to highlight models that are applicable and can be integrated for security enhancement of different 5G technologies. Reviewing how machine learning can be used for securing 5G is a motive, but on the other hand, reviewing how machine learning can be used against 5G security is another motive, since even machine learning models can be vulnerable to attacks. These vulnerabilities are also presented in the literature review. Finally, motive for this literature review is also to identify research gaps for future works.

1.3 Methodology

For thesis follows a literature review methodology, meaning that there was a comprehensive study of existing relevant literature, and provided an overview of the information gathered. There are many research papers about 5G security that needed to be filtered based on their relevance on the research topic. The research of information started from a wider topic being the fifth generation of cellular networks, its innovations, architecture, and technologies, to understand the environment working on. Then, the research narrowed down to the threat landscape of 5G, in order to collect information about the vulnerabilities of this generation, how they can be exploited and what an attacker can gain from exploiting them, and a summary of the findings is provided in chapter 4. Finally, the research concluded on the topic of machine learning based security solutions for 5G, where various solutions for the previously studied vulnerabilities were

analyzed and an overview of these solutions is presented in chapter 5. The aim of this literature review was to present relevant research papers of the community that provide smart solutions to security problems of our current cellular network.

1.4 Outline

The thesis is structured as follows. In chapter 2 background information about the fifth generation of cellular networks is provided, including its history and features. Next, in chapter 3 is presented background information about the machine learning models that mentioned in chapter 5. The thread vectors of 5G are presented in chapter 4, while in chapter 5 machine learning approaches by various researchers are presented to mitigate the challenges previously mentioned. In chapter 6 the vulnerabilities of machine learning are mentioned. Finally, in chapter 7 conclusions and future work are presented.

Chapter 2

Background information about cellular networks

2.1 History of cellular networks

The history of cellular network communication dates to the early 1980s when the first generation of mobile networks (1G) was introduced [1]. The first generation only transmitted voice signals over a frequency modulated circuit switched network, with a maximum speed of only 2.4 kbps. In addition, since the first generation used to transmit analog signals, they were able to communicate over long distances, but the said analog signal would be degraded by factors like interference and noise. The debut of the first generation marked the beginning of the development of more advanced cellular networks that lead to today's cellular communication networks.

Continuing, the second generation of cellular networks was launched in the early 1990s bringing a significant innovation over the first generation. From the second generation and beyond, the cellular networks switched from analog and frequency modulated schemes, to digital signalling and its according modulation schemes like Time Division

Multiple Access (TDMA) and Code Division Multiple Access (CDMA), increasing the capacity of the network and providing a more efficient use of the available spectrum [2]. Along with those innovating changes, the second generation introduced the Global System for Mobile Communications (GSM) standard, that included new features like Short Message Service (SMS), Subscriber Identity Module (SIM) cards, and international roaming. The second generation, even though its technological limitations, it paved the way for more advanced generations in the future.

Following, the third generation of mobile network communication (3G) was introduced in the early 2000s. The third generation was a major leap from the previous generations since it switched from circuit to packet switching, increasing the peak data rate from 384kbps to 2Mbps [3]. Since 3G is the first generation that is fully IP-based system, it brought advanced IP-based services, such as web browsing, Multimedia messaging service (MMS), streaming and location-based services. Overall, the third generation had significant advancements, and can be considered the foundation for the next generations of cellular networks.

Moving onto, the fourth generation of cellular networks was developed in the late 2000s and deployed in the early 2010s. The fourth generation has all-IP architecture, making the IP-based system that introduced in the third generation more optimal. The all-IP architecture in combination with the use of unified-IP for all data traffic, enabled high speed communications by providing a higher maximum data rate of 1Gbps, lowered the latency of the network, and enhanced the flexibility of the mobile network. Finally, even though the fourth led to the latest cellular generation deployed, it is still extensively used today.

2.2 Fifth Generation of Cellular Networks (5G)

The most advanced cellular network to date is the fifth-generation mobile commination system. The fifth generation was introduced in the early 2010s, promising high data rates, low latency, and high bandwidth. The introduction of massive Multiple-Input Multiple-Output (MIMO) in 5G increased the capacity allowing massive connectivity, which is a feature needed due to the increasing number of connected devices. Massive MIMO also

provided improved coverage and spectral efficiency in comparison to previous generations. This generation of cellular networks was the first to use the Millimetre-wave (mmWave) frequencies that provide wider bandwidth, hence faster data transfers rates. In addition, innovative technologies were introduced such as Multi-access Edge Computing (MEC), that helped reduce latency and preserve privacy, and network slicing that improved security. Overall, the fifth generation is a major leap forward for cellular communications, justifying its rapid spread with its promise of low latency, high speed, and new innovations.

	1G	2G	3G	4G	5G
Introduced	Early 1980s	Early 1990s	Early 2000s	Late 2000s	Early 2010s
Year Deployed	1979	1991	2001	2009	2018
Data Rate	2,4 Kbps	64 Kbps	2 Mbps	1 Gbps	20 Gbps
Features	Voice only, Analog, FDMA	Digital, TDMA & CDMA, International Roaming, SMS, SIM	IP-based system, Web browsing, MMS, location-based services	All-IP architecture	Massive MIMO, mmWave, MEC, Network slicing

2.3 Architecture

5G uses a Service-Based Architecture (SBA), which means that the communication in between network functions is done by a service-based interface [4]. The SBA provides flexibility, scalability, and heterogeneity to the network since it enables various services and devices.

The 5g system can be divided into three components, the User Equipment (UE), the Radio Access Network (NG-RAN) and the Core Network (5GC) [5]. The UE is a device of a

user that is connected to the network, the NG-RAN is the radio transmitter that receives and transmits signals, and the 5GC that is responsible for the network functions, such as the Authentication Server Function (AUSF) which is responsible for authentication and key management, the Mobility Management Function (AMF) that is responsible for authentication and mobility of users, etc.

2.4 Technologies of Fifth Generation

2.4.1 Massive Multiple-Input-Massive-Output

Massive Multiple-Input-Massive-Output (massive MIMO) is an enhanced MIMO system with a large quantity of antennas that was introduced in 5G [6]. This technology uses antennas that send and receive multiple signals simultaneously, and therefore can communicate with many users simultaneously, enhancing the capacity of the network.

Also, Massive MIMO, due to the high number of antennas, enables 3D Beamforming [7], which is a technique for controlling the signal's direction. In a massive MIMO system, a base station can distinguish between users with the help of angle of arrival (AoA) and angle of direction (AoD) of a user's signal. This information helps beamforming to control the direction of a signal, in combination with adjusting and combining signals from using multiple antennas to add up constructively towards the node desired and destructive in the other directions, reducing interference and even jamming, and as a result increasing SINR. This technique increases signalling performance for devices indoors or in dense environments with multiple devices like in 5G.

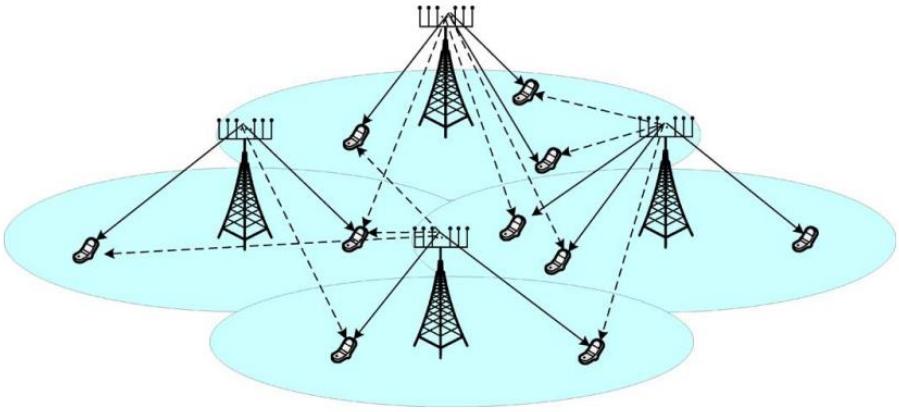

Figure 2.4.1 - 1 Massive MIMO system [8]

Concluding, Massive MIMO is an impressive innovation in 5G that with the usage of multiple antennas and beamforming increases reliability and data rates by providing more signal routes.

2.4.2 Millimetre Wave

The millimetre wave (mmWave) is the frequency range from 24GHz to 100GHz introduced in the fifth generation of cellular networks [9]. As the frequency increases, the wavelength decreases, and since the mmWave is classified the Extremely High Frequency (EHF) rage, its wavelength is one millimetre short. Since the mmWave works at higher frequencies, it provides bigger bandwidth, and therefore higher data rates to the network. The theoretical data rate referred can reach up to 4Gbps, and that enables services like ultra-high-definition video (UHDV) in 5G.

2.4.3 Network Slicing

The fifth generation is a heterogenous cellular network, with various devices, and therefore many differentiated services. All these differentiated services have different requirements of functionality and performance. To accommodate these requirements, network slicing was introduced. Network slicing is a network architecture that divides the network into many virtual networks over a shared infrastructure [10]. With the help of cloud computing and virtualization [11], the network can be divided into logical

networks, that even though they share resources, they are isolated in between them to enhance their security. In this way, the slices can have not only optimized requirements based on their custom needs, but also the slices can adapt to changing requirements.

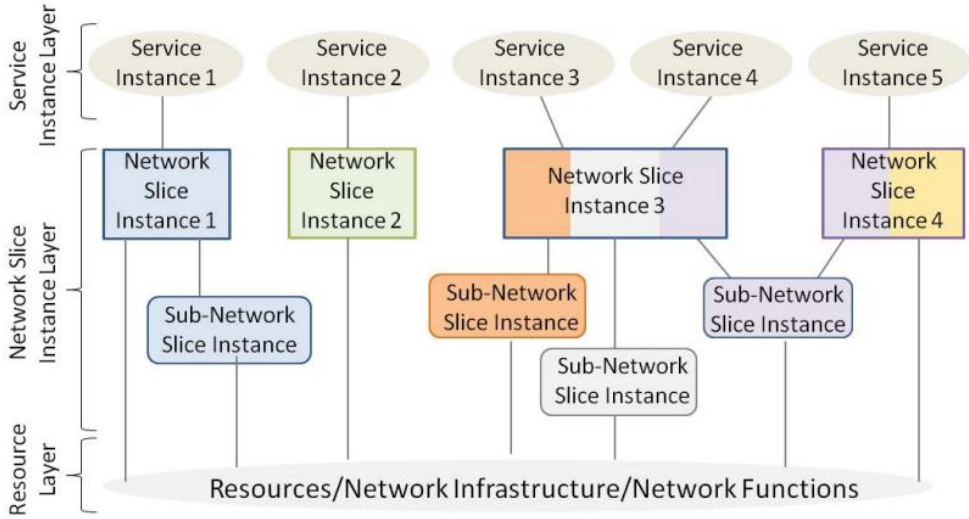

Figure 2.4.3 - 1 Example of Network Slicing Technology [12]

2.4.4 Multi-access Edge Computing

Multi-access Edge Computing (MEC) is a network architecture introduced in 5G, that brings computational and storing services closer to the end-users of the network [13]. One of the services included is computational offloading, where a mobile node offloads a computational complex task to the edge if the end node is not powerful enough to compute it on its own or to save energy. MEC also enables distributed caching, where an end node can fetch from the edge cached information, reducing the traffic from the backhaul to the core network. All these services reduce latency since the mobile nodes only need to communicate with the edge nodes which are in proximity. Concluding, services at edge provided by MEC, in a dense or data centric network, enable mobile nodes to access computational and storage resources faster, enhancing network performance, by reducing latency due to the proximity and increases data rates due to reduced traffic to the core network.

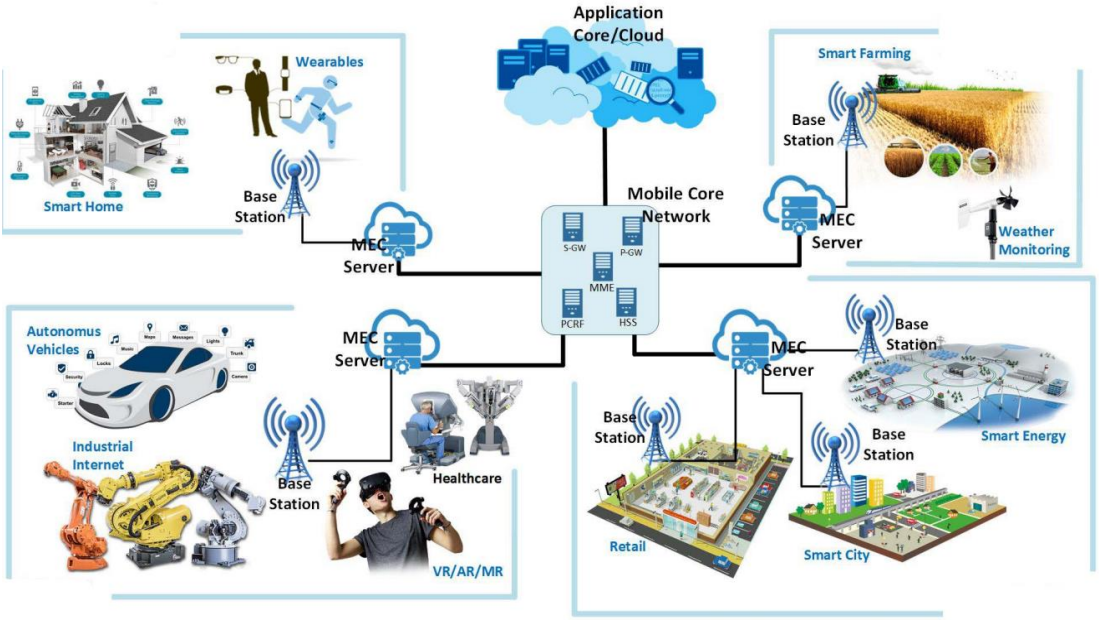

Figure 2.4.4 - 1 Multi-access Edge Computing [14]

2.5 CIA triad

CIA is a security model used to evaluate the security of systems and data. C stands for confidentiality, I stands for integrity and A stands for availability [15].

Confidentiality means that the sensitive data are protected to not be accessed by unauthorised adversaries. This can be achieved by encryption or by adding access restrictions.

Integrity refers to ensuring the information has not been modified during the communication. It promises valid data to the receiver, that were unchanged from the moment they were send by the sender. Integrity can be achieved by digital signatures, and checksums.

Availability ensures that the system and data is available anytime, meaning that whenever there are request for such services or information, the system is accessible. Availability can be achieved by ensuring fault tolerance and load balancing.

Chapter 3

Background information about Machine Learning

3.1 Reinforced Learning

Reinforced Learning (RL) is a type of machine learning method that an agent learns by experience while interacting with its environment [16]. Through trial and error, the agent receives rewards or punishments by the critic of the environment, and its goal is to maximize the reward received.

RL consists of the environment, the agent, or learner, and the actions an agent can take. Continuing, the environment provides the inputs, the critic that evaluates the actions taken, and the reinforced signal. The reward is maximized since, just like in human psychology, the agent tends to choose actions that previously received a higher reward. The reward received represents the reward of current action, not what is best for the long term. In addition, the agent should choose a policy, greedy, non-greedy, or balanced. The policy specifies how the agent should make decisions, by exploration or exploitation. Exploration means the agent chooses to explore new options, and exploitation means the agent uses its knowledge. Greedy policy is based on exploiting knowledge, non-greedy is based on exploring new states, and balanced is a combination of both.

As mentioned above, the agent learns with the goal of maximizing its cumulative reward. The decisions are made consulting the value, and weight function, where the value function helps the agent get the optimal known path to maximize reward. The value function takes into consideration whether future rewards are taken more into consideration than immediate rewards. The weight function helps the agent to determine the importance of the knowledge previously acquired, in order to apply it to make a better decision. The weights are updated according to the error function, which is calculated based on the reward received by the critic, meaning if the agent got higher reward, the error is lower, and if the agent gets punished, the error is higher, but in both situations the weights change accordingly based on the error.

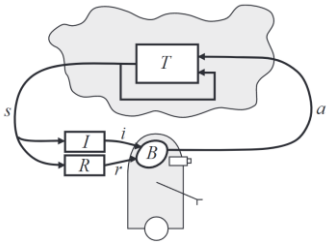

Figure 3.1 - 1 Simple RL model [16]

3.2 Deep Learning

Deep Learning is a machine learning method where the model learns based on training and testing data [17]. The training data is used to train and therefore create the model, and after many epochs to reach convergence, the testing data is used to evaluate the accuracy of the model on new data.

The models consist of three kinds of layers, input layer, multiple hidden layers, and output layer. The models can have only one input and output layer but can have multiple hidden layers. For each link in between layers there are weights that are considered when computing the sum for the activation function at the nodes of the following layer. The activation function represents a threshold, and if the sum of the linked nodes based on their weights surpass the threshold, the node is activated. Finally, when reaching the output nodes, the node with the highest sum is activated and that node represents the actual output calculated by the model for the input. When the actual output is presented,

it is compared to the expected output from the training data. Then the model back propagates from the output layer back to the input layer, where in each layer it passes backwards, changes the weights of each node based on the error function, meaning based on the difference of the expected and actual output of the model. This is the training phase of a deep learning model, where this process is repeated for each entry of the training data, completing an epoch, and then the whole epoch is repeated multiple times until convergence.

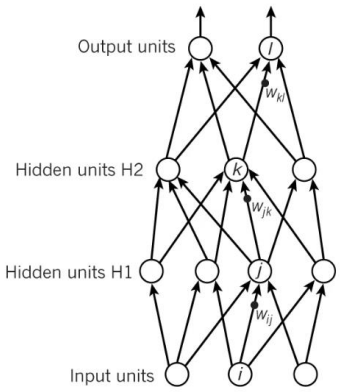

Figure 3.2 - 1 Deep Learning model with two hidden layers [18]

Following, the testing phase just runs new data on the trained model and calculates the accuracy of the model by comparing the output of the model to the expected output. The issue of overfitting can be presented in testing, by observing high detection accuracy on training data but low detection accuracy on testing data. Overfitting is when the model has been overtrained, and instead of generalizing, it memorized the input and output result. Overfitting leads to bad performance of the model on data that was not a part of the training data, in other words unseen data.

3.3 Deep Reinforced Learning

Deep Reinforced Learning (DRL) is a Reinforced Learning machine learning method enhanced with a little bit of Deep Learning [19]. The environment is the same as Reinforced Learning, it still has the agent, the critic, and the reinforced signal. The difference between RL and DRL is in the policy and value functions [20]. In RL the policy and value functions are represented by an equation, in contrast to DRL that those

functions can be calculated through a DL model. For example, for the value function, the DL model takes as input the state and action of the agent and outputs the estimated expected reward. The cost function is introduced to evaluate and improve the accuracy of the estimated reward and it is based on the difference of the expected and actual reinforced signal. Moreover, if the DL model needs to represent the policy function, it takes as input the state and the output activated represents the action chosen based on agent's policy. The weights of the DL model are still updated normally based on gradient descent to optimize the functions. Finally, even though DRL is not that different than traditional RL, the combination enhanced RL.

3.4 Q-Learning

Q-Learning is a Reinforced Learning algorithm that focuses on the maximization of the long-term cumulative reward, in contrast to RL where the discount factor defines whether the reward needed is based on the immediate or distant future [21]. Needless to mention that Q-Learning since it is a RL algorithm, it inherits the same logic of learning with a critic. The cumulative reward is represented by the Q-function. The downside of the algorithm is that in order to calculate the optimal Q-function, the model needs multiple iterations to update its lookup table with more knowledge [22].

3.5 Deep Q-Network

Deep Q-Network (DQN) is a combination of Reinforced Learning, that maximizes the reward, Q-Learning, that maximized the long-term cumulative reward, and Deep Learning, that optimizes the policy and value functions [23]. DQN enhances those models by improving the Q-function, by stabilizing the learning and minimizing the error of the Q-values. DQN uses DL to calculate the Q-function which returns the expected long-term reward, so an agent can choose more correctly which action to take based on the available inputs. Moreover, DQN uses experience replay to stabilize the learning process. In experience replay, a small batch of recent past experiences are saved and used during training to update parameters, with the goal to minimize the correlations between sequential data. DQN not only uses an error function for the expected and actual Q-value, but it also performs gradient descent to minimize the difference between the calculated

and actual Q-value of the model, and thus training the model with higher accuracy. Even though DQN is a combination of other algorithms, it brought innovative improvements that enriched the machine learning techniques.

3.6 Federated Learning

Federated learning is a machine learning approach where multiple nodes work independently on their models, in order to be aggerated into a shared model with comprehensive knowledge [24]. This is achieved by nodes sharing their trained model to a node called the aggregator. The aggregator is responsible for aggregating the parameters from all the models received by creating a global model. Then the global aggregated model is returned to the individual nodes by the aggregator, so the nodes have enhanced knowledge that came from other nodes. Later, the nodes send updates of their models to keep the global model up to date. Since the training is done at the nodes, and not done by a central unit, it is considered a decentralized approach.

The benefit of federated learning is that, since the node only sends a trained model to the aggregator, the private local data that was used to train the model is not distributed, safeguarding the privacy of the node's data. In addition, another benefit is that a node has limited data to train a model, so its model's knowledge is enhanced by models from other nodes.

Federated learning is a popular machine learning approach in wireless networks since conventional machine learning approaches are not appropriate for their complicated nature. It is not always possible in wireless networks to have a central node to store and process the training data, therefore this distributed machine learning solution is more suitable. Networks may also be resource restricted, federated learning combats this by sending models for aggregation instead of multiple data per node, reducing traffic in the network [25]. Federated learning is also appropriate for heterogenous networks, since each network can build its own model based on its attributes to be aggregated and distributed. Furthermore, federated learning is fault tolerant [26]because if a node leaves unexpectedly due to faults, the model can continue the aggregation normally when using an aggregation algorithm like FedAvg [27], which provides an average aggregation based

on the present nodes. The only negative of continuing without the faulty node is that if the node was a part of heterogenous network, the model will no longer have a complex sample. Finally, to not forget the main and most important benefit, that federated learning preserves the privacy of a node instead of sending confidential data for training, it sends a trained model.

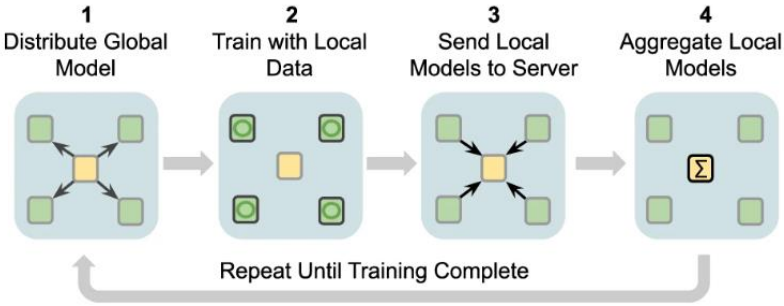

Figure 3.6 - 1 Federated learning training [28]

Chapter 4

Threat landscape of 5G

4.1 Authentication Key Agreement (5G-AKA) protocol

The 5G-AKA, as specified in 3GPP TS 33.501 v0.7.0, is the protocol responsible for authentication and key agreement of a user to a serving network (SN), by establishing shared secret keys between them. The 5G-AKA is based on the previous generations AKA protocol, the EPS-AKA protocol from 4G LTE, but with enhanced privacy since the new generation's AKA uses asymmetric randomized encryption, since the User Equipment (UE) uses the public key of its Home Network (HN) to encrypt its Subscription Permanent Identifier (SUPI). The newly introduced asymmetric encryption protects the users against IMSI-catcher attacks, where an attacker eavesdrops on an authentication session to get the SUPI of a user and impersonate the user [29].

First, a normal execution [30] as seen in the Figure 4.1-1, the protocol starts with a user sending its Subscription Concealed Identifier (SUCI) from the HN to the SN Security Anchor Function (SEAF), to check user's identity before granting access in the SN. After, the SEAF sends a 5G Authentication and Authorization Request (5G-AIR) message to the Authentication Server Function (AUSF), to request authentication from the SN. The AUSF also sends an Auth-Info Request message to the Authentication credential Repository and Processing Function (ARPF) to request the authentication information of the user, including user's long-term subscriber key. Following, the ARPF responds to the AUSF with an Auth-Info Response containing the required authentication information, the AUSF generates an authentication vector that responds to SEAF with in the 5G

Authentication and Key Agreement Request (5G-AIA) message. The SEAF requests authentication from the UE via the Auth-Req message, and the UE replies with the Auth-Resp, in order to authenticate and establish secure communication between the UE and the SN. Finally, the SEAF sent to the AUSF a 5G Authentication Challenge (5G-AC) message to verify that the UE has the correct session key to guarantee that the communication is secure.

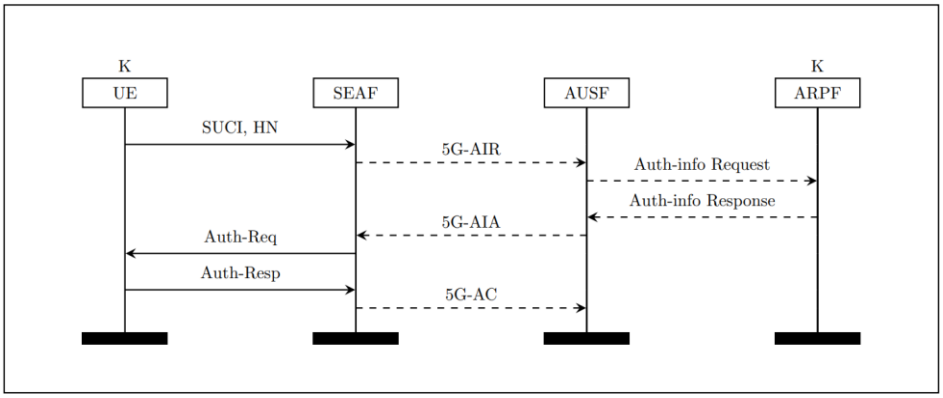

Figure 4.1 - 1 Normal Execution of 5G-AKA [30]

The vulnerability presented in the 5G-AKA protocol is that the protocol does not specify any security for the channel from the HN to the SN's SEAF, so when the UE sends its SUCI to the SEAF, the communication may not be secure, as depicted in Figure 4.1-2, therefore an intruder can eavesdrop on it and steal the user's SUCI. This vulnerability can lead to an impersonation attack.

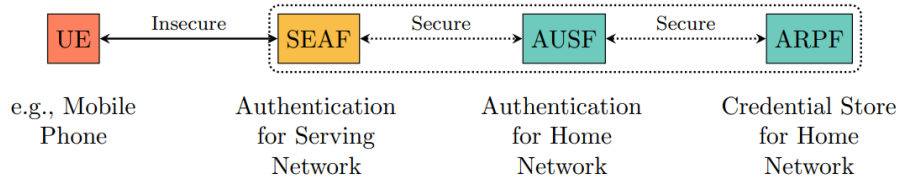

Figure42.1 - 2 Security of channels in 5G-AKA [30]

An Impersonation attack is an integrity and confidentiality attack since an intruder can access confidential information or manipulate information by impersonating a legitimate user of the network. This attack can be achieved when a malicious actor starts two

concurrent authentication sessions, one session with the overheard SUCI of a legitimate user, and a session with their own SUCI. The execution proceeds as described above in the normal execution, until the AUSF receives two parallel Auth-Info responses from the ARPF. The Auth-Info has no binding to the authentication session it belongs to, meaning there is no SUCI on the message, so there is a race condition, which may cause the AUSF to respond the wrong 5G-AIA to the incorrect user, that contains the generated K_{seaf} key which is used for further authentication in the SN. Since the AUSF confused the concurrent sessions, the SEAF continues responding to the sessions with the wrong messages. In this way, an attacker can get the K_{seaf} of a legitimate user in the SN and can use it to impersonate the user in the SN.

Acronym		Role
UE	User Equipment	Subscriber's physical device
HN	Home Network	Subscriber’s service provider
SN	Serving Network	Nearby network that controls the base station that the user is communicating
SUPI	Subscription Permanent Identifier	Unique identifier for UE, assigned by the HN so it is only shared between the UE and HN, and it should never be exposed publicly
SUCI	Subscription Concealed Identifier	To not expose the SUPI, authentication and authorisation process uses a SUPI encrypted with the public key of the HN
SEAF	Security Anchor Function	Responsible for verifying user's identity to provide access to the network
AUSF	Authentication Server Function	Responsible for the authentication and key agreement, and generates the authentication vector for the user creates an authentication vector for a user based on their SUCI
ARPF	Authentication credential Repository and Processing Function	Stores the required authentication information for users

5G-AIR	5G Authentication and Authorization Request	Requests authentication from the SN
5G-AIA	5G Authentication and Key Agreement Request	Provides the SEAF with the 5G-AIR requested authentication information
5G-AC	5G Authentication Challenge	Verifies that the UE has the correct session key to guarantee secure communication

4.2 Network Slicing

Network Slicing divides the physical network into multiple logical ones over a share infrastructure. One benefits for security provided by network slicing is the isolation of the slice. Meaning that technically if a slice was compromised the attack should not spread to the other slices of the network. The thread vectors of network slicing can be divided according to the lifecycle of a slice, which consists of the preparation phase, installation, runtime, and decommissioning, and the other thread vectors can be divided into threats in the slice, intra-slice, and in between slices, inter-slice [31].

First of all, the preparation phase is when the slice template is created. The main vulnerability of this phase is if the design of the template was incorrect, injected, or not up to date to security standards, all the slices created following the template will inherit the vulnerabilities, affecting the integrity of the system, the isolation of slices, or confidential information. Moving to the installation phase of the lifecycle of a slice, where in this phase the slice is installed, configures, and activated. The main point of attack during this phase is to either reconfigure the slice, or create fake slices, which can be achieved by manipulating the API, in order to authorize an adversary into the network and tamper the slice. After activation, the next phase of the lifecycle is the runtime phase, where the API is still the point of attack. In this phase a slice is vulnerable to attack to compromise the availability of the system, for example DoS, and the confidentiality via privacy leaks. Another availability attack during the runtime phase is the deactivation of a slice that can lead to service disruption across multiple slices. The final stage of the lifecycle of a slice is the decommissioning phase, where the slice is being deactivated. Even though the slice is terminated there are still vulnerabilities linked to the phase. One

of the vulnerabilities is not handling the slice data properly at deactivation, which may lead that data to be available and ready to be exposed even after deactivation. Another vulnerability marked in the decommissioning phase is the usage of not freed resources of the slice in order to conduct a DoS attack against the slicing system.

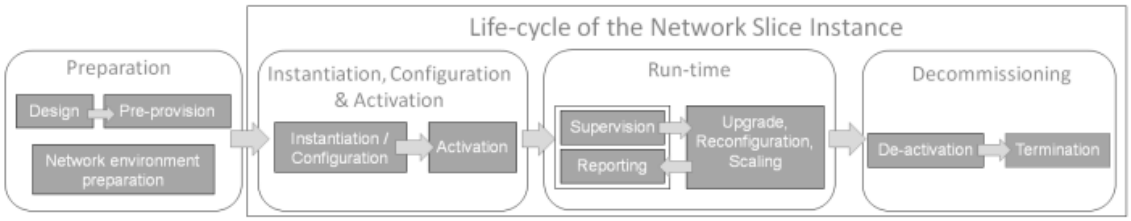

Figure 4.2 - 1 Lifecycle of a Network Slice [12]

The next thread vector is the intra-slice threats, which can appear in the devices of the slice, in the services interface, and in the network functions. Starting from the devices in the slice, which are the weakest point of attack. A malicious device with unauthorize access brings privacy concerns for the other nodes within the slice, or even perform DoS from inside of the slice. Another threat within the slice, is the disruption of communication of the slice with services, by attacking the interface in-between them. By the disruption of the services an attacker can achieve damaging other services that are depended on it. The network functions of the slice can also be vulnerable to attacks, via physical attacks, and can be tampered to cause resource depletion.

Even though network slicing promises isolation in between slices, there are still threats concerning the security inter-slice. To begin with the device of a slice. A device from a slice can try to gain access in another slice, which can be considered easier since the malicious device is already in the system. The intruder may attempt to conduct a DoS attack in order to consume reduces, and therefore reducing the resources for easier authorization. Furthermore, even if the slices are isolated in-between them, they still use the same services, so by damaging a service from a slice, the damage will spread to other slices that use that service, compromising the availability of the whole system. Finally, an important note for network slicing, is that the system is just as weak as the weakest slice, or sub-slice, since if an attacker can gain access or compromise a slice, the attack can spread through changing or leaking shared parameters or depleting shared resources.

4.3 Multi-access Edge Computing

As mentioned previously in subchapter 2.3.4, Multi-access Edge Computing (MEC) is an innovative technology in 5G that brings computational and storage services on the edge, closer to the end users, reducing latency and enabling real-time applications. MEC brings many advancements in 5G, yet its security and privacy should be taken into consideration. Its threat vectors can be divided into three sections, threads in the Access Network, threats in Mobile Edge Network, and threats in the Mobile Core Network [32].

Starting from the access network, where the vulnerabilities can be detected in the end nodes and their communication channels. The Users' Equipment (UE) is a vulnerable component of MEC, since they can be easily affected by physical attacks or malware, which can lead to unauthorized access in the network or resource depletion. The UEs are interconnected in an ad-hoc manner, meaning the access network inherits the vulnerabilities of Device-to-Device (D2D) communication, for example Distributed Denial-of-Service (DDoS), eavesdropping, impersonation, etc. Another vulnerable link in the access network, is the one between the UEs and the Base Station (BS). The link can suffer from hijacking, jamming, or even malicious node injection.

Next up is the threats of Mobile Edge Network (MEN), meaning the vulnerabilities of the MEC components in the edge and host level. A component of MEC is the Mobile Edge Host (MEH) which is responsible for computational and storage operations of mobile edge applications. When compromised the MEH can led to false resource allocation at the Mobile Edge Platform (MEP) and resource exhaustion at Virtualization Infrastructure (VI) due to service continuation. It can also lead to feeding inaccurate feedback to the Mobile Edge Platform Manager (MEPM), that can cause misconfigurations, service disruption and resource depletion. MEPM is responsible for MEH monitoring, resource monitoring and allocation, and due to infected applications can lead to device disruption and feeding wrong data to upper system components. Another component of MEC is the Virtualization Infrastructure Manager (VIM), which manages the virtualization resources. The compromise of VIM has the goal of resource exhaustion via Virtual Machine (VM) based attacks. Components of MEC are not the only things vulnerable in

MEC, channels of communications are also vulnerable. For example, the link between MEHs is vulnerable to injections, or to be infected by other MEHs. In addition, the channel connecting the MEHs to third party cloud servers can be vulnerable to attacks like Man-in-the-Middle, masquerading, injections, wormhole attack, or even packet sniffing compromising confidentiality.

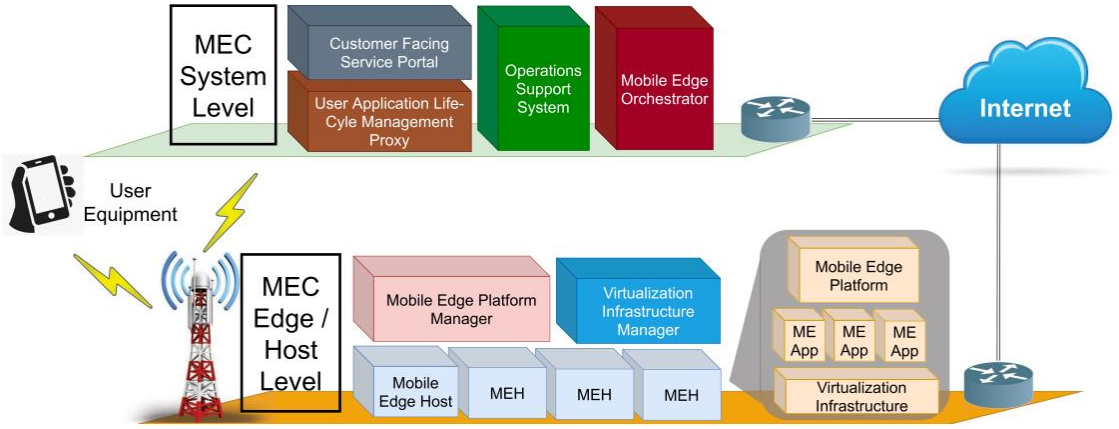

Figure 4.3 - 1 Structure and components of MEC [33]

Acronym		Role
MEN	Mobile Edge Network	MEC components
MEH	Mobile Edge Host	Responsible for computational and storage operations of mobile edge applications
MEP	Mobile Edge Platform	Responsible for resource scheduling and allocation of mobile edge applications
VI	Virtualization Infrastructure	Platform that provides virtual computational and storage resources for mobile edge applications
MEPM	Mobile Edge Platform Manager	Responsible for MEH monitoring, and resource monitoring and allocation
VIM	Infrastructure Manager	Manages the virtualization resources.

UALCMP	User Application Life-Cycle Management Proxy	Responsible to handle app requests from UEs and to manage their lifecycle, while linking UE applications to mobile edge application
MEO	Mobile Edge Orchestrator	Hypervisor of MEC, that manages the MEC hosts, resources, VMs, services, and traffic
OSS	Operation Support System	Responsible of the requests that go through UALCMP and getting control information from MEPM and Mobile Edge Orchestrator (MEO), for authenticating requests to go through UALCMP, and register while starting and ending the lifecycle of mobile edge applications, and approving nodes to use said edge applications

The last threat vector is the Core Network, which is the system level devices, all the way to the backhaul that connects to the Internet. Component of the Core Network is the User Application Life-Cycle Management Proxy (UALCMP) which is responsible to handle app requests from UEs and to manage their lifecycle, while linking UE applications to mobile edge applications. The attacks on UALCMP target service disruption due to DoS/DDoS or by increasing the lifecycle of apps, or they target obtaining unauthorized access by masquerading attacks. Mobile Edge Orchestrator (MEO) in the Core Network is the hypervisor of MEC, that manages the MEC hosts, resources, VMs, services, and traffic. The MEO acts as a hypervisor for the MEC system since it observes the MEC hosts, VMs and resources at the edge. Compromising the MEO of the system can lead to resource depletion or compromising the communication channel in between the MEO and the 5G Core, which is vulnerable to TCP/IP attacks such as eavesdropping, DoS and spoofing. The Operation Support System (OSS) is also a part of the Core. OSS is an important component since it is responsible of the requests that go through UALCMP and getting control information from MEPM and Mobile Edge Orchestrator (MEO), meaning the availability of the component shall not be compromised by a DoS or DDoS attack. In addition, OSS is responsible for authenticating requests to go through UALCMP, register while starting and ending the lifecycle of mobile edge applications, and approving nodes

to use mobile edge applications. A risk in the OSS is injection, since the application registers in the OSS, a malicious actor may try to inject code in the OSS or impersonate valid entries of applications to gain access.

Chapter 5

Machine learning for secure communications

5.1 Federated Learning for 5G HetNet cooperation security

This paper [34] proposes a Federated Learning approach to enhance the security of Heterogeneous Networks (HetNets). This approach incorporates attack detection in each layer and enriches the threat information with the cooperation of the layers. Given their diverse range of devices and distributed architecture, traditional security solutions are insufficient for HetNets. Therefore, this solution uses nodes from upper layers to enhance detection models of lower layers and uses other networks' knowledge to improve the security of similar networks, taking advantage of the factors that complex Hetnet's security.

The solution based on federated learning aims to deploy attack detection in the end, edge, and cloud. Attack detection in the end nodes is used for to secure the local access network. This Deep Reinforced Learning (DRL) detection model will be only deployed on end nodes with efficient computational power since the end nodes of the network are computationally restricted. Besides, the end nodes only have partial knowledge, based on their experiences, but full knowledge is needed when it comes to security. As a result, an edge node from upper layer will aggregate the end nodes' local trained models to create a detection model with full knowledge, and it will return the integrated model with enhanced dataset to end nodes.

The detection model in the end nodes will be trained using the Deep Q-Network (DQN) algorithm, due to its adaptability in dynamic environments. The knowledge needed for DQN is the current states, the action selected a, the reward r from environment after the action a, and the next state s'. The states of the end nodes are represented by the features of the transmitted packages in the local network. An action a is represented by the values 0 and 1, where 0 means that the node does not detect a risk and 1 that node detects a risk. The reward r can take 4 values, 1, -100, -1, and 100. If the end node takes a correct action in a secure network, then it is rewarded with the value 1, but if the node takes a wrong action in a secure network, then it is given the punishment of -1. In the scenario where the local network is in a risk, and the node takes a correct action, it is rewarded with 100, but when the network is at risk and end node take a wrong answer, it is punished by -100. The current network is updated based on the gradient descent of loss function of DQN:

$L_i(\theta_i) = E_{(s, \alpha, r, s')} [(r+ \gamma \max_{\alpha'} Qtarget(s', \alpha'; \theta_i) -Q(s, \alpha; \theta_i))^2]$

where E is the expected value of sampled experiences in the replay buffer, γ is the discount factor, Qtarget is the target network which is a more stable copy of the Q-network due to periodic updates, θ is the parameters of the node i, and r, a, s are the reward, action and state accordingly as mentioned above. In addition to the information aggregation, the edge node calculates the parameters θ of the nodes, meaning the update of the weight and biases of the nodes, and then informs the end nodes the updated parameters. With such detection model, an end node after completing training can not only perform attack detection, but also share to an edge node its model parameters to be shared to other nodes.

Continuing to the edge, the bridge in between the end and the cloud. The edge connects the local access network to the 5G backbone network. As a result, the edge has full knowledge of the security in the local access network, but only has partial knowledge about the cloud's security. The edge node's role apart from decreasing the training speed due to model aggregation for the end nodes, is to also attack detection for itself, since the edge is battling threads from both the local and the backbone network. In addition, just like previously in the end nodes, in the edge layer too the upper layer, meaning the cloud, aggregates the models from the edge and returns to them an enhanced model.

The detection model of the edge is based on Deep Learning. The training data of the model has as input transmitted packets and as output the verdict if the packets are malicious or not. The model is updated using the gradient descend with the goal to minimize the average cumulative error:

$$E_k = | y_k^{out} - y_k^{lab} |$$

where Ek is the error of node k, y_k^{out} is the expected output of the given input of node k, and y_k^{lab} is the actual output of the model with the given input of node k.

$$\sum^{m} \frac{E_k}{m}$$

where E is the average cumulative error, Ek is the error of node k, and m is the number of training inputs.

Any attack samples detected by an edge or from the end nodes are stored in a local database and the edge nodes share those samples to the cloud to aggregate them in order to produce a more accurate model.

Moreover, since Heterogeneous Networks are characterized by their heterogeneity, not all models' parameters can be aggregated and intergraded into other local access networks. The cooperation of the cloud in this federated learning approach is to match parameters of the edge models to other networks, and to aggregate appropriate threat information for matching networks. This can be achieved because when an edge sends to the cloud an attack detection model to aggregate, it also sends information to the cloud about the environment of the local access network such as security level, density, topology, etc. Then all the information of the local access networks is stored in a database and compared with the other networks' information. In that way, the aggregated model is constructed based on networks with matching environments, and thus the appropriate enriched model is returned to the appropriate networks.

The authors of the paper also add the performance of their machine learning solution. They present that initially in the local access network, each training set of a node contains packets of both malicious from various attacks and non-malicious packets, therefore nodes got initially partial knowledge. The results shown in Figure 5.1-1, of the converge shows that over multiple epochs the federated learning cooperative scenario with ten end nodes had reached higher reward over fewer iterations in comparison to the same scenario

with less nodes which was the second that reached the high continuous reward, and over the other schemes. Though this performance it can concluded that when using the cooperating proposed federated learning solution the convergence time needed is less rather than when having a non-cooperative scheme or traditional federated or distributed learning, and the solution is more effective when having more cooperating nodes. A graph about the detection error is also presented in Figure 5.1-2 from where it can be concluded that the scheme proposed with the higher cooperation has the lowest error, meaning higher accuracy, in comparison the same scheme with lower cooperation or even to traditional scenarios. Furthermore, the authors of the paper state that the solution can be adapted and applied for intrusion prevention and vulnerability scanning, apart from attack detection.

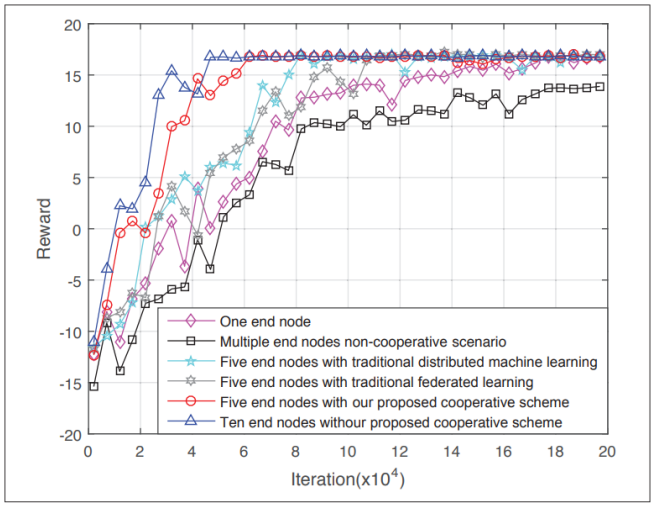

Figure 5.1 - 1 The average reward of the end nodes [34]

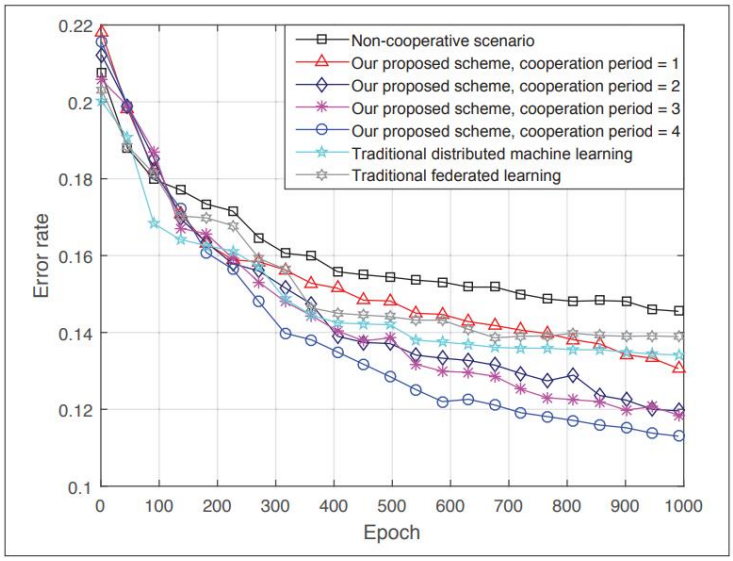

Figure 5.1 - 2 Error rate of the edge nodes based on cooperation periods [34]

Concluding, the authors proposed a federated learning solution for 5G heterogeneous networks where the end, edge and cloud cooperate to improve the attack detection, decrease training time, and aggregate models of matching network environments. This solution improved the security in such distributed network, while preserving privacy of nodes. The performance of the approach shows that the HetNet can be secured over less epochs and with less errors than traditional security solutions. Finally, the solution can be adapted for other security purposes such as intrusion prevention and vulnerability scanning, thus enhancing 5G HetNets security.

5.2 Reinforced Learning for mobile offloading against jamming in Multi-access Edge Computing

Multi-access Edge Computing (MEC) is a new emerging technology of 5G. MEC brings closer to end users' computation and storage resources by placing edge nodes in-between end nodes and the cloud. As a result, MEC reduces latency and bandwidth requirements. MEC also enables mobile edge caching and offloading. Mobile edge caching reduces the traffic load and therefore the latency of the network by reducing duplicate transmissions. Offloading improves the performance of an application in a mobile device by transferring the application to an edge node with higher computational power.

In addition, MEC can improve the security of the network by monitoring and by bringing closer to the end devices security functions. On the other hand, as a distributed system, MEC introduces a new threat environment in 5G. Some vulnerabilities presented in MEC offloading include compromising weak nodes to infiltrate the network and attack other nodes, jamming, rogue node, eavesdropping, man-in-the-middle and smart attacks, and in MEC caching privacy leakage can be presented. As per the authors in [35], MEC security need better solutions since most of them are fixed on a network or attack model or are not practical since the parameters are not only difficult to be gathered by an edge node, but also change over time for optimization. Their proposed solution is to secure offloading against jamming attacks by using Reinforced Learning (RL).

Each mobile device in a MEC system must decide on its offloading policy, determining parameters like the data to offload, the channel and transmit power, and the which edge node to transmit. The policy is based on maximizing Signal-to-Interference-plus-Noise Ratio (SINR) and minimizing Bit Error Rate (BER) of the signals received by the edge nodes, mitigating the effects of jamming and interference, and therefore optimizing the quality of offloading. A mobile device can dynamically choose the optimal policy using reinforced learning, without having full knowledge about the jamming or MEC model. A Reinforced Learning (RL) model only needs as parameters the state of the device, its available actions, its chosen offloading strategy, and jamming sources.

The state of the mobile device can be calculated from the received jamming power, bandwidth, and user density. The device's decision on offloading policy in the presence of jamming can be guided using the Q-function, that calculates the reward for each action on the current state based on the current knowledge of the RL. The Q-function is updated each time the mobile node decides on an offloading policy and visits a new state, meaning new knowledge to take into consideration by the system. An e-greedy algorithm is recommended for mobile networks, to explore the dynamic environment leading to an optimal policy.

Unfortunately, the disadvantage of Q-learning is its performance in a network, especially if smart attackers are present. Since the RL uses its prior knowledge that is based on explored states for the Q-function, the node would have to explore all its possible

combinations of actions and states to converge to an optimal solution, before the network state changes, or before the policy of the jammer changes. For those reasons, the author presents Dyna-Q, an extension of the Q-learning algorithm, that in addition to real defence experiences, it also uses virtual experiences to train the model. These virtual experiences may not always be accurate, but they decrease the learning duration of the model. A reinforced learning model needs accurate information with no delays, which can be a difficult task for a mobile node to gather fast the current network state to choose offloading policy. Moreover, the need for balance of exploration and exploitation of reinforced learning, leads the model to choose wrong policies in order to learn and to lead to the optimal solutions, especially in the early learning stages, marking this learning process as risky and dangerous when applied to edge security.

On the other hand, this machine learning approach offers several benefits too, for instance it has low computational overhead, meaning it can be applied to both mobile and edge nodes of the MEC system. In addition, when using reinforced learning, the node does need any knowledge of the network nor attack model, which is an important benefit when dealing with MEC nodes with low computational, energy and storage requirements. Lastly, the implementation of Q-learning to a MEC system invaded by a jammer, guarantees to converge to the optimal solution, securing the network of jamming.

5.3 A Federated Learning Approach for Improving Security in Network Slicing

One of the new technologies introduced in 5G is network slicing. Network slicing is an architecture that allows the creation of multiple isolated logical networks over a common physical network. The division to multiple logical networks meets the need for the diversity of services' requirements in the network. Many services have diverse requirements in terms of their latency, bandwidth, scalability, Quality of Service (QoS), and security. The logical division of the network optimizes resources of the network, while enhancing the security of the network due to slices isolation. The slicing isolation though poses a disadvantage for machine learning models, due to information isolation for privacy reservation in-between slices. The approach proposed in [36] preserves the privacy of each slice, while improving security in a network slicing ecosystem, with the use of federated learning. The approach that promises pro-active security with a

centralized security orchestrator is called Federated Learning enabled Security Orchestrator (FLeSO).

FLeSO uses a federated learning approach since this machine learning algorithm preserves the confidentiality of the slices by not sharing the data of the slice, but instead it aggregates individual models of the slices into a singular model. The approach promises pro-active security in the slices, since if an attack is detected in a slice, the other slices will be updated of the detected attack thought the model aggregation, marking federated learning a fitting method for the privacy requirements in network slicing.

The FLeSO architecture promises features such as enhanced security, privacy, pro-active security, and centralized security management. Due to the distribution of the model information that FLeSO provides, the security of each network slice is enhanced by the distribution of the attack information by the aggregated model. Moreover, the privacy requirements of the network slicing ecosystem are achieved through the sharing of output model and not the sensitive information collected from each slice for model training. The approach is centralized managed, since in order the security operations can be performed by using the FLeSO architecture, and not the network slices.

The model proposed consists of different components as seen in Figure 5.3-1. First of all, the Security Orchestrator Client (SOC) is a node in each slice that is responsible for the security operation of the slice, in addition to collecting the slice information for training, sending the local trained model for aggregation, and receiving the aggregated model. The Federated Management Component (FMC) collects models from the SOCs, aggregates them, and distributes the output model for the slices. The aggregation method used in this approach is FedAvg due to its simplicity. Next, there is the Slice Security Monitoring Component (SSMC) that collects data from the SOCs and forwards them to the Data Evaluation Component (DEC), that analyses the information collected for attack detection and determines the action needed to mitigate the detected attack, and Solution Life-Cycle Management Component (SLCMC), that manages the deployment of security operations of the slice under attack. In addition, there is the System Evaluation Component (SEC) that is evaluates the operations performed by the network slicing environment, and the Security Solution Deployment Component (SSDC) that

collaborates with the Networks Slice Manager (NSM) to manage the security operations in the slices.

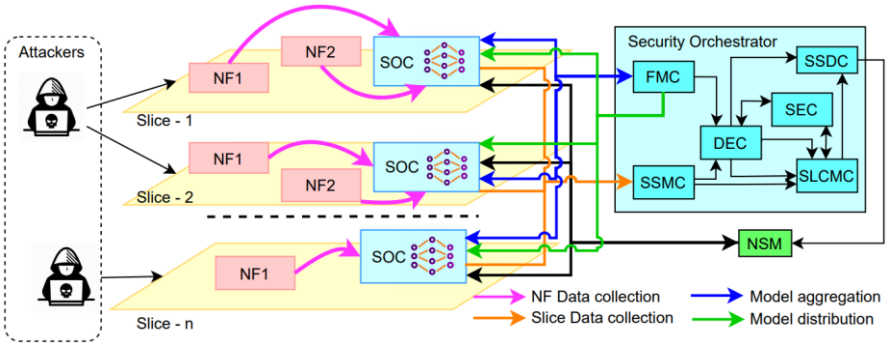

Figure 5.3- 1Model proposed for FLeSO [36]

The evaluation of the model was carried out on a real-world slicing testbed. A balanced of Independently and Identically distributed (IID) data set was used to detect 5 classes of data: normal, Denial of Service (DoS), User to Root (U2R), Root to User (R2U), and probe. The first feature evaluated was the architecture's performance on data distribution through model sharing, in comparison to legacy security models. As seen in Figure 5.3-2, the accuracy of FLeSO is overall higher than the legacy models. Next feature tested was the proactive security that the model provides. The network slice is supposed to detect attacks that were not present in the perspective slice. There were two experiments made for this feature, the first one each slice was presented with only one distinct attack, and the second scenario is that a slice can be affected by many attacks. In both scenarios, as seen in Figure 5.3-3 and Figure 5.3-4, the model was presented a combination of normal and attack data according to the scenario, with the results of both environments' network slices to be able to detect unseen for the slice attacks, meaning proactive security was achieved with security mechanisms in the environment. A downside of this feature is that to achieve such a result a lot of federated rounds are needed. Also, note that the U2R and R2U attack detection ratio is significantly lower, since there is a smaller amount of training data in the set for the two attacks. Following up, the convergence analysis of the model was based on different training IID data distributed to different slices and with different number of iterations. As shown in Figure 5.3-5, the models' convergence independently of the attack distribution to the slices after many federated rounds, but prior to that the accuracy of IID distribution and many attacks per slices distribution were

higher than the scenario of one attack per slice distribution approach. A limitation presented during the performance is the increase of network slices reduces the accuracy of the model, since the larger number of slices means larger distribution of data which reduced the performance of machine learning models.

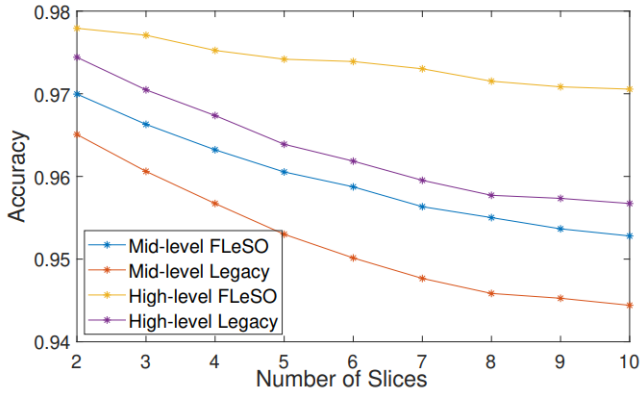

Figure 5.3- 2 Accuracy of FLeSO [36]

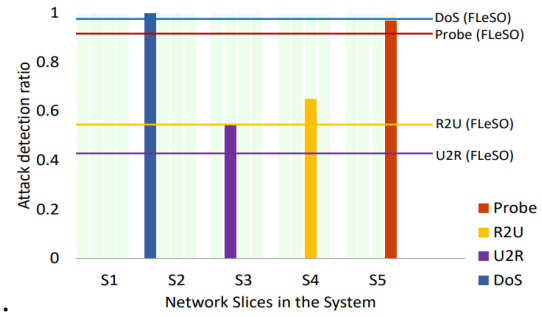

Figure 5.3- 3 Attack detection, only one attack per slice scenario

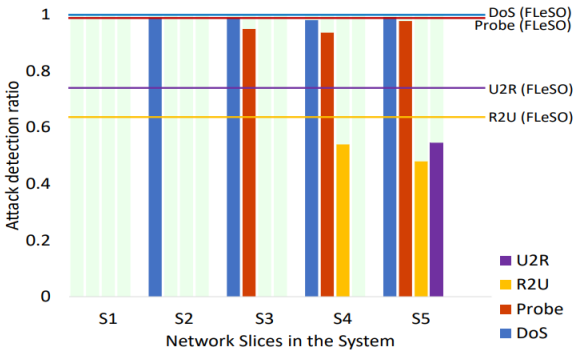

Figure 5.3- 4 Attack detection, multiple attacks per slice scenario [36]

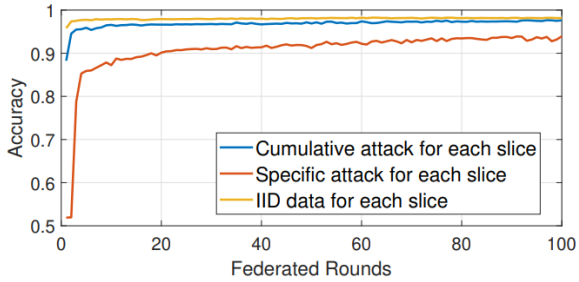

Figure 5.3- 5 Convergence of FLeSO [36]

In conclusion, the proposed solution of using federated learning for a more secure network slicing ecosystem, as it is tested on a real word network slicing environment, ensures not only enhanced security and privacy, but also pro-active slice security, with centralized security management. Finally, the FLeSO architecture outperforms the legacy security models, in terms of accuracy, data distribution in slices, and convergence, making it a great and feasible solution to implement in real world networks.

Chapter 6

Machine learning against secure communications

6.1 Label manipulation attack

In label manipulation, the attacker has access to the training dataset, and is able to modify the labels of the input data [37]. This label modification takes place during the training process of the model, and reduces the accuracy of the model, therefore violating its integrity. Even modifying a small percentage of the labeling, for example 10%, the accuracy of the model reduces to 90%.

6.2 Input manipulation attack

Input manipulation is an attack targeting the integrity of machine learning, in which data is added to the model during runtime [38]. The poisoned data, also known as adversarial examples, inserted resembles the data provided by the model's environment, but the adversarial examples gradually shift the output, decreasing the accuracy of the model and leading to misclassification. For example, in reinforced learning, an agent can be led to choose the wrong policy, by inserting poisoned data that affects the gradient ascent function that achieves optimization.

6.3 Model poisoning

One attack in federated learning is the model input manipulation attack [39]. In federated learning, the model is gradually taking as input models from nodes to aggregate. A malicious node can send to the aggregator a tampered model, and therefore spread the polluted data to the other nodes of the environment. Therefore, by poisoning a local model, the global model's integrity is affected.

6.4 Model extraction

An adversary can reconstruct the machine learning model via the model extraction attack. This can be achieved through black box querying. Black box adversaries have no information of the model’s parameters, architecture, or training data, but they can only query the model. By constantly querying the model and receiving output, the attacker can construct a similar new model, called shadow model [40], with the sample of data from the queries as training data. The sample data used for training constructs a model that imitates the target model with high accuracy. If the attacker has access to the class probabilities, the similarity of the adversary and target model increases, and the complexity of reconstruction decreases. The possession of an identical model affects the model’s integrity, due to adversarial example transferability, and confidentiality, due to model's information extraction.

Adversarial example transferability is a property of machine learning models [41]. This property implies that there is a probability that if an adversarial example is misclassified by the adversary identical model, then the target will misclassify the adversarial example too, affecting the integrity of the target model.

The extraction of the model's information can be achieved by the observation of the input and output data of the model. For example, model's parameters can be derived from the observation of the data. In addition, another issue with machine learning models is the unintentional memorization of data [42]. It is a rare occurrence that happens with unique sequences of the training data, which caused overfitting in the model. It is worth noticing

that the extraction of the model's parameters can be done not only with the help of an adversary model, but it can also be done on the target model too.

6.5 Membership attack

This confidentiality attack reveals to the adversary whether the input presented was in the training set of the model. The membership attack is a black box attack, meaning the adversary has no information about the model, and they only can interact with the model by querying an input and observing its output. The adversary can determine if the input belongs to the training set by observing the probability vector of the classes in the output of the model [43]. The probability vector shows the model's confidence value that the input value belongs in the respective class. The higher the confidence value, the higher of the accuracy of the output. Moreover, machine learning models' accuracy tends to be higher in data that were already inputs of the model, and higher precision if the input was a part of the training data, which can be viewed as a result of model's overfitting. The membership attack takes advantage of this property of the models, by corelating that the input with high confidence value belongs to the training dataset.
This attack affects the confidentiality of the model since the adversary can determine the private training data of the model that were collected from the environment of the model.

6.6 Model evasion

The model evasion attack goal is an adversary to evade detection by the machine learning model due to misclassification [44]. The attack is carried out during the testing phase of a model where the adversary must generate attack samples to evade the converged model. The adversarial data that will lead the model to misclassification can be perturbated in normal input data. The slight alteration of the malicious input cannot be observed by humans, but it still led to misclassification by a machine learning model, for example altering some pixels for a picture input in convolutional networks [45]. Model evasion by classifying adversarial data as benign affects the integrity of the model.

Chapter 7

Conclusions

7.1 Future works

Many machine learning models have been proposed to fortify the security of 5G. 5G is a heterogeneous network, something that should be kept under consideration. Therefore, more heterogeneous suitable machine learning approaches should be developed to be more realistic in a 5G network scenario. In addition, in order for the solutions to be more realistic, they should be tested on a real-world testbed. In addition, in a complex dynamic and scalable wireless network, like 5G, traditional machine learning will not be suitable since there is no continuous training. Online learning is a more realistic machine learning approach since online learning does not require a completed defined training set from the beginning, but instead it learns from batches of data [46]. Online learning benefits cellular networks since the machine learning model is not static but can scale and adapt based on the new data, new occurrences, or changes of the network.

Furthermore, online learning could prevent machine learning model attacks, like model extraction. Since the model is dynamic, and even if an adversary acquired the model, the dynamic model is going to be changed over time, so the adversary obtained an older, unusable version of the machine learning model. Moreover, another tactic of preventing machine learning attacks, for example label or input manipulation and model evasion, is introducing validation of information of the training and testing sets. Another powerful tool to be considered to enhance the security of machine models is homomorphic encryption [47]. Homomorphic encryption enables the node to perform computations on

encrypted data. Homomorphic encryption ensures confidentiality since the data does not have to be decrypted in order to be used. On the other hand, these extra steps or turning the machine learning model into a dynamic one, would add additional computing and energy cost which may be critical in a network with restricted resources.

7.2 Conclusions

The introduction of 5G revolutionized cellular network technology, due to the new technologies introduced, such as massive MIMO, the mmWave, multi-access edge computing, and network slicing. It is beyond doubt that the fifth generation of cellular networks outperforms the previous generations, while also being the most secure compared to them. Also, even the vulnerabilities presented can be prevented by deploying various machine learning models. But machine learning is considered to be a double-edged sword for security, because even though the models are deployed to prevent vulnerabilities of the network, machine learning itself introduces new model vulnerabilities in the network. Nonetheless, machine learning is recommended by various researchers to mitigate challenges in 5G, and therefore fortifying its security. Finally, the fifth generation not only impacted, but also paved the way for the upcoming generations of cellular networks.

Bibliography

[1] A. U. Gawas, “An overview on evolution of mobile wireless communication networks: 1G-6G,” International Journal on Recent and Innovation Trends in Computing and Communication, vol. 3, no. 5, pp. 3130-3133, 2015.

[2] J. Lundberg, “Evolution of Wireless Communication Technologies,” 2020.

[3] E. Ezhilarasan and M. Dinakaran, “A review on mobile technologies: 3G, 4G and 5G,” pp. 369-373, 2017.

[4] D. a. Manoj, U. Shrivastava and J. K. Verma, “2021 International Conference on Computational Performance Evaluation (ComPE),” in 2021 International Conference on Computational Performance Evaluation (ComPE), 2021, pp. 365-371.

[5] “5G System Overview,” 3GPP, 8 August 2022. [Online]. Available: https://www.3gpp.org/technologies/5g-system-overview.

[6] S. Khwandah, J. Cosmas, P. Lazaridis, Z. Zaharis and I. Chochliouros, “Massive MIMO systems for 5G communications,” Wireless Personal Communications, vol. 3, pp. 2102--2115, 2021.

[7] J. Wang, W. Deng, X. Li, H. Zhu, M. Nair, T. Chen, N. Yi and N. Gomes, “3D beamforming technologies and field trials in 5G massive MIMO systems,” IEEE Open Journal of Vehicular Technology, vol. 1, pp. 362--371, 2020.

[8] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin and R. Zhang, “An Overview of Massive MIMO: Benefits and Challenges,” IEEE Journal of Selected Topics in Signal Processing, vol. 8, no. 5, pp. 742-758, 2014.

[9] C. Seker, M. T. Güneser and T. Ozturk, “A review of millimeter wave communication for 5G,” in 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2018, pp. 1--5.

[10] R. F. Olimid and G. Nencioni, “5G Network Slicing: A Security Overview,” IEEE Access, vol. 8, pp. 99999-100009}, 2020.

[11] S. Zhang, “An Overview of Network Slicing for 5G,” IEEE Wireless Communications, vol. 26, no. 3, pp. 111-117, 2019.

[12] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini and H. Flinck, “Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions,” IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2429-2453, 2018.

[13] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, “On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration,” IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1657-1681, 2017.

[14] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila and T. Taleb, “Survey on Multi-Access Edge Computing for Internet of Things Realization,” IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2961-2991, 2018.

[15] L. O. Nweke, “Using the CIA and AAA models to explain cybersecurity activities,” PM World Journal, vol. 6, no. 12, pp. 1--3, 2017.

[16] L. P. Kaelbling, M. L. Littman and A. W. Moore, “Reinforcement learning: A survey,” Journal of artificial intelligence research, vol. 4, pp. 237--285, 1996.

[17] A. Shrestha and A. Mahmood, “Review of Deep Learning Algorithms and Architectures,” IEEE Access, vol. 7, pp. 53040-53065, 2019.

[18] Y. LeCun, Y. Bengio and G. Hinton, “Deep learning,” Nature, p. 436–444, 2015.

[19] A. Plaat, Deep Reinforcement Learning, Springer, 2022.

[20] Y. Li, “Deep reinforcement learning: An overview,” arXiv preprint arXiv:1701.07274, 2017.

[21] C. J. Watkins and P. Dayan, “Q-learning,” Machine learning, vol. 8, pp. 279--292, 1992.

[22] B. Jang, M. Kim, G. Harerimana and J. W. Kim, “Q-Learning Algorithms: A Comprehensive Classification and Applications,” IEEE Access, vol. 7, pp. 133653-133667, 2019.

[23] M. Roderick, J. MacGlashan and S. Tellex, “Implementing the deep q-network,” arXiv preprint arXiv:1711.07478, 2017.

[24] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li and Y. Gao, “A survey on federated learning,” Knowledge-Based Systems, vol. 216, p. 106775, 2021.

[25] S. Niknam, H. S. Dhillon and J. H. Reed, “Federated learning for wireless communications: Motivation, opportunities, and challenges,” IEEE Communications Magazine, vol. 58, no. 6, pp. 46--51, 2020.

[26] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, “Federated learning: Challenges, methods, and future directions,” IEEE signal processing magazine, vol. 37, no. 3, pp. 50--60, 2020.

[27] L. Collins, H. Hassani, A. Mokhtari and S. Shakkottai, “Fedavg with fine tuning: Local updates lead to representation learning,” arXiv preprint arXiv:2205.13692, 2022.

[28] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman and K. Maier-Hein, “The future of digital health with federated learning,” NPJ digital medicine, vol. 3, no. 1, p. 119, 2020.

[29] A. Koutsos, “The 5G-AKA Authentication Protocol Privacy,” 2019 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 464-479, 2019.

[30] M. D. Wild and . C. Cremers, “Security vulnerability in 5G-AKA draft,” pp. 14--37, 2018.

[31] R. F. Olimid and G. Nencioni, “5G network slicing: A security overview,” IEEE Access, vol. 8, pp. 99999--100009, 2020.

[32] P. Ranaweera, A. D. Jurcut and M. Liyanage, “Survey on multi-access edge computing security and privacy,” IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1078--1124, 2021.

[33] P. Ranaweera, A. Jurcut and M. Liyanage, “MEC-enabled 5G use cases: a survey on security vulnerabilities and countermeasures,” ACM Computing Surveys (CSUR), vol. 54, no. 9, pp. 1--37, 2021.

[34] Y. Wei, S. Zhou, S. Leng, S. Maharjan and Y. Zhang, “Federated Learning Empowered End-Edge-Cloud Cooperation for 5G HetNet Security,” IEEE Network, vol. 35, no. 2, pp. 88-94, 2021.

[35] L. Xiao, X. Wan, C. Dai , X. Du, X. Chen and M. Guizani, “Security in Mobile Edge Caching with Reinforcement Learning,” IEEE Wireless Communications, vol. 25, no. 3, pp. 116-122, 2018.

[36] S. Wijethilaka and M. Liyanage, “A Federated Learning Approach for Improving Security in Network Slicing,” GLOBECOM 2022 - 2022 IEEE Global Communications Conference, pp. 915--920, 2022.

[37] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu and V. Leung, “A survey on security threats and defensive techniques of machine learning: A data driven view,” IEEE access, vol. 6, pp. 12103--12117, 2018.

[38] N. Papernot, P. McDaniel, A. Sinha and M. Wellman, “Sok: Security and privacy in machine learning,” 2018 IEEE European Symposium on Security and Privacy, pp. 399--414, 2018.

[39] S. Niknam, H. Dhillon and J. Reed, “Federated learning for wireless communications: Motivation, opportunities, and challenges,” IEEE Communications Magazine, vol. 58, no. 6, pp. 46--51, 2020.

[40] S. Li, Y. Wang, Y. Li and Y.-a. Tan, “l-Leaks: Membership Inference Attacks with Logits,” arXiv preprint arXiv:2205.06469, 2022.

[41] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, B. Celik and A. Swami, “Practical black-box attacks against machine learning,” Proceedings of the 2017 ACM on Asia conference on computer and communications security, pp. 506--519, 2017.

[42] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos and D. Song, “The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks,” USENIX Security Symposium, vol. 267, 2019.

[43] R. Shokri, M. Stronati, C. Song and V. Shmatikov, “Membership inference attacks against machine learning models,” 2017 IEEE symposium on security and privacy, pp. 3--18, 2017.

[44] A. Ayub, W. Johnson, D. Talbert and A. Siraj, “Model evasion attack on intrusion detection systems using adversarial machine learning,” 2020 54th annual conference on information sciences and systems (CISS), pp. 1--6, 2020.

[45] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, B. Celik and A. Swami, “Practical black-box attacks against machine learning,” Proceedings of the 2017 ACM on Asia conference on computer and communications security, pp. 506--519, 2017.

[46] S. C. Hoi, D. Sahoo, J. Lu and P. Zhao, “Online learning: A comprehensive survey,” Neurocomputing, vol. 459, pp. 249--289, 2021.

[47] K. Munjal and R. Bhatia, “A systematic review of homomorphic encryption and its contributions in healthcare industry,” Complex & Intelligent Systems, pp. 1--28, 2022.

[48] P. A. Fouque, C. Onete and R. Benjamin, “Achieving better privacy for the 3GPP AKA protocol,” Cryptology ePrint Archive, 2016.