

TITLE OF INVENTION: A METHOD OF DETECTING AN ATTACK IN A CORE NETWORK OF A MOBILE COMMUNICATION NETWORK**TECHNICAL FIELD**

[0001] The present application relates generally to network security. More particularly, the present application relates to detecting an attack in a core network of a mobile communication network.

BACKGROUND

[0002] Typically, a fifth-generation mobile networks mobile network (5G mobile network) includes a core network and a radio access network (RAN). The core network provides many of the key network functions for the mobile network, including but not limited to, an Access and Mobility Management Function (AMF), an Authentication Server Function (AUSF), a Session Management Function (SMF), a User plane function (UPF), a Unified Data Management (UDM) etc., while the RAN provides a connection between an individual device, e.g. a mobile device, and the data network through a radio link.

[0003] The 5G mobile networks have achieved a series of performance goals including high speed, low latency, and large capacity, thanks to the new architecture of the 5G core network. However, the 5G core network has a higher level of exposure and so is faced with many new security threats pertaining to the network functions such as AMF, AUSF, UDM, etc.

[0004] Conventionally, flow-based-detection (flow-based features are derived from network traffic flow) is adopted for network attack detection. It is understood that the network traffic flow is represented by a collection of packets between the source and the destination nodes. Information about the collection of packets such as a total connection duration, an average number of packets sent etc. can be derived to provide the details about the network connection and enable attack detection. However, obtaining the flow-based features from the collection of packets requires processing effort and could be time-consuming. Therefore, there are some detection solutions that are based on the collected packets directly, i.e. packet-based-detection. The packet-based-detection eliminates the need of calculating flow-based features and thus reduces the complexity and processing time of the attack detection. Furthermore, the packet-based-detection

can identify malicious samples for each individual packet, which provides actionable information on a packet basis and thus helps to improve the network security in a timely manner.

[0005] Meanwhile, Artificial Intelligence (such as Machine Learning, Deep Learning) techniques have been widely adopted to build effective attack detection systems. This is because Artificial intelligence techniques are capable of learning complicated behaviors and use the learned system for discovering and detecting network attacks effectively. Hence, different machine-learning-based attack detection techniques have been developed for securing communication within the 5G core network.

[0006] An example of a packet-based-detection technique using a Machine Learning model has been published in the article "Machine learning 5G Attack Detection in Programmable Logic" by C. Coldwell, D. Conger, E. Goodell, B. Jacobson, B. Petersen, D. Spencer, M. Anderson and M. Sgambati, in the Seventh IEEE Workshop on 5G and Beyond Wireless Security. The article discloses an autoencoder implementation consisting of an encoder which compresses a reshaped 32x32x1 input data into a 8x8x1 square tensor using four Conv2D layers. The 8x8x1 square tensor is flattened and reduced to 3D data using a Dense layer. The decoder reconstructs this 3D data into a 1024x1x1 shape using a Conv2DTranspose layer. The latent spaces are then provided to a trained one-class neural network (OCNN) for attack detection. However, both the autoencoder and the OCNN classifier are fixed structures for detecting any types of network attack in a 5G core network. Such a one-size-fits-all approach may not best suit the user's application and requirements and thus lacks flexibility and accuracy.

[0007] There is therefore a need for a more efficient, accurate, versatile and flexible method for detecting an attack in a 5G core network.

SUMMARY

[0008] According to an aspect of the present disclosure, there is provided a method of detecting an attack in a core network of a mobile communication network. The method includes the steps of extracting a payload content from a packet, obtaining multiple initial detection outcomes using multiple classifiers, each of the multiple initial detection outcomes is obtained based on the payload content by a corresponding one of the multiple classifiers, and determining a final detection outcome associated with the packet based on the multiple initial detection outcomes.

[0009] In some embodiments of the method, obtaining multiple initial detection outcomes includes analyzing the one-dimensional vector by the multiple classifiers in parallel or in sequence.

[0010] In some embodiments of the method, each classifier includes one of a Tree-Based classifier, a Support Vector Machines based classifier, a Neural Network based classifier, a Logistic Regression based classifier, a Linear Discriminant Analysis (LDA) classifier, a Quadratic Discriminant Analysis (QDA) classifier and a Naïve Bayes classifier.

[0011] In some embodiments of the method, the Tree-Based classifier includes one of a Random Forest classifier and a Single Decision Tree classifier.

[0012] In some embodiments of the method, the Neural Network based classifier includes one of a Convolutional Neural Network (CNN) based classifier, a Multilayer-perceptron (MLP) based classifier, a Long-Short Term Memory (LSTM) based classifier, a Generative Adversarial Networks (GAN) based classifier and an Autoencoder based classifier.

[0013] In some embodiments of the method, the final detection outcome is determined via a voting algorithm including one of a plurality voting algorithm, a majority voting algorithm and an ALL-voting algorithm.

[0014] In some embodiments of the method, the final detection outcome is determined based on a secondary machine-learning based classifier.

[0015] In some embodiments of the method, the attack includes at least one of an "AMF Looking For UDM" attack, a "Get All Network Functions" attack, a "Get User Data" attack, a "Random Data Dump" attack, an "Automated Redirect with Timer" attack, a "Fake AMF Insert" attack, a "Random AMF Insert" attack, a "Crash Network Repository Function" attack, a "Fake AMF Delete" attack, and an "Automated Drop with Timer" attack.

[0016] According to another aspect of the present disclosure, there is provided a non-transitory storage medium having stored therein instructions that are executable by at least one processor to perform the method of detecting an attack in a core network of a mobile communication network.

[0017] According to another aspect of the present disclosure, there is provided a server including at least one processor configured to perform the method of detecting an attack in a core network of a mobile communication network.

[0018] According to another aspect of the present disclosure, there is provided a mobile communication system including a server, in which at least one processor configured to perform the method of detecting an attack in a core network of a mobile communication network.

[0019] Other aspects and advantages of the invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention.

BRIEF DESCRIPTION OF DRAWINGS

[0020] The invention will be better understood with reference to the drawings, in which:

Figure 1 is a block diagram showing components of a typical 5G mobile network;

Figure 2 is a flow chart showing a sequence of steps for detecting an attack in a 5G core network according to an embodiment of the invention;

Figure 3 shows two examples of benign payload contents;

Figure 4A shows an example of malicious payload content during a Reconnaissance attack;

Figure 4B shows an example of malicious payload content during a Network Reconfiguration attack;

Figure 4C shows an example of malicious payload content during a Denial-of-Service (DoS) attack;

Figure 5A is a block diagram showing an attack detection framework according to an embodiment of the invention;

Figure 5B is a block diagram showing a detailed example of the attack detection framework of Figure 5A;

Figure 6 is a block diagram illustrating the structure of the Convolutional Neural Network (CNN) used in a Neural Networks block of Figure 5B.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0021] Throughout this document, unless otherwise indicated to the contrary, the terms “comprising”, “consisting of”, “having” and the like, are to be construed as non-exhaustive, or in other words, as meaning “including, but not limited to.”

[0022] Furthermore, throughout the specification, unless the context requires otherwise, the word “include” or variations such as “includes” or “including” will be understood to imply the inclusion of a stated integer or group of integers but not the exclusion of any other integer or group of integers.

[0023] Unless defined otherwise, all technical and scientific terms used herein have the same meaning as is commonly understood by a skilled person to which the subject matter herein belongs.

[0024] As shown in the drawings for the purpose of illustration, the invention may be embodied in a method, a server, and/or a mobile communication system for detecting an attack in a core network of a mobile communication network by examining the payload of a packet. More specifically, the implemented packet-based-detection does not only reduce the processing workload and the complexity of the attack detection system but also provides actionable information timely to avoid further damages to the network. Furthermore, a final attack detection is generated based on multiple initial detection outcomes from multiple individual classifiers via an algorithm. Firstly, the use of multiple individual classifiers can ensure the accuracy of the prediction as well as high system versatility. Secondly, the user is provided with the flexibility to choose the individual classifiers and the algorithm for combining the multiple initial detection outcomes according to the application and the user’s requirements. Thus, an attack detection method with high flexibility is achieved.

[0025] Specifically, Figure 1 shows an example of a typical 5G mobile communication network architecture, which mainly includes a core network (hosted by a Control Plane and a User Plane) and a Radio Access Network (RAN). A person skilled in the art would readily appreciate that the Control Plane includes majority of the 5G core network functions and repositories such as an Access and Mobility Management Function (AMF), a Session Management Function (SMF), an Authentication Server Function (AUSF), a Unified Data Management (UDM), a Network Slice Section Function (NSSF), a NF Registry Function (NRF), a Policy Control Function (PCF) and Unified Data Repository (UDR) while the User Plane includes the User Plane Function (UPF) of the 5G core network.

[0026] A network attack in the 5G core network refers to any network attack aiming to compromise the functions provided by the 5G core network. Network attacks refer to

any attacks targeted at 5G core network, where the attacks may be reflected in the network traffic.

[0027] The method may be used to detect any type of network attacks, which include but not limited to Reconnaissance attack, Network Reconfiguration attack, and Denial-of-Service (DoS) attack. The Reconnaissance attack includes, but is not limited to, impersonating access and mobility management function to request for information on unified data management (“AMF Looking For UDM” attack), maliciously obtaining information on all network functions (“Get All Network Functions” attack), maliciously obtaining user data (“Get User Data” attack), maliciously dumping data (“Random Data Dump” attack), automatically redirecting with timer, (“Automated Redirect with Timer” attack). The Network Reconfiguration attack includes, but is not limited to, fake and malicious access and mobility management function insertion (“Fake AMF Insert” attack and “Random AMF Insert” attack). The DoS attack includes, but is not limited to, crashing network repository functions (“Crash Network Repository Function” attack), accessing and deleting mobility management function to cause service disruption (“Fake AMF Delete” attack), and maliciously dropping user traffic (“Automated Drop with Timer” attack). The attack detection method can be deployed to detect any type of network attacks (in a 5G core network) from any of the entry points as discussed above.

[0028] Figure 2 is a flow chart showing a sequence of steps for detecting an attack in the 5G core network according to an embodiment of the invention.

[0029] S100: OBTAIN PAYLOAD CONTENT

[0030] The sequence starts with an “OBTAIN PAYLOAD CONTENT” step (S100). A person skilled in the art would readily understand that a command or information data specified in the application layer of a packet transmitted in the 5G core network forms payload content of the packet. When a packet is received by an attack detection framework implemented either on an independent server or an integrated server hosting the network function(s) of a 5G core network, the payload content of the packet is extracted for further processing. A person skilled in the art would also understand that the received packet can contain either benign payload content or malicious payload content.

[0031] Figure 3 shows two examples of benign payload content while Figures 4A-4C show malicious payload content during different network attacks in the 5G core network.

Figure 4A shows an example of malicious payload content during a “Get User Data” attack under the category of Reconnaissance attack. The purpose of this attack is to obtain unauthorized information from the UDM regarding a user with a random ID. Figure 4B shows an example of malicious payload content during a “Fake AMF Insert” attack under the category of Network Reconfiguration attack. This attack registers a fake AMF with the NRF by a “PUT” a JSON object into the NRF. Figure 4C shows an example of malicious payload content during an “Automated Drop with Timer” attack under the category of Denial-of-Service (DoS) attack. This attack code listens for a packet forwarding control protocol session establishment request and checks if the User Equipment (UE) address is in a list of victim address subsequently. If the UE address is listed, the UE will be disconnected from the data network.

[0032] S200: CONVERT PAYLOAD CONTENT TO 1-D VECTOR

[0033] The sequence next proceeds to a “CONVERT PAYLOAD CONTENT TO 1-D VECTOR” step (S200). After being extracted from the received packet, the payload content is preferably converted to a one-dimensional (1-D) vector of a pre-determined size. In one embodiment, the payload content is converted to a 1024-byte sized vector. The conversion can be performed by either padding the payload content with 0s or truncating the payload content depending on the actual size of the payload content. A person skilled in the art would appreciate that the truncation of the payload content will increase the attack detection speed while still fully retaining the malicious information since most malicious payload content is shorter than 1024 bytes. ~~Furthermore, truncating the payload content (instead of the received packet) would keep more payload information for the attack detection.~~ Moreover, converting the payload content to a vector of a single size requires a small amount of processing effort at the initial stage but can save the data processing effort at the multiple individual classifiers.

[0034] S300: OBTAIN MULTIPLE INITIAL DETECTION OUTCOMES

[0035] After the payload content is converted to a 1024-Byte sized vector, the sequence proceeds to a “OBTAIN MULTIPLE INITIAL DETECTION OUTCOMES” step (S300). As shown in Figure 5A, the 1024-Byte vector (210A) is input into a plurality of individual classifiers 220A-1, 220A-2 220A-N. The user can configure the number of classifiers N depending on the application and the user’s requirements. The use of

multiple classifiers instead of a single classifier helps to improve the accuracy of the attack detection result.

[0036] Furthermore, the user is provided with the flexibility of selecting the multiple classifiers from a wide range of classifiers, including but not limited to, a Tree-Based classifier (including Random Forest classifier and Single Decision Tree classifier), a Support Vector Machines based classifier, a Neural Network based classifier, a Logistic Regression based classifier, a Linear Discriminant Analysis (LDA) classifier, a Quadratic Discriminant Analysis (QDA) classifier and a Naïve Bayes classifier. In one embodiment, the Neural Network based classifier can be any of a Convolutional Neural Network (CNN) based classifier, a Multilayer-perceptron (MLP) based classifier, a Long-Short Term Memory (LSTM) based classifier, a Generative Adversarial Networks (GAN) based classifier and an Autoencoder based classifier. As each of these classifiers has its own advantages and limitations, such flexibility would best suit the user's application and requirements. In one embodiment, a combination of classifiers with different properties is preferred to achieve high system versatility. In another embodiment, where processing time of the individual classifiers is an important factor to be considered, classifiers requiring shorter processing time may be selected. Classifiers with minimal testing time or requiring fewer computing resources may be selected as per requirements to achieve an efficient attack detection. A person skilled in the art would understand that the selected classifiers are trained using a sample dataset before being implemented for attack detection. The training can be conducted with a supervised approach, an unsupervised approach, or a semi-supervised approach.

[0037] After receiving the input of a 1024-Byte sized vector from 210A, each of the trained classifiers examines the input vector and outputs an initial detection outcome. In one embodiment, the initial detection outcome from each classifier is a "True"/"False" indicator (True: Attack or Malicious Payload is detected; False: no attack or Benign Payload is detected). In another embodiment, the initial detection outcome from each classifier is a prediction probability indicating a possibility of the 1-D vector under examination being an attack or a malicious payload.

[0038] S400: DETERMINE A FINAL DETECTION OUTCOME

[0039] After the initial detection outcomes are obtained from the individual classifiers, the sequence proceeds to a "DETERMINE A FINAL DETECTION OUTCOME" step

(S400). The initial detection outcomes from the multiple individual classifiers will be combined by an algorithm (230A of Figure 5A) to generate a final detection outcome. Different algorithms can be applied at 230A depending on the user requirements and preference. When the initial detection outcomes from the individual classifiers are "True"/"False" indicators, a voting algorithm may be used as the algorithm. The voting algorithm can be implemented as a majority voting algorithm, a plurality voting algorithm, an ALL-voting algorithm, or the like. The final detection outcome based on the voting algorithm is preferably a "True"/"False" indicator.

[0040] When the initial detection outcomes are prediction probabilities, a secondary machine-learning based classifier can be implemented as the algorithm. The secondary classifier takes the prediction probabilities from the individual classifiers as the input features and outputs a final label as the final detection outcome. In one embodiment, the algorithm can be implemented as an average probabilistic computation or a weighted average probabilistic computation. The final label can be in the form of a "True"/"False" indicator or a final prediction probability. In one embodiment, a final prediction probability above a pre-determined threshold indicates a network attack or a malicious payload associated with the packet under examination.

[0041] A person skilled in the art would appreciate that a voting algorithm is easy to implement and requires less processing effort compared to the machine-learning based secondary classifier. However, the machine-learning based secondary classifier can generate a prediction probability as the final detection outcome. The user can then make further decisions based on the probability value according to his/her own preference or requirement. The user has the flexibility to determine the algorithm based on the actual needs, and thus obtain an attack detection method/system that best addresses the user's application and requirements.

[0042] The final detection outcome is subsequently presented to the user at 240A for any further action if necessary. In one embodiment, the final detection outcome is pushed to the user once the attack detection for a packet is completed regardless of whether the outcome is attack or benign. A person skilled in the art would appreciate that to push the outcome on a packet basis in a timely manner would prevent the spread of a malicious packet and with appropriate action taken can reduce/avoid further damage to the network. In another embodiment, the final detection outcome will only be pushed to the user when an attack is detected. For a packet being determined as a benign

packet, the final detection outcome will be discarded to reduce the communication workload.

[0043] One detailed example of the attack detection method is illustrated in Figure 5B. In this example, the user selects three types of classifiers, namely, a Tree-Based model (220B-1), a Support Vector Machines (220B-2) and a Neural Network (220B-3). These three classifiers offer different properties and thus form a versatile attack detection method/system. Before being used for attack detection, the three classifiers are trained based on both samples with benign payload and samples with malicious payload using a supervised learning approach. A person skilled in the art would readily understand, under the supervised learning approach, a training data sample is labeled data (e.g., a predetermination is made as to whether the data includes malware or is benign). Table 1 provides the details of the sample dataset used in the classifier training while Table 2 provides details of the classifier training.

TABLE 1: Sample Dataset for Training Classifiers

Sample Type	Main Dataset	Training	Test
Benign Payload	24174	19339	4835
Malicious Payload	24174	19339	4835
Total	48348	38678	9580

TABLE 2: Training Details for Classifiers

Model Name	Training Parameters	TrainingTime per sample (ms)	TestingTime per sample (ms)
Tree-based Model (Random Forest)	1.Number of trees = 100 2. Criterion for quality of split= Gini Impurity	0.2	0.01
Support Vector Machine	1. Kernel = Radial Basis Function (RBF)	2.8	0.35
Neural Network (CNN)	1. Learning Rate=0.001, 2. Epochs=10, 3. BatchSize=32 4. Embedding Layer Dimension = 32 5. Number of Convolution-pooling layer combinations = 3	0.5	0.12

[0044] In one embodiment, the Tree-Based model (220B-1) is implemented as a Random Forest model. The Random Forest model is a classifier that contains a number of decision trees (100 decision trees in this particular example). Decision trees are created based on multiple data subsets which are generated by picking vectors randomly from the given dataset (210B) with replacement. Preferably, Gini Impurity is used as a criterion to determine how the features of a dataset should split nodes to form the tree. Instead of relying on one decision tree, the random forest computes a final prediction result based on the prediction result from each of the decisions trees via a voting technique (e.g. majority voting).

[0045] At 220B-2, one of the classifiers is implemented using Support Vector Machines (SVM). The objective of the Support Vector Machines is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. Thus, a Support Vector Machine constructs a set of hyperplanes in a high or infinite-dimensional space, which can be used for classification. In this example, a non-linear classification using a kernel function (e.g. Radial Basis Function (RBF)) is implemented to map the inputs from 210B into the high-dimensional feature spaces.

[0046] At 220B-3, a CNN classifier is designed and trained according to the parameters in Table 2 as follows:

- 1) Construct embedding layer to reduce dimension of input vectors;
- 2) Construct and configure convolutional layer (e.g. kernel size, kernel count, stride, activation function) to extract features from the input vectors;
- 3) Construct pooling layer to generate abstract significant features for reduced computational resource requirements;
- 4) Repeat 3) and 4) for m round(s);
- 5) Combine each convolution layer and corresponding pooling layer to form the m -depth neural network;
- 6) Perform flattening (if needed) or global averaging of the output of the m -depth neural network to convert the data dimension for input to the classification layer;
- 7) Construct the fully connected classification layer;
- 8) Combine the m -depth neural network, flattening or global average pooling process and the fully connected classification layer.

[0047] After being trained, the classifiers are tested using the testing data samples. A person skilled in the art would readily understand that a testing data sample is

unlabeled data (e.g., there is no predetermination made as to whether the data is malicious or benign) used to evaluate the accuracy of the trained/learned classifiers. Table 2 shows that different classifiers require different test times. As the SVM classifier is considerable slower than the rest of the classifiers, it could be replaced by another classifier to increase the attack detection speed. Furthermore, to run the individual classifiers in parallel as shown in Figure 5B would further improve the attack detection speed. In another embodiment, the individual classifiers can be configured in sequence depending on the user's requirement.

[0048] After receiving the input 1024-Byte sized vector from 210B of Figure 5B, each of the trained classifiers (220B-1, 220B-2 and 220B-3) will analyze the input and generate an initial detection outcome. In this example, the initial detection outcome is in the form a "True/False" indicator illustrated as "Attack Detection Vote #1", "Attack Detection Vote #2" and "Attack Detection Vote #3" in Figure 5B.

[0049] A voting algorithm is implemented as the algorithm to combine the initial detection outcomes at 230B. In this example, the classifier 220B-1, the classifier 220B-2 and the classifier 220B-3 predict the given input vector as a malicious payload, a malicious payload and a benign payload, respectively. The corresponding initial detection outcomes are True, True and False respectively. If a majority/plurality voting algorithm is implemented, the input vector will be classified as a malicious payload. However, if an ALL-voting algorithm is implemented, the same input vector will be classified as a benign payload since the classifier 220B-3 does not predict it to be a malicious payload.

[0050] The final detection outcome of a "Yes" indicator (if the majority/plurality voting algorithm is implemented) is presented to the user at 240B. In one embodiment, actions such as quarantining the packet, raising an alarm etc. can be taken by the user to prevent further damage to the network.

[0051] In one embodiment, the attack detection method can be deployed as a supplementary network function within the Control Plane of a 5G core network similar to the Network Data Analytics Function (NWDAF) introduced in TS 29.520 of 3GPP (3GPP. 5G System; Network Data Analytics Services; Stage 3. Technical Specification (TS) 29.520, 3rd Generation Partnership Project (3GPP), 2023. Version 18.1.0.). In one embodiment, the supplementary network function is hosted by an independent server.

In another embodiment, the supplementary network function is integrated within a server hosting the network functions of the 5G core network.

[0052] Another embodiment of the current invention is directed to a server being configured to detect an attack in a core network of a mobile communication network. The server includes at least one processor, which is configured to extract payload content from a received packet, obtain multiple initial detection outcomes using multiple classifiers and determine a final detection outcome based on the multiple initial detection outcomes. In one embodiment, the server is an independent server which works collaboratively with other servers (hosting the network function(s) of a 5G core network) for attack detection. In another embodiment, the server is an integrated server hosting both the network function(s) of the 5G core network and the attack detection function.

[0053] Another embodiment of the current invention is directed to a mobile communication system. The mobile communication system at least includes a 5G core network, a RAN and a server as described in the above paragraph.

[0054] As an example, the method is described in the context of a 5G mobile network. However, the method may also be implemented to detect an attack in a core network of any other mobile communication networks.

[0055] It should be further appreciated by the person skilled in the art that one or more of the above modifications or improvements, not being mutually exclusive, may be further combined to form yet further embodiments of the present invention.

CLAIMS

[Claim 1] A method of detecting an attack in a core network of a mobile communication network, comprising:

- extracting payload content from a packet;
- obtaining multiple initial detection outcomes using multiple classifiers, each of the multiple initial detection outcomes is obtained based on the payload content by a corresponding one of the multiple classifiers; and
- determining a final detection outcome associated with the packet based on the multiple initial detection outcomes.

[Claim 2] The method according to Claim 1 further comprises converting the payload content to a one-dimensional vector of a pre-defined size for input into the multiple classifiers.

[Claim 3] The method according to Claim 2, wherein obtaining multiple initial detection outcomes comprises analyzing the one-dimensional vector by the multiple classifiers based on at least one of parallel processing and sequential processing.

[Claim 4] The method according to Claim 1, wherein each classifier comprises one of a Tree-Based classifier, a Support Vector Machines based classifier, a Neural Network based classifier, a Logistic Regression based classifier, a Linear Discriminant Analysis (LDA) classifier, a Quadratic Discriminant Analysis (QDA) classifier and a Naïve Bayes classifier.

[Claim 5] The method according to Claim 4, wherein the Tree-Based classifier comprises one of a Random Forest classifier and a Single Decision Tree classifier.

[Claim 6] The method according to Claim 4, wherein the Neural Network based classifier comprises one of a Convolutional Neural Network (CNN) based classifier, a Multilayer-perceptron (MLP) based classifier, a Long-Short Term Memory (LSTM) based classifier, a Generative Adversarial Networks (GAN) based classifier and an Autoencoder based classifier.

[Claim 7] The method according to Claim 1, wherein the final detection outcome is determined via a voting algorithm comprising one of a plurality voting algorithm, a majority voting algorithm and an ALL-voting algorithm.

[Claim 8] The method according to Claim 1, wherein the final detection outcome is determined based on a secondary machine-learning based classifier.

[Claim 9] The method according to Claim 1, wherein the attack comprises at least one of an “AMF Looking For UDM” attack, a “Get All Network Functions” attack, a “Get User Data” attack, a “Random Data Dump” attack, an “Automated Redirect with Timer” attack, a “Fake AMF Insert” attack, a “Random AMF Insert” attack, a “Crash Network Repository Function” attack, a “Fake AMF Delete” attack, and an “Automated Drop with Timer” attack.

[Claim 10] A server comprising at least one processor configured to perform the method as defined in any one of Claims 1 to 9.

[Claim 11] A mobile communication system, comprising:

a server, in which at least one processor being configured to perform the method as defined in any one of Claims 1 to 9.

[Claim 12] A non-transitory storage medium having stored therein instructions that are executable by at least one processor to perform the method as defined in any one of Claims 1 to 9.