(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 May 2020 (14.05.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/094276 A1

(51) International Patent Classification:
H04L 29/06 (2006.01) *G06F 21/55* (2013.01)
H04W 12/12 (2009.01) *G06F 21/56* (2013.01)

(21) International Application Number:
PCT/EP2019/074274

(22) International Filing Date:
11 September 2019 (11.09.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/757,769 09 November 2018 (09.11.2018) US

(71) Applicants: **NEC LABORATORIES EUROPE GMBH** [DE/DE]; Kurfürsten-Anlage 36, 69115 Heidelberg (DE). **UNIVERSIDAD DE MURCIA** [ES/ES]; Edificio Rector Soler, 1a Planta, Campus de Espinardo (Murcia), 30071 Murcia (ES).

(72) Inventors: **PAPAMARTZIVANOS, Dimitrios**; Agiou Dimitriou 5, 63081 Neos Marmaras (Chalkidiki) (GR). **BIFULCO, Roberto**; Bergheimerstraße 38, 69115 Heidelberg (DE). **KAMBOURAKIS, Georgios**; 4 Faethonos St., 83100 Samos (GR). **GÓMEZ MÁRMOL, Felix**; Universidad de Murcia, Faculty of Computer Science, 30100 Murcia (ES).

(74) Agent: **ULLRICH & NAUMANN**; Schneidmühlstraße 21, 69115 Heidelberg (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,

(54) Title: METHOD AND SYSTEM FOR ADAPTIVE NETWORK INTRUSION DETECTION

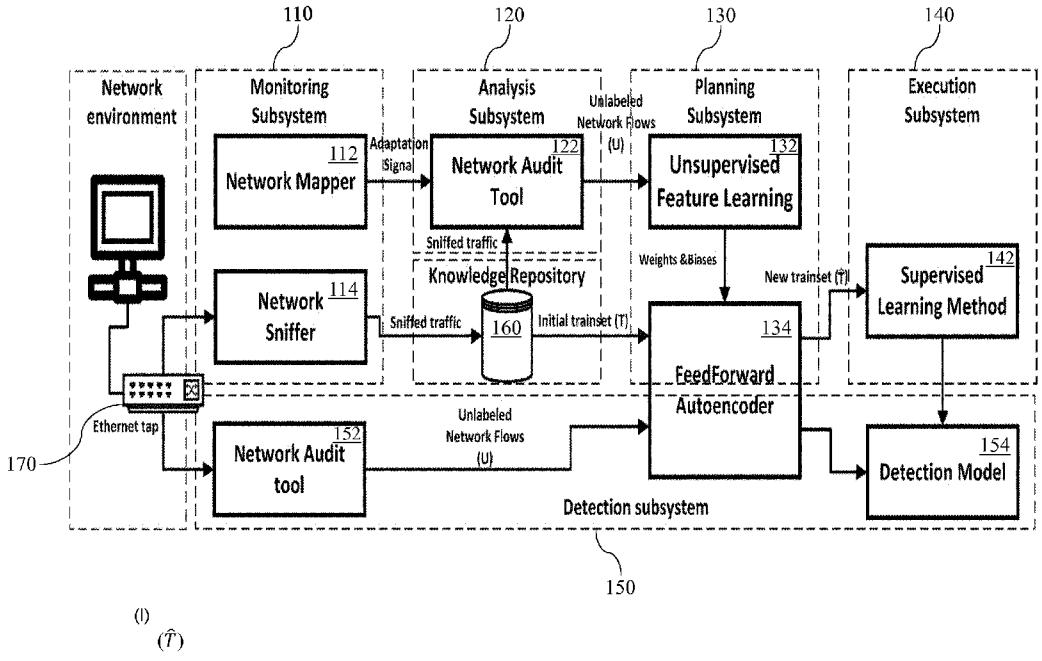


FIG. 1

(57) Abstract: The present invention provides a method for adaptive network intrusion detection that includes: a) deploying a network traffic capture system and collecting network packet traces; b) using a network audit tool (122) to extract features from the collected network packet traces; c) feeding the extracted features as unlabeled data (U) into a representation function and utilizing the representation function as an unsupervised feature learning algorithm (132) to learn a new representation of the unlabeled data (U); d) providing a labeled training set (T) capturing examples of malicious network traffic and using the learned new representation of the unlabeled data (U) to modify the labeled training set (T) to obtain a new training set (formula (I)); and e) using the new training set (formula (I)) to train a traffic classification machine learning model. Furthermore, the present invention provides a respective system for adaptive network intrusion detection.

[Continued on next page]

WO 2020/094276 A1

HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

METHOD AND SYSTEM FOR ADAPTIVE NETWORK INTRUSION DETECTION

FIELD

[0001] The present invention relates to a method and system for adaptive network intrusion detection.

BACKGROUND

[0002] Intrusion detection systems (IDSs) are one of the most important entities when it comes to information and communications technology (ICT) infrastructure protection against cyberattacks. IDSs weaponize defenders with fundamental means to detect offensive events and consequently trigger optimal counteraction plans against them. Because new attacks continue to emerge, the industry needs new methods that are able to adapt rapidly to the changes in the field.

[0003] In principle, misuse detection systems are the most widely deployed kind of intrusion detection systems. Misuse IDSs rely on known signatures trying to designate network traffic instances to legitimate or attack traffic classes. This kind of IDS lacks the ability of identifying new attack patterns or deviations from known ones, and their performance depends on the freshness of the signatures database. Hence, the IDS's administrator needs to put significant effort to keep the misuse detection model up to date. Because the protected environment may be a dynamic ecosystem where new devices and/or services may appear or leave the network at any moment (e.g., the Internet of Things (IoT)), adaptability can become a burden for system administration.

[0004] Keeping any type of IDS up-to-date is a demanding task for several reasons, for example, due to issues pertaining to environmental changes. Environmental changes refers to any aspect of a network that can change and consequently affect the profile of the generated network traffic. In practice, the addition (or disengagement) of a device in a network can affect different network aspects, including the topology, the running services, the open ports, the communication protocols and/or applications, the network traffic load, and others. In turn, these environmental changes affect fundamental security features such as the vulnerabilities appearing in the network, which can generate multiple penetration paths for the attackers. Considering a more dynamic network like an IoT environment, an Ad Hoc network, or even a corporate network with a Bring your own device (BYOD) policy applied, one can understand

that the attack surface of the network can be increased unexpectedly. It is plausible that, the newly introduced device might be already infected by a malware and act as a stepping stone for an attacker to conquer more assets within the network. Yet, new devices are not the only enemies of an IDS in a network, as also already installed devices will eventually proceed with software/OS updates or new software installations that again will bring in alterations in the environment.

[0005] Overall, the above mentioned changes are routine actions that constantly appear in every common network, rather than unusual events. In practice all sorts of modifications can significantly affect the performance of an IDS, which is placed to protect an ever-changing infrastructure. This reality, combined with the lack of adaptable detection engines, forces a legacy IDS to become quickly outdated and inadequate as it inevitably has to operate in new and "unknown" environments for which its engine was not trained to do so. Thus, security administrators undertake the task of constantly retraining the IDS by considering all the new environmental changes to regain the reliability and the performance of the detection system. All in all, the cardinal challenge for any IDS designer, i.e., find proper ways to automatize a retrain process, remains largely unsolved.

[0006] The following references provide further background relevant to the present invention, and each of which are hereby incorporated by reference herein in their entirety: R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," Proceedings of the 24th International Conference on Machine Learning, ICML '07, 759-766 (2007) ("Raina"); J. O. Kephart, D. M. Chess, "The vision of autonomic computing," Computer 36 (1), 41-50 (2003) ("Kephart"); W. Lee, S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," ACM Trans. Inf. Syst. Secur. 3 (4), 227-261 (2000) ("Lee"); and M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications, CISDA'09, 53-58 (2009) ("Tavallaee").

SUMMARY

[0007] An embodiment of the present invention provides a method for adaptive network intrusion detection that includes: a) deploying a network traffic capture system and collecting network packet traces; b) using a network audit tool to extract features from the

collected network packet traces; c) feeding the extracted features as unlabeled data into a representation function and utilizing the representation function as an unsupervised feature learning algorithm to learn a new representation of the unlabeled data; d) providing a labeled training set capturing examples of malicious network traffic and using the learned new representation of the unlabeled data to modify the labeled training set to obtain a new training set; and e) using the new training set to train a traffic classification machine learning model. Embodiments of the method may further include: f) deploying the traffic classification machine learning model for examining live traffic. Before feeding the live traffic to the traffic classification machine learning model the live traffic's extracted features may be in embodiments of the method modified using the representation function. In embodiments, operations c)-e) can be repeated periodically to adapt to network traffic changes.

[0008] Furthermore, an embodiment of the present invention provides a system for adaptive network intrusion detection, wherein the system comprises a network traffic capture system configured to collect network packet traces; a network audit tool configured to extract features from the collected network packet traces; a representation function configured to receive from the network audit tool the extracted features as unlabeled data, to execute an unsupervised feature learning algorithm to learn a new representation of the unlabeled data, to receive a labeled training set capturing examples of malicious network traffic, and to use the learned new representation of the unlabeled data to modify the labeled training set to obtain a new training set; and a traffic classification machine leaning model configured to be trained using the modified training set.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present invention will be described in even greater detail below based on the exemplary figures. The invention is not limited to the exemplary embodiments. All features described and/or illustrated herein can be used alone or combined in different combinations in embodiments of the invention. The features and advantages of various embodiments of the present invention will become apparent by reading the following detailed description with reference to the attached drawings which illustrate the following:

[0010] FIG. 1 shows an overview of a system for adaptive network intrusion detection in accordance with an embodiment of the present invention;

[0011] FIG. 2 illustrates deviation of IDS accuracy over 100 consecutive environmental states;

[0012] FIG. 3 illustrates deviation of IDS attack detection ratio over 100 consecutive environmental states;

[0013] FIG. 4 illustrates a performance comparison over all the average metrics; and

[0014] FIG. 5 illustrates a processing system for implementing an embodiment of the present invention.

DETAILED DESCRIPTION

[0015] The present invention provides a technique to improve the detection accuracy of Network Intrusion Detection Systems when network conditions and attacks change, by employing periodic learning of feature maps via autoencoders. For example, an embodiment provides a method and a system for self-adaptive and autonomous IDSs that addresses the above-mentioned inherent limitations of state of the art intrusion detection systems.

[0016] Additionally, embodiments of the present invention address the following problems: legacy misuse IDSs lack the ability of identifying new attack patterns or deviations from known ones; legacy misuse IDSs become quickly outdated and inadequate as they inevitably have to operate in new and unknown environments, whose engine was not trained to do so; keeping a legacy misuse IDS up-to-date is a demanding engineering task as security administrators need to manually investigate for new and unknown offensive network incidents, label them, and then retrain the detection engine; the dynamic nature of the state-of-the art networks brings in network environmental changes that render the legacy IDSs ineffective; and proper methods to automatize the retraining process of a misuse IDS remain unsolved.

[0017] In the context of the present invention, numerous types of events that lead the network into a new state, and thus affect the IDS's operational environment are perceived. Such changes also affect the network's behavioral profile, which in turn is reflected in the network flows. According to RFC 2722 [12], a network flow can be seen as an artificial logical equivalent to a call or connection, which has as attribute values aggregated quantities which reflect the events that take place during this connection. These attribute values can bear

valuable information regarding numerous aspects of the network's behavior ranging from the topology to the workload and the active services. Thus, network flows are a rich source of information that can improve the network security visibility as they can be leveraged by security analysts to identify and assess hostile actions, new attacks, and the network's security state in general. As a result, when a network is overwhelmed by unknown and previously unseen network flows, an IDS which has been trained to defend a network based on a static training set needs to be retrained in order to sustain a credible security level. This however implies the need of a demanding process on behalf of the security analyst to identify and label manually new network instances for creating a new dataset that can be used to retrain the IDS. Considering that most of the network changes are common actions that can happen regularly, it becomes clear that there is a need for methods capable of automating the retraining process.

[0018] To this end, embodiments of the present invention aim to offer an automated way to keep the detection ratio of a misuse IDS to acceptable levels regardless of the environmental changes that may indicate the presence of previously unknown attacks. Embodiments of the invention can empower autonomous and self-adaptive misuse IDSs by enabling them to adapt to their environment and significantly contribute in keeping a high or at least acceptable security level. This quality also significantly alleviates security experts from the demanding task of retraining the IDS. Unlike the current state-of-the-art IDSs, which do not use self-adaptive and autonomous methods to automatize the retraining process of a misused IDS, embodiments of the present invention overcome limitations of the state-of-the-art IDSs and provide methods for automatically adapting IDSs without the constant need to manually refresh a training set and retrain.

[0019] An embodiment of the present invention provides a method for adaptive network intrusion detection that includes learning of feature mapping function's parameters from unlabeled network traffic samples and usage of such parameters to transform the feature of a labeled data set's samples, which are then used to autonomously re-train an intrusion detection classifier.

[0020] An embodiment of the present invention provides a method for adaptive network intrusion detection that includes the following operations:

1) Deploy a network traffic capture system and collect network packet traces;

2) Use a network audit tool for extracting features from the collected packet traces;
3) Use the extracted features to train a representation function using machine learning (e.g., by using an autoencoder);
4) Use the trained representation function to modify a labeled training set that captures examples of malicious network traffic;
5) Use the modified training set to train a machine learning algorithm for traffic classification (traffic classification model);
6) Deploy the traffic classification model for examining live traffic. Before feeding the live traffic to the model, the live traffic's extracted features may be modified using the same the learned representation function used to modify the training set; and/or
7) Repeat the points 3-7 periodically to adapt to network traffic changes.

[0021] Embodiments of the present invention are able to exploit unlabeled data $U = \left\{x_u^{(1)}, x_u^{(2)}, \ldots, x_u^{(k)}\right\}$, which can: 1) be of any class and not necessarily to coincide with the classes of the labeled data T, and 2) be drawn from a different distribution from the labeled data T. In an unknown network environment, an IDS will face both known and unknown attacks, which both stem from different distributions. Thus, embodiments of the present invention are able to uncover new attack patterns or deviations from known ones.

[0022] Embodiments of the present invention alleviate the burden of retraining an IDS every time a change appears in its environment. This can eliminate burdens for the administrator, because the retraining process can require significant effort to assign labels by hand to large-scale data such as network data. Embodiments of the present invention significantly extend the autonomy of an IDS. For example, embodiments enable minimizing's significantly the engagement of the security administrator in the maintenance of the IDS as it is a self-adaptive and autonomous approach.

[0023] Embodiments of the present invention, unlike the state-of-the-art, keep the attack detection ratio to high levels in situations where normally the security administrators would be forced to manually retrain and reset the IDS. In fact, by implementing the present invention the need for human intervention may not be eliminated completely, that is, the administrator should initialize the system and configure its parameters accordingly.

[0024] An embodiment of the present invention uses principles described by the MAPE-K reference model (see, e.g., Kephart) to build autonomous and self-adaptive systems, while

it utilizes Self-taught learning (STL) (see, e.g., Raina) for grasping network traffic dynamics based on generalized features reconstructions stemming directly from the unknown network environment and its unlabeled data. Embodiments enable self-adaptation and autonomic computing in an IDS system by taking advantage of transfer learning from unlabeled data via a systemic method.

[0025] Self-Taught Learning (STL) is a machine learning framework that is able to exploit unlabeled data with the purpose of improving a supervised classification problem. In the STL concept, both labeled and unlabeled data are provided.

[0026] The labeled data are used as the initial training set of m samples for a given classification task $T = \left\{\left(x_l^{(1)}, y^{(1)}\right), \left(x_l^{(2)}, y^{(2)}\right), \ldots, \left(x_l^{(m)}, y^{(m)}\right)\right\}$, where $x_l^{(i)} \in \mathbb{R}^n$ is the i-th sample with n features, $y^{(i)} \in \{1, \ldots, C\}$ is the class label, and the l symbol stands for "labeled".

[0027] The set of k unlabeled samples $U = \left\{x_u^{(1)}, x_u^{(2)}, \ldots, x_u^{(k)}\right\}$, where $x_u^{(i)} \in \mathbb{R}^n$ is the i-th unlabeled sample with n features, and u stands for "unlabeled". U is given as input to an unsupervised learning method to learn a higher level structure of those data. This structure is then used as a base to transform the initial labeled dataset T and obtain a new training set $\hat{T} = \left\{\left(a_l^{(1)}, y^{(1)}\right), \left(a_l^{(2)}, y^{(2)}\right), \ldots, \left(a_l^{(m)}, y^{(m)}\right)\right\}$, where $a_l^{(i)} \in \mathbb{R}^v$ represents the i-th new training example. In consequence, the new training dataset $\hat{T}$ can be used to train a supervised learning method.

[0028] Embodiments of the present invention take advantage of the beneficial characteristics of STL to provide a broader and holistic system for self-adaptive and automatic misuse IDSs. More specifically, the latter is trained based on an initial basic labelled training set T. However, due to the environmental network changes, the IDS inevitably will face performance issues. That is, by exploiting the unknown and unlabeled traffic U of the network, an embodiment of the present invention, through the cooperative operation of its diverse subsystems, is able to revitalize autonomously the initial dataset T and generate a new training set $\hat{T}$, which can be used to "on-the-fly" retrain the IDS engine and sustain its detection ability to high levels.

[0029] FIG. 1 illustrates a system according to an embodiment, which, in turn, is made up of five subsystems following the principles of a MAPE-K method, while the interconnections among the subsystems are annotated with the exchanged information. According to embodiments of the invention, the benefits of MAPE-K and STL co-work toward coping with the challenges of building a solid basis for misuse adaptive IDSs. Generally, MAPE-K is a reference model to build autonomous and self-adaptive systems, wherein the model comprises 5 activities that operate over a Domain Specific System (DSS) and a Context. In case of the present invention, the DSS is the IDS per se, while the Context can be adjusted to any given type of network where there is a need of an adaptive IDS. The cardinal operation of capturing any new characteristics of unknown traffic and the autonomous generation of the new training set is undertaken by the planning subsystem. The latter enables adaptive intrusion detection.

[0030] Monitoring subsystem: According to embodiments of the invention, the monitoring subsystem 110 is configured as a network traffic capture system that collects network packet traces. Specifically, the monitoring subsystem 110 undertakes the task of coordinating the sensors for acquiring the basic knowledge that will reveal the need of IDS adaptation. Network mappers 112 can be used as the basic sensors for network inventory. Such entities are able to determine various characteristics of the network including its topology, the available hosts, the running services, open ports, the operating systems, and even potential vulnerabilities. By collecting such information, the Monitoring subsystem 110 is able to determine any alteration event that requires an IDS adaptation. The monitoring activity is able to determine the environmental changes in collaboration with the Knowledge activity, which serves as a repository 160 for reference purposes. The Monitoring subsystem 110 can schedule the network mapping process to occur periodically according to the characteristics of the network.

[0031] In parallel, another sensor type which is controlled by the Monitoring subsystem 110 is the Network Sniffers 114. The latter are used to capture the network traffic through the Ethernet tap 170. The captured traffic is stored in the Knowledge repository 160. This traffic is used as the basis to extract in a later stage the network flows which have to pass through the detection engine/model 154 of the IDS. Additionally, the network traffic is stored in the repository 160 to serve the purpose of adaptation as it is described further down in the Planning/Execution subsystems 130/140.

[0032] Analysis subsystem: According to embodiments of the invention, the analysis subsystem 120 comprises network audit tools 122 to extract features from network packet traces connected by the network traffic capture system, e.g. the monitoring subsystem 110. Specifically, after collecting the necessary data, the Analysis subsystem 120 performs the transformation of the raw network traffic into network flows. By using the stored traffic of the repository component 160, the Analysis subsystem 120 utilizes network audit tools 122 such as Argus (cf. Argus. The Network Audit Record Generation and Utilization System. Accessed: Jan. 23, 2019. [Online]. Available: https://qosient.com/argus/index.shtml) or CICFlowMeter (cf. CICFlowMeter. UNB CIC Network Traffic Flow Generator (Formerly SCXflowmeter). Accessed: Jan. 23, 2019. [Online]. Available: http://www.unb.ca/cic/datasets/flowmeter.html) in order to generate the network flows. These tools are able to analyze large amounts of network traffic even in an in-line manner and process them accordingly to generate highly informative network flows with various features. These features include the machine learning features of the network traffic instances which are fed into the IDS engine 154 for detection purposes. These flows constitute the unlabeled dataset U, which are given into the supervised model of the IDS to detect potential attacks, and are used as the unlabeled data fed to the Planning subsystem 130 to fuel the adaptive process. Consequently, according to embodiments of the invention, during the IDS operation, the adaptive process may be simultaneously executed with the aim of coming up with a new detection model 154 that will replace the existing one.

[0033] Planning subsystem: The Planning subsystem 130 undertakes the process of leveraging the unlabeled data for initiating the adaptive process. Until that point, the Monitoring subsystem 110 and the Analysis subsystem 120 identified environmental changes in the network, the Knowledge repository 160 consolidated the network flows, which were generated by the time that the change(s) occurred. This moment is the beginning of a time interval when the IDS may face unknown network instances that can undermine its performance. In this direction, the Planning subsystem 130 is configured to cope with this ambiguity by utilizing unsupervised feature learning methods. An embodiment of the present invention utilizes, for example, Sparse Autoencoders as the unsupervised learning method 132 to learn informative and sparse new representations ($\hat{T} = \left\{\left(a_l^{(1)}, y^{(1)}\right), \left(a_l^{(2)}, y^{(2)}\right), \ldots, \left(a_l^{(m)}, y^{(m)}\right)\right\}$) of the unlabeled data ($U = \left\{x_u^{(1)}, x_u^{(2)}, \ldots, x_u^{(k)}\right\}$) and thus benefit the supervised task of the misuse IDS.

[0034] Unsupervised feature learning through Sparse Autoencoders

[0035] An autoencoder is a neural network that applies backpropagation and aims to reconstruct a given input to an output that approximately resembles to the initial input. That is, the neural network, given an input x, attempts to learn a function $h_{W,b}(x) \approx x$, where W, b vectors denote the weights and biases among the layers and their units of the neural network. This process can be driven also by other objectives apart from minimizing solely the reconstruction error. Embodiments of the present invention utilize a sparse autoencoder in order to learn sparse representations of the input data. For example, an embodiment utilizes a sparse autoencoder of three layers ($n_l = 3$). The backpropagation process can be driven by the following cost function (Equation 1):

$$J(W,b) = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{1}{2} \parallel x^{(i)} - \hat{x}^{(i)} \parallel_2^2\right) + \frac{\lambda}{2}\sum_{l=1}^{2}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2 + \beta \sum_{j=1}^{s_2} KL(\rho||\hat{\rho}) \qquad (1)$$

where:

$x^{(i)} \in \mathbb{R}^n$ is the i-th input of an unlabeled example;

$\hat{x}^{(i)} \in \mathbb{R}^n$ is the i-th output given the i-th input;

k is the number of the examples in the unlabeled training set;

λ is the weight decay parameter;

l index denotes the number of a layer;

s_l is the number of nodes in the l layer;

β is the weight of the sparsity penalty; and

$\parallel x^{(i)} - \hat{x}^{(i)} \parallel_2^2$ is the squared L^2 norm.

[0036] Through backpropagation, the sparse autoencoder aims to minimize the cost function (1). As can be seen, the cost function includes three terms. The first term represents the average accumulated squared error among the input and the output terms of the network. Thus, by using the first term the network tries to reconstruct the output and achieve high similarity with the input. The output terms derive as follows: $\hat{x}^{(i)} = h_{W,b}\left(x^{(i)}\right) =$

$f\left(\sum_{j=1}^{s_2} W_{ij}^{(2)} a_j^{(2)} + b_i^{(2)}\right)$, where $a_j^{(2)}$ are the activations of the hidden units (2nd layer) and the sigmoid function $f(z) = \frac{1}{1+\exp(-z)}$ has been chosen as the activation function for the neurons. This activation function gives values between 0 and 1, while it regulates the weights of the network to change gradually and output better results. Additionally, the sigmoid function introduces non-linearity into the model, thus aiding in capturing non-linear combinations of the input data. The second term refers to the weight decay term that tries to decrease the magnitude of the weights ($W_{ji}^{(l)}$) among the nodes of the layers, while λ controls the importance of the weight decay term. The last term is a function that applies the sparsity penalty, where $KL(\rho||\hat{\rho}_j) = \rho \log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{(1-\rho)}{(1-\hat{\rho}_j)}$ is the Kullback-Leibler (KL) divergence that can determine the difference between two distributions having ρ and $\hat{\rho}_j$ mean values respectively. That is, ρ defines a desired level of sparsity, while $\hat{\rho}_j$ is the average activation of the j-th hidden unit. The magnitude of the sparsity penalty is regulated by the β weight.

[0037] Input reconstruction through Feedforward Autoencoder 134.

[0038] The training process of a sparse autoencoder defines weight and biase vectors $(W, b) = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$, in an embodiment utilizing an encoder with three layers. Next, these vectors can be used in a feedforward manner over a new input for finding a new and more informative structure of this input. In other words, the knowledge acquired from the unlabeled data U that fed into the sparse autoencoder can now be exploited for restructuring another dataset. This reconstruction is driven by a new representation which is learned out from unlabeled data, i.e., data that stem from an unknown environment.

[0039] Following this principle, a system according to the present invention can generate a new representation of the basic labeled dataset ($T = \left\{\left(x_l^{(1)}, y^{(1)}\right), \left(x_l^{(2)}, y^{(2)}\right), \ldots, \left(x_l^{(m)}, y^{(m)}\right)\right\}$), which was initially used to train the IDS. This is done toward producing a new labeled training set $\hat{T}$ that has as features the activations of the hidden units. That is, given T as the new input in a feedforward autoencoder 134, the system can calculate the new activation vectors using the weights and biases of the first layer $W^{(1)}, b^{(1)}$ by applying the activation function. As a result, the system produces a new dataset $\hat{T} = \left\{\left(a_l^{(1)}, y^{(1)}\right), \left(a_l^{(2)}, y^{(2)}\right), \ldots, \left(a_l^{(m)}, y^{(m)}\right)\right\}$, where $a_l^{(i)}$ represents the i-th new training

example. Thus, each $a_l^{(i)}$ example is a vector set of s_2 activations, $a_l^{(i)} = \left\{a_{l_1}^{(i)}, a_{l_2}^{(i)}, \ldots, a_{l_{s_2}}^{(i)}\right\}$, and each activation is given as follows (Equation 2):

$$a_{l_p}^{(i)} = f\left(\sum_{j=1}^{s_1} W_{pj}^{(1)}\, x_{l_j}^{(1)} + b_p^{(1)}\right), \qquad where\ p = \{1, \ldots, s_2\} \quad (2)$$

[0040] Finally, the new training dataset $\hat{T}$ can be used to train a supervised learning method 142.

[0041] The feedforward autoencoder 134 method adds an extra layer of data transformation. That is, any instance which will be subjected into the final detection model 154 for detection purposes, passes first through the feedforward autoencoder 134 to acquire the same transformation properties. This is why the feedforward autoencoder 134 component in the embodiment of Figure 1 is extended also into the detection subsystem 150.

[0042] All in all, the planning subsystem 130 takes as inputs the T and U and produces $\hat{T}$. Embodiments use the benefits of STL to provide an improved method enabling self-adaptation in the context of intrusion detection systems.

[0043] Execution subsystem: The outcome of the Planning subsystem 130 is a feedforward autoencoder 134 which is used for reconstructing the initial labeled dataset T and acquire $\hat{T}$. Hence, the Execution subsystem 140 undertakes the training of a supervised learning method 142 based on the new dataset $\hat{T}$. This step does not impose any constrains regarding the supervised learning method 142 that can be used to empower the detection system 154. An embodiment of the present invention makes use of Support Vector Machine (SVM) to deliver a multi-classification detection model. After training the new model, the old one, which due to the environmental changes had started facing efficiency problems, can now be replaced.

[0044] Knowledge repository**:** During the adaptive system loop, the Knowledge repository 160 component is accountable for storing purposes. The Knowledge repository 160 supports the adaptive operations and helps exchanging the inputs and outputs of each subsystem among them. More specifically, the repository 160 stores the sniffed network traffic as the result of the Monitoring subsystem 110. Upon the adaptation signal of the

network mapper 112, these captures will become the input of the network audit tool 122 for generating the network flows. Additionally, the repository 160 holds the initial labeled dataset T, which is used as a basis every time the system performs an adaptation loop.

[0045] Detection subsystem: The detection subsystem 150 undertakes the detection of offensive incidents occurring to the protected network. To do so, the unlabeled network traffic (U) passes through the Ethernet tap 170 and gets subjected to the same transformation applied to the training set $\hat{T}$ used to generate the detection model 154. That is, the network traffic gets translated by the Network audit tool 152 into machine learning-ready instances, and then is passed through the feedforward autoencoder 134 in order to acquire the same representation of the new training set $\hat{T}$. In this way, the unknown traffic passes through the detection model 154 for detecting potential offensive incidents.

[0046] In order to evaluate the performance of the present invention, an example embodiment was subjected to 100 consecutive network environmental changes and its performance was compared against a statically trained IDS.

[0047] The evaluation of the example embodiment was based on the KDDCup'99 (see, e.g., Lee) and NSL-KDD (see, e.g., Tavallaee) datasets. The aforementioned datasets were merged to create a single voluminous dataset that bears as many network traffic instances and as many attack classes as possible. Table 1 presents the instances of the used dataset. In total, the compiled dataset has approximately 1.3 million network instances and includes 40 classes (1 normal + 39 attacks), which come under different probability distributions and fall into the following 5 major categories Normal, DoS (denial of service), PRB (probing), R2L (remote to local) and U2R (user to root).

Class	**KDDCup'99 and NSL-KDD subclasses and the number of instances**	**#Instances**
Normal	Normal traffic is not divided into sub-classes	936,152
DoS	back (2,633), neptune (297,085), smurf (6,688), teardrop (1,828), land (46), pod (448), apache2 (1,531), mailbomb (601), processtable (1,429), udpstorm (4)	312,293
PRB	satan (10,226), portsweep (6,787), ipsweep (7,411), nmap (3,200), mscan (2,044), saint (683)	30,351

R2L	ftp_write (22), warezclient (1,783), spy (4), named (34), warezmaster (1,986), multihop (50), xsnoop (8), sendmail (29), snmpguess (690), imap (25), snmpgetattack (357), worm (4), xlock (18), phf (12), guess_passwd (2,639)	7,661
U2R	buffer_overflow (102), httptunnel (278), loadmodule (22), perl (10), rootkit (46), xterm (26), ps (31), sqlattack (4)	519
Total	40 classes	1,286,976

TABLE 1: Normal and Attack Classes in KDDCup'99 and NSL KDD

[0048] All duplicates were removed from the merged dataset to avoid any bias to the classification end-model. Hence, the compiled dataset consists of 1.3 million instances without duplicates. The KDDCup dataset created over a network experiment that lasted for 9 weeks and the final result was a dataset of approximately 7 million network instances with duplicates. The compiled dataset consists of 1.3 million instances without duplicates. This implies that the dataset corresponds to a data collection period of at least 12 days. Hence, the compiled dataset comprises a realistic collection of network traffic that spans adequately over time and it is thus suitable for evaluating an adaptive mechanism.

[0049] Additionally, the example embodiment was evaluated using the following metrics:

$$Accuracy\ (Acc) = \frac{\sum_{i=1}^{C} TP_i}{N}$$

$$Mean\ F-Measure\ (MFM) = \frac{\sum_{i=1}^{C} FMeasure_i}{C}$$

$$FMeasure_i = \frac{2 \cdot Recall_i \cdot Precision_i}{Recall_i + Precision_i}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i}$$

$$Average\ Accuracy\ (AvgAcc) = \frac{1}{C}\sum_{i=1}^{C} Recall_i$$

$$Attack\ Accuracy(AttAcc) = \frac{1}{C-1}\sum_{i=2}^{C} Recall_i$$

$$Attack\ Detection\ Ratio\ (ADR) = \frac{\sum_{i=2}^{C} TP_i}{\sum_{i=2}^{C} TP_i + FP_i}$$

$$False\ Alarm\ Rate\ (FAR) = \frac{FN_1}{TP_1 + FN_1}$$

Where:

TP_i, FP_i, TN_i, FN_i are the True positives, False Negatives, True Negatives and False Negatives of the i-th class, respectively;

C is the number of classes of the dataset; and

Index $i = 1$ stands for the normal traffic class.

[0050] The example embodiment was exposed to an ever-changing environment to illustrates its ability to adapt. In the evaluation the self-adaptive and autonomous example embodiment of the present invention was compared against a statically trained IDS. Both IDSs were initially trained with the same representative dataset T, which included a fraction of 10% of normal traffic and a randomly chosen subset of attack traffic. This attack-focused subset consists of 3, 3, 3 and 4 attacks subclasses of the major classes DoS, PRB, U2R, and R2L respectively.

[0051] Consequently, both IDSs were imposed to 100 network environment changes. Each environment is a randomly selected piece of the dataset which consists of 10% of normal traffic and 5, 5, 5, 8 attacks subclasses of the major classes DoS, PRB, U2R and R2L, respectively. In fact, these dataset pieces constitute the unlabeled and unknown dataset U. U might or might not contain the classes or the instances gathered in T. Additionally, apart from the diversity of the classes, these random dataset pieces have high diversity in their features. This means that, among others, features such as the protocols, services, and incoming/outbound traffic patterns constantly change. Hence, depending on the divergence between T and U the new environment can be either slightly or very different from the initial one. That is, it is expected to witness a low or even high drop of the IDS efficiency respectively.

[0052] As can be seen in Figure 2, the self-adaptive and autonomous example embodiment of the present invention surpasses the static IDSs in most of the environmental

states. More specifically, in 84% of the states the adaptive method of the example embodiment achieved a higher accuracy score compared to the static method of the comparison IDS. The average accuracy of the static method was 59.71%, while the average accuracy of the example embodiment's adaptive method was 77.99%. This means that in average the example embodiment performs better by 18.28% over the 100 unknown states. Additionally, the standard deviation is 30.79% and 18.78% for the static and the adaptive methods respectively. This fact quantifies what intuitively can be observed from Figure 2, where the adaptive curve witnesses less and smaller efficiency drops over the vast majority of the states. The maximum positive accuracy difference between the two methods is 56.92% (state #8), while the maximum negative difference is -1.6% (state #36). In fact, as can be seen in Figure 2, in critical cases where the IDS accuracy drops significantly due to a state's high deviation with respect to the initial training set (T), the adaptive methodology demonstrates a significantly higher contribution that can sustain the IDS to acceptable detection levels. All in all, the adaptive approach of the present invention greatly outperforms the static approach. especially when it comes to critical states.

[0053] Figure 3 presents the ADR performance over the 100 environmental states. The ADR measures the accuracy in detecting exclusively attacks instances, and thus reveals the performance in offensive incident detection. Overall, the adaptive method of the example embodiment scores an average ADR of 60.34% and outweighs the comparison static method by 23.8%, as the latter scores an average ADR of 36.54%. The standard deviations are 28.34% and 19.69% for the static and the adaptive method respectively. In total, the adaptive approach of the present invention is proved better for the 86% of the states and, notably, the maximum ADR increment is 73.37% (state #8), while the maximum deficient percentage is -5.67% (state #36). As in the case of the accuracy metric, ADR achieves high scores for those states where the static approach witnesses significant performance drops.

[0054] The overall performance of the adaptive method of the example embodiment and the comparison static method is illustrated in Figure 4. The dominance of the adaptive method is verified by all the metrics. Apart from the accuracy and the ADR metrics analyzed above in detail, also the rest of metrics prove the superiority of embodiments of the present invention. The difference of 4.78% in the MFM metric reveals that the adaptive method is able to keep the balance between Recall and Precision among all the dataset classes to a greater extent. Note that the MFM metric, as defined above, is the unweighted average of

recall and precision. That is, the unweighted MFM constitutes a stricter metric to evaluate the methods, as it treats all classes equally independently of the classes' size. This means that the adaptive method of the present invention is not only able to provide better attack detection rates, but it is also capable of identifying with higher precision the correct class where the attack instances belong to. Finally, the small deficiency (0.2%) in the FAR metric can be characterized as negligible.

[0055] One value of embodiments of the present invention lies in the fact that embodiments can breathe new life into the IDS in critical/sudden situations and increase ADR by up to 73.37%. In principle, in critical situations where the IDS performance drops significantly there is an urgent need for human intervention. Namely, in these cases, ADR can drop to such deficient levels that most of the attacks occurring in the network can go completely unnoticed. Hence, instead of triggering a process of manually retraining the IDS, our method empowers a self-adaptive and autonomous system to keep the IDS's operational ability to high levels.

[0056] FIG. 5 is a block diagram of a processing system according to an embodiment. The processing system 700 can be used to implement the protocols, devices, mechanism, systems and methods described above. The processing system 700 includes a processor 704, such as a central processing unit (CPU) of a computing device or a distributed processor system. The processor 704 executes processor executable instructions comprising embodiments of the system for performing the functions and methods described above. In embodiments, the processor executable instructions are locally stored or remotely stored and accessed from a non-transitory computer readable medium, such as storage 710, which may be a hard drive, cloud storage, flash drive, etc. Read Only Memory (ROM) 706 includes processor executable instructions for initializing the processor 704, while the random-access memory (RAM) 708 is the main memory for loading and processing instructions executed by the processor 704. The network interface 712 may connect to a wired network or cellular network and to a local area network or wide area network, such as the Internet.

[0057] While the invention has been illustrated and described in detail in the drawings and foregoing description, such illustration and description are to be considered illustrative or exemplary and not restrictive. It will be understood that changes and modifications may be made by those of ordinary skill within the scope of the following claims. In particular, the present invention covers further embodiments with any combination of features from

different embodiments described above and below. Additionally, statements made herein characterizing the invention refer to an embodiment of the invention and not necessarily all embodiments.

[0058] The terms used in the claims should be construed to have the broadest reasonable interpretation consistent with the foregoing description. For example, the use of the article "a" or "the" in introducing an element should not be interpreted as being exclusive of a plurality of elements. Likewise, the recitation of "or" should be interpreted as being inclusive, such that the recitation of "A or B" is not exclusive of "A and B," unless it is clear from the context or the foregoing description that only one of A and B is intended. Further, the recitation of "at least one of A, B and C" should be interpreted as one or more of a group of elements consisting of A, B and C, and should not be interpreted as requiring at least one of each of the listed elements A, B and C, regardless of whether A, B and C are related as categories or otherwise. Moreover, the recitation of "A, B and/or C" or "at least one of A, B or C" should be interpreted as including any singular entity from the listed elements, e.g., A, any subset from the listed elements, e.g., A and B, or the entire list of elements A, B and C.

CLAIMS

What is claimed is:

1. A method for adaptive network intrusion detection, the method comprising:

a) deploying a network traffic capture system and collecting network packet traces;

b) using a network audit tool (122) to extract features from the collected network packet traces;

c) feeding the extracted features as unlabeled data (U) into a representation function and utilizing the representation function as an unsupervised feature learning algorithm (132) to learn a new representation of the unlabeled data (U);

d) providing a labeled training set (T) capturing examples of malicious network traffic and using the learned new representation of the unlabeled data (U) to modify the labeled training set (T) to obtain a new training set ($\hat{T}$); and

e) using the new training set ($\hat{T}$) to train a traffic classification machine learning model.

2. Method according to claim 1, wherein the representation function is an autoencoder neural network.

3. The method according to claim 1 or 2, wherein the method further comprises:

using live traffic for the traffic classification machine learning model and deploying the traffic classification machine learning model for examining the live traffic.

4. The method according to claim 3, wherein before using the live traffic for the traffic classification machine learning model, extracting features from the live traffic and modifying the live traffic's extracted features using the new representation of the unlabeled data (U) learned in operation c).

5. The method according to any of claims 1 to 4, wherein operations c)-e) are repeated periodically to adapt to network traffic changes.

6. The method according to any of claims 1 to 5, further comprising:

collecting, by network mappers (112) of a monitoring subsystem (110), information about network characteristics, including its topology, the available hosts, the running services, open ports, the operating systems, and/or potential vulnerabilities.

7. The method according to claim 6, further comprising, by the monitoring subsystem (110):

determining, based on the collected information, a need for adaptation, and

issuing an adaptation signal towards the network audit tool (122) for triggering execution of operations c)-e).

8. The method according to any of claims 1 to 7, wherein the traffic classification machine learning model is based on a supervised learning method (142) that makes use of a support vector machine, SVM, to deliver a multi-classification detection model.

9. A system for adaptive network intrusion detection, the system comprising:

a network traffic capture system (114) configured to collect network packet traces;

a network audit tool (122) configured to extract features from the collected network packet traces;

a representation function configured to receive from the network audit tool (122) the extracted features as unlabeled data (U), to execute an unsupervised feature learning algorithm (132) to learn a new representation of the unlabeled data (U), to receive a labeled training set (T) capturing examples of malicious network traffic, and to use the learned new representation of the unlabeled data (U) to modify the labeled training set (T) to obtain a new training set ($\hat{T}$); and

a traffic classification machine leaning model configured to be trained using the modified training set ($\hat{T}$).

10. The system of claim 9, further comprising:

a planning subsystem (130) that is configured to deploy the traffic classification machine learning model to examine live traffic on the network.

11. The system according to claim 10, wherein the planning subsystem (130) is further configured to extract features from the live traffic and to modify the live traffic's extracted features using the new representation of the unlabeled data (U) learned by the execution of the unsupervised feature learning algorithm (132), before using the live traffic for the traffic classification machine learning model.

12. The system according to any of claims 9 to 11, further comprising:

a monitoring subsystem (110) including one or more network mappers (112), the network mappers (112) being configured to collect information about network characteristics, including its topology, the available hosts, the running services, open ports, the operating systems, and/or potential vulnerabilities.

13. The system according to claim 12, wherein the monitoring subsystem (110) is further configured

to determine, based on the collected information, a need for adaptation, and

to issue an adaptation signal towards the network audit tool (122) for triggering activation of the representation function, the representation function preferably being an autoencoder neural network.

14. The system according to any of claims 9 to 13, further comprising:

a repository (160) that is configured to store network traffic collected by the network traffic capture system and to hold the labeled training set (T).

15. The system according to any of claims 9 to 14, further comprising a detection subsystem (150) including

a network audit tool (152) that is configured to receive network traffic and to translate received network traffic into machine learning-ready instances, and

a feedforward autoencoder (134) that is configured to receive the machine learning-ready instances from the network audit tool (152), and to acquire the same representation of

the instances as the new training set ($\hat{T}$), before the network traffic passes through a detection model (154).

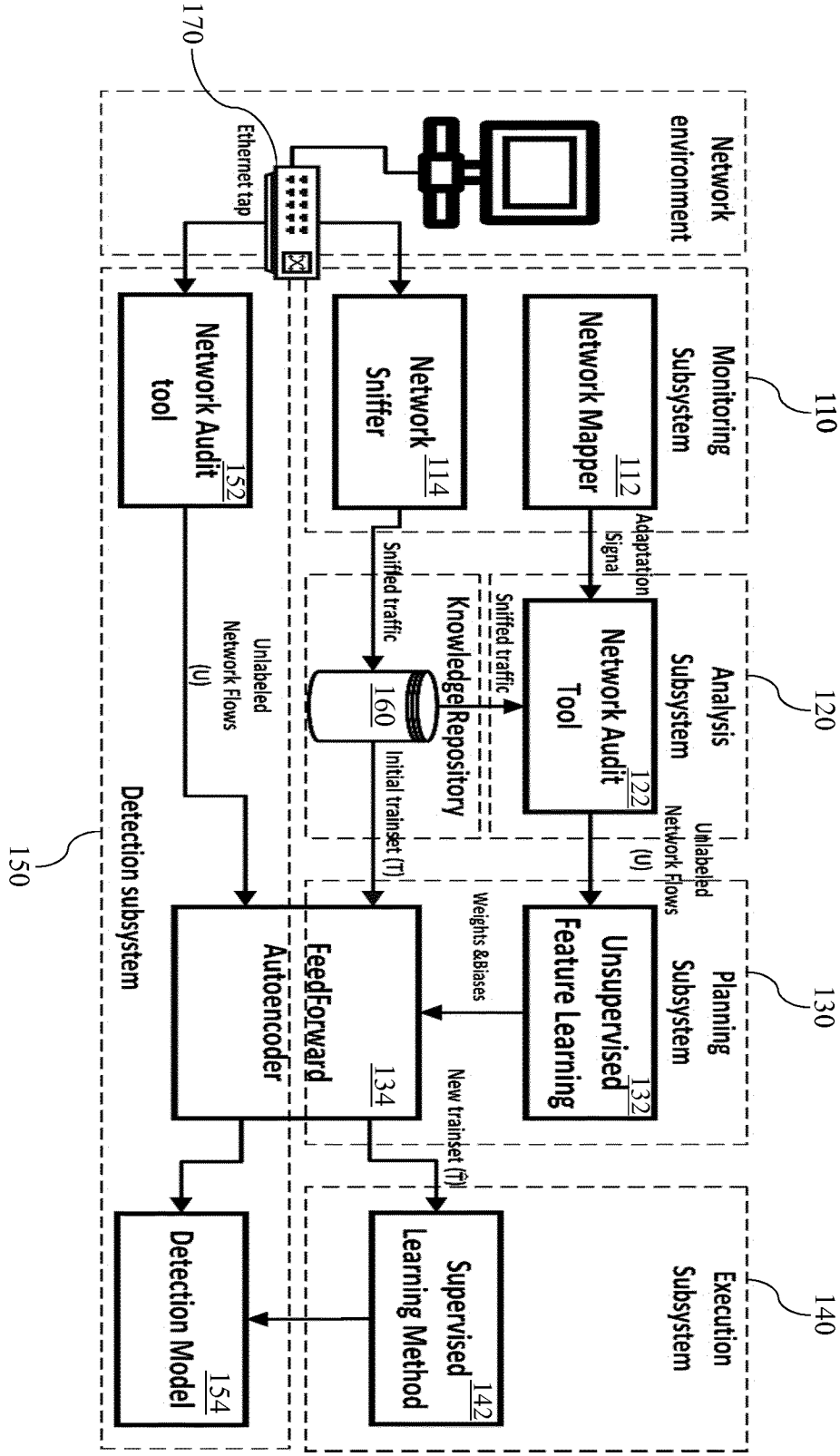

170
Ethernet tap
Network environment
110
Monitoring Subsystem
Network Mapper 112
Network Sniffer 114
Network Audit tool 152
Adaptation Signal
Sniffed traffic
Sniffed traffic
Knowledge Repository 160
Unlabeled Network Flows (U)
120
Analysis Subsystem
Network Audit Tool 122
Unlabeled Network Flows (U)
Initial trainset (T)
130
Planning Subsystem
Unsupervised Feature Learning 132
Weights &Biases
FeedForward Autoencoder 134
New trainset (T')
140
Execution Subsystem
Supervised Learning Method 142
Detection Model 154
150
Detection subsystem

FIG. 1

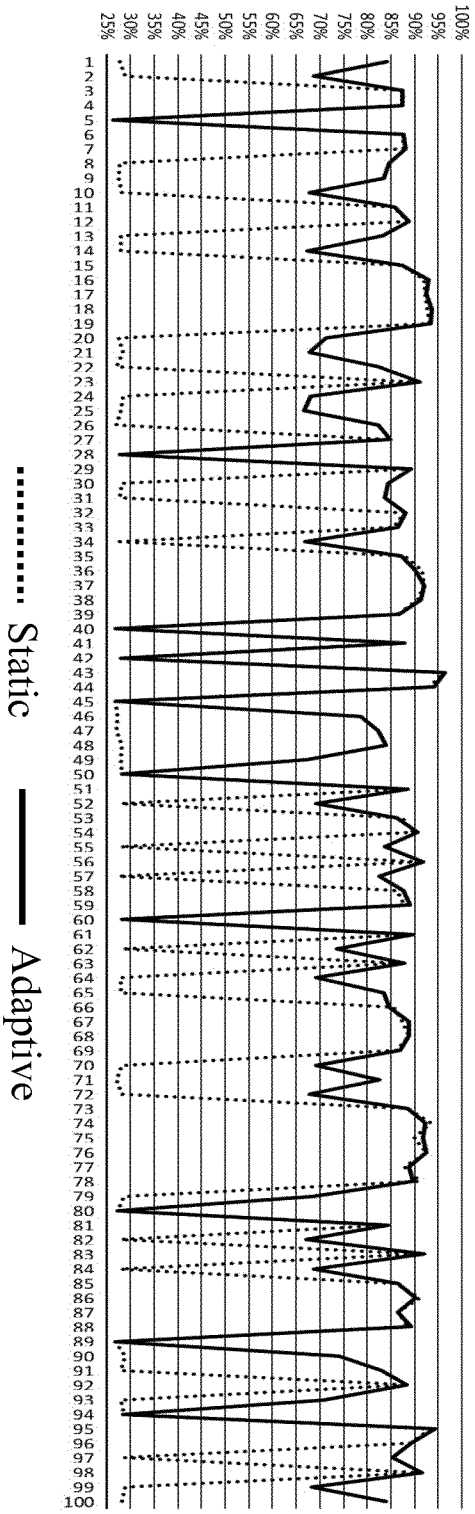
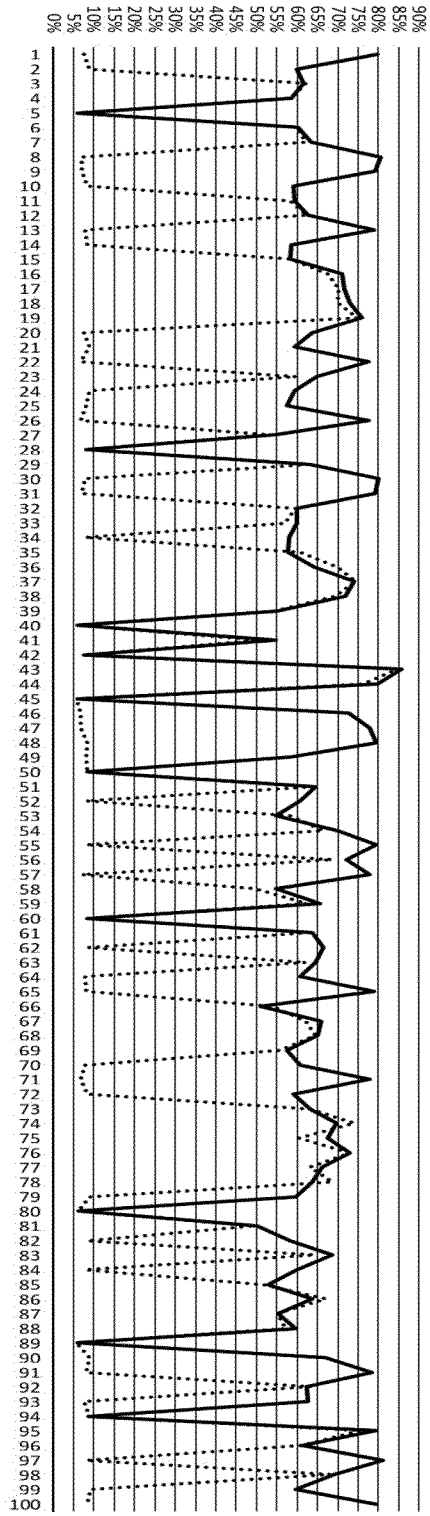

100%
95%
90%
85%
80%
75%
70%
65%
60%
55%
50%
45%
40%
35%
30%
25%
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
Static
Adaptive

FIG. 2



..... Static ——— Adaptive

FIG. 3

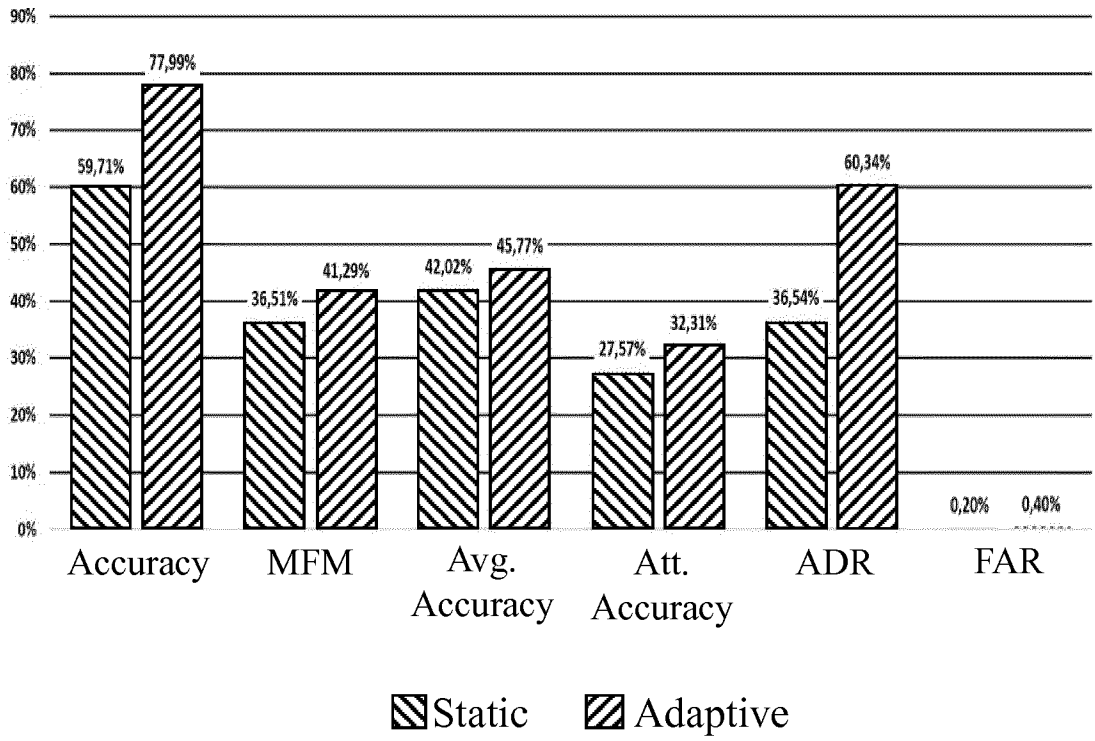

90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
59,71%
77,99%
36,51%
41,29%
42,02%
45,77%
27,57%
32,31%
36,54%
60,34%
0,20%
0,40%
Accuracy
MFM
Avg. Accuracy
Att. Accuracy
ADR
FAR
Static
Adaptive

FIG. 4

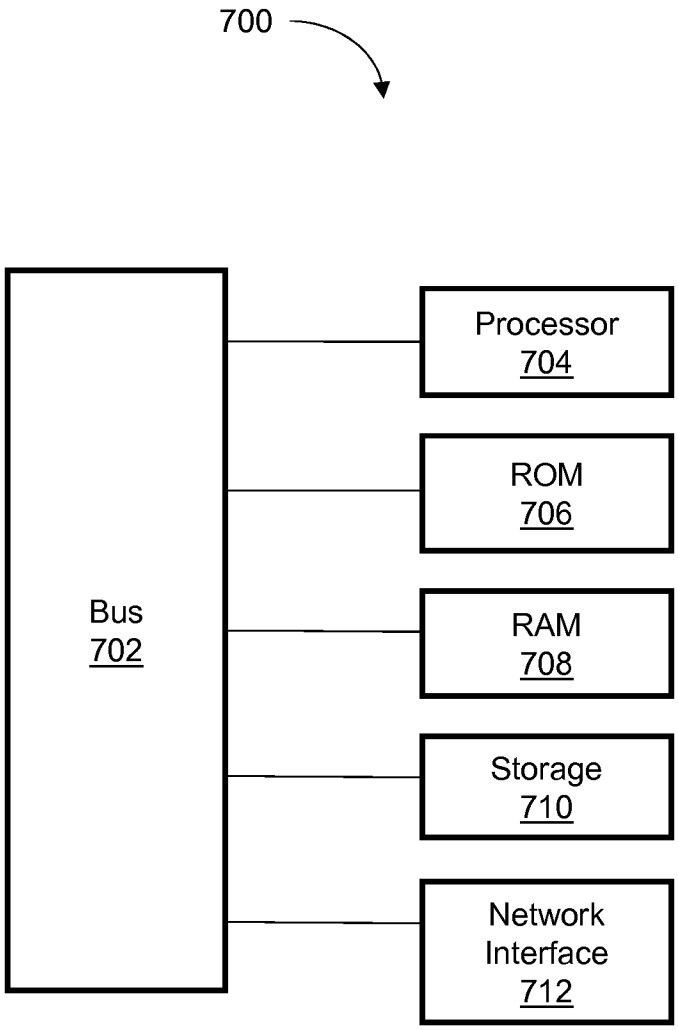

700
Bus
702
Processor
704
ROM
706
RAM
708
Storage
710
Network
Interface
712

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2019/074274

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06 H04W12/12 G06F21/55 G06F21/56
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L H04W G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	KIM KWANGJO ET AL: "Deep learning in intrusion detection perspective: Overview and further challenges", 2017 INTERNATIONAL WORKSHOP ON BIG DATA AND INFORMATION SECURITY (IWBIS), IEEE, 23 September 2017 (2017-09-23), pages 5-10, XP033308856, DOI: 10.1109/IWBIS.2017.8275095 [retrieved on 2018-01-29] Sections I-III the whole document	1-15

-/--

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
2 December 2019	13/12/2019

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Martínez Cebollada

2

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	MIN ERXUE ET AL: "SU-IDS: A Semi-supervised and Unsupervised Framework for Network Intrusion Detection", 13 September 2018 (2018-09-13), INTERNATIONAL CONFERENCE ON FINANCIAL CRYPTOGRAPHY AND DATA SECURITY; [LECTURE NOTES IN COMPUTER SCIENCE; LECT.NOTES COMPUTER], SPRINGER, BERLIN, HEIDELBERG, PAGE(S) 322 - 334, XP047485226, ISBN: 978-3-642-17318-9 [retrieved on 2018-09-13] abstract Sections 2-3 the whole document -----	1-15
X	AHMAD A AL SALLAB ET AL: "Self learning machines using Deep Networks", SOFT COMPUTING AND PATTERN RECOGNITION (SOCPAR), 2011 INTERNATIONAL CONFERENCE OF, IEEE, 14 October 2011 (2011-10-14), pages 21-26, XP032028187, DOI: 10.1109/SOCPAR.2011.6089108 ISBN: 978-1-4577-1195-4 page 22, column 1, paragraph 2 - paragraph 3 Sections I, IV -----	1-15