Artificial Intelligence in Network Intrusion Detection

Miroslav Stampar

Information Systems Security Bureau
Fra Filipa Grabovca 3, Zagreb, Croatia

mstampar@zsis.hr

Abstract *– In the beginning of the Internet era detection of network attacks has been almost solely done by human operators. They anticipated network anomalies in front of consoles, where based on their expert knowledge applied necessary security measures. With the exponential growth of network bandwidth, this task slowly demanded substantial improvements in both speed and accuracy. One proposed way how to achieve this is the usage of artificial intelligence (AI), progressive and promising computer science branch, particularly one of its sub-fields – machine learning (ML) – where main idea is learning from data. In this paper author will try to give a general overview of AI algorithms, with main focus on their usage for network intrusion detection.*

Keywords*: network intrusion detection, artificial intelligence, machine learning, classification*

I. Introduction

Today, lack of network access is unthinkable in both business and personal life. With unrestrained and ever-increasing number of electronic devices connected to the Internet, attackers never had such large attack surface. This brings up the question how to successfully defend from both know and unknown dangers. The answer is not so straightforward, especially if we know that the number of threats is rising every year [1][2].

Valuable resources having network access require constant protection from all attempts to destroy, expose, alter, disable, steal or gain unauthorized access to and/or make unauthorized use of them. Their confidentiality, integrity and availability have to remain intact. Intrusion detection system (IDS) is a system specially designed to detect such malicious attempts.

As traditional IDSes are mainly signature-based, detecting only known attacks, their biggest problem is the inability to detect new or variant attacks [3]. One topic that intuitively stands out as a potential solution for solving this problem is ML, a type of AI that provides computer the ability to learn without being explicitly programmed for particular purpose.

ML deals with the construction and study of algorithms that can generalize (i.e. learn) from limited sets of data. Such algorithms operate by building models based on input and using those to make predictions or decisions, rather than following only explicitly programmed instructions. Having such characteristics makes them ideal candidates for intrusion detection purposes.

We structure the remainder of the paper as follows. In Section II, we begin with a brief overview of AI concepts. We then in Section III give a brief overview of intrusion detection concepts. In Section IV we present specific AI types used in intrusion detection field, together with different challenges and evaluation results found in literature. In Section V we present quantitative results gathered for observed trends in related scientific publications. Finally, in Section VI we summarize our paper and give a final conclusion.

II. Artificial Intelligence

AI is by definition the intelligence exhibited by machines and/or software. It is an academic field of study which studies the goal of creating intelligence. The central problems (or goals) of AI research include reasoning, knowledge representation, automated planning and scheduling, ML, natural language processing, computer vision, robotics and general intelligence (strong AI) [4]. In this paper we will mainly focus on ML as it seems to be the most promising AI sub-field for intrusion detection task.

There are three common problems that ML tries to solve: classification, regression and clustering. Classification means identifying group membership, regression involves estimating or predicting a response, while in clustering a set of inputs is divided into groups where members have similar characteristics. Depending on output, if the output variable takes class labels, we are talking about classification, if the output variable takes continuous numerical values, we are talking about regression, while if the output are subsets (i.e. *clusters*) we are talking about clustering. Thus regression is applicable for prediction type of problems as opposed to classification and clustering.

Depending on ways of learning, ML can be further split into three main categories: supervised, unsupervised and reinforcement learning. Supervised learning is fairly common in classification problems where the goal is often to get the computer to learn a classification system that we have created. Unsupervised learning seems much harder. The goal is to have the computer learn how to do something that we don't tell it how to do. It is a powerful tool for identifying structure in unlabeled data, reflecting the statistical properties of the overall collection of input patterns. Reinforcement learning is performed by interacting with an environment, where learning agent learns from the consequences of its actions, rather than from being explicitly taught. It is essentially trial and error learning.

There is no single AI algorithm achieving best accuracy for all situations. Hence, one way how to improve results is the usage of combined multiple algorithms to obtain better quality of reasoning than from any single one. There are generally two approaches: ensemble and hybrid. In case of ensemble classifiers, multiple but homogeneous weak models are combined, typically at the level of their individual output, using various merging algorithms (e.g. majority voting) [5]. Hybrid algorithms, in turn, combine completely different, heterogeneous AI approaches (e.g. through cascading).

III. Intrusion Detection

Intrusion detection can be defined as the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource. IDS is a device or software application that gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, including data for both intrusion (i.e. attacks from outside the organization) and misuse (i.e. attacks from within the organization). IDS come in a variety of “flavors” and approach the goal of detecting suspicious traffic in different ways.

Based on a running location, IDS can be network-based (NIDS) or host-based (HIDS). NIDS is used to monitor and analyze network traffic, collected using either a network tap, span port or hub, to protect a system from network-based threats. HIDS attempts to identify unauthorized, illicit, and anomalous behavior on a specific device. HIDS generally involves an agent installed on each system, monitoring and remotely alerting on local OS and application activity. In this paper we will mainly focus on NIDS systems.

Based on an underlying detection logic, IDS can be signature-based (misuse) or anomaly-based. A signature-based IDS monitors suspicious behavior and compare its characteristics against a database of signatures or attributes from known malicious threats, similar to the way most anti-virus (AV) software work. Main issue is the lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to the IDS. During that lag time IDS is unable to detect the new threat. An anomaly-based IDS monitors behavior and compares its characteristics against an established baseline. The baseline will identify what is “normal” for that subject and alert when behavior is detected which is anomalous, or significantly different, than the baseline. Its main problem is the higher false positive rate.

Based on how it reacts to a detected threat, IDS can be passive or reactive. Passive IDS simply detects and alerts, leaving decision to administrator to take action and block the activity or respond in some way. Reactive IDS, also known as intrusion prevention system (IPS), will take predefined proactive actions to respond to the threat. Typically, this means blocking any further network traffic from the source IP address or user.

IV. Using Artificial Intelligence for Network Intrusion Detection

A. Datasets

The Cyber Systems and Technology Group (formerly the DARPA Intrusion Detection Evaluation Group) of MIT Lincoln Laboratory, under Defense Advanced Research Projects Agency (DARPA ITO) and Air Force Research Laboratory (AFRL/SNHS) sponsorship, has collected and distributed the first standard corpora for evaluation of computer network IDSes [6]. Such efforts have been carried out in both 1998 and 1999, resulting with DARPA 1998 and DARPA 1999 datasets respectively, with DARPA 1998 being more popular one.

KDD Cup is the annual Data Mining and Knowledge Discovery competition organized by ACM Special Interest Group on Knowledge Discovery and Data Mining. The task for the classifier learning contest organized in conjunction with the KDD'99 conference was to learn a predictive model (i.e. a classifier) capable of distinguishing between legitimate and illegitimate connections in a computer network [7]. Dataset used during the competition was based on a DARPA 1998 dataset [8] and has become widely accepted as just KDD'99.

Both DARPA and KDD datasets consist of nearly 5 million training connection records (i.e. events) labeled as an intrusion or not an intrusion, and separate testing dataset consisting of seen and unseen attacks. Each connection record consists of 41 different attributes that describe the different features of the corresponding connection, categorized as follows: basic TCP features, content features, time-based traffic features and host-based features. All the forms of attack fell into one of three categories: Remote-to-Local (R2L), User-to-Root (U2R), Denial-of-Service (DOS) or Probing [9].

As already mentioned, KDD'99 is based on the DARPA 1998, which by itself has been strongly criticized by McHugh [10], mainly because of having the characteristics of the synthetic data. As result, some of the existing problems in DARPA 1998 remain in KDD'99 too. One of the most important deficiencies in both data sets is the huge number of redundant records, which causes the learning algorithms to be biased towards the frequent records, and thus prevent them from learning infrequent records which are usually more harmful to networks such as U2R and R2L attacks.

Currently, as there are only few public datasets like DARPA 1998 and KDD'99, much related work considers these for their experiments, while very few studies use non-public or their own datasets. From literature overview results given by Tsai et al. [11] it can be easily concluded that those two datasets are recognized as de-facto standards in intrusion detection field of study.

B. Evaluation

While evaluating IDS, for every possible test value there are two kinds of error: false positive (FP) and false negative (FN). FP occurs when an event is predicted as intrusive but it is in fact normal, while FN occurs when a truly intrusive event occurs without being recognized as one. In the other hand, true positive (TP) measures the proportion of actual positives which are correctly identified as such, while true negative (TN) measures the proportion of negatives which are correctly identified as such.

The performance of each classifier can be quantified using the detection rate (DR) and overall accuracy (OA) measures. DR shows the percentage of the true intrusions that have been successfully detected:

$$DR=\frac{TP}{TP+FN}\times 100\% \qquad (1)$$

OA is calculated as the total number of correctly classified intrusions divided by the total number of observations:

$$OA=\frac{TP+TN}{TP+TN+FP+FN}\times 100\% \qquad (2)$$

An efficient IDS necessitates high degree of DR and OA, while maintaining low false alarm rates. Accuracy is critical to develop an effective IDS, since high FP rate or low DR will make it practically unusable.

C. Neural network

When used without qualification, the terms “neural network” (NN) and “artificial neural network” (ANN) usually refer to a multilayer perceptron (MLP). MLP is a supervised learning algorithm based on the feed-forward neural network with one or more layers between input and output layer. Feed-forward means that data flows in one direction from input to the output layer (i.e. forward). This type of network is trained with the error back-propagation learning algorithm. The true power and advantage of MLP lies in its ability to represent both linear and non-linear relationships and in its ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics [13].

Anup et al. [14] used MLP for anomaly detection, where the proposed model is a single hidden layer neural network. The performance of this model tested on the DARPA 1998 dataset was a DR of 77% with 2.2% FP. Lippmann et al. [15] used selected generic keywords to detect the attack preparations and actions after the break-in. Results were used to train the MLP using back-propagation algorithm. This approach ensured a DR of 80% when it has been tested on the DARPA 1998 dataset. Debar et al. [16] stated that NNs are very slow in converging, they suffer from dimensioning and stability problems, training takes a lot of time to achieve a reasonable level of performance, and their adaptability is low because a partial retraining can lead to a network that forgets everything it has learned before.

D. Genetic algorithm

Genetic algorithm (GA) is a search algorithm that works similar to the process of natural selection. It begins with a sample set of potential solutions which then evolves toward a set of more optimal solutions. The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population “evolves” toward an optimal solution. Within the sample set, solutions that are poor tend to die out while better solutions mate and propagate their advantageous traits, thus introducing more solutions into the set that boast greater potential (the total set size remains constant; for each new solution added, an old one is removed). A little random mutation helps guarantee that a set won't stagnate and simply fill up with numerous copies of the same solution.

Li [17] proposed an IDS with 57 genes in chromosomes, where each gene represents single connection feature, like: source IP address, destination IP address, source port, destination port, duration, protocol, number of bytes sent by originator, number of bytes sent by responder, etc. Due to the effectiveness of the evaluation function, the succeeding populations are biased toward rules that match intrusive connection. Mukkamala et al. [18] achieved 97.04% OA on DARPA 1998 dataset for their implementation of IDS based on linear genetic programming (LGP). However, Teodoro et al. [19] stated that main disadvantage of this kind of IDS is the high resource consumption involved.

E. Fuzzy logic

Fuzzy logic (FL) is a form of many-valued logic where one deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets where variables may take on true or false values, FL variables may have a truth value that ranges in degree between 0 and 1. FL proponents claim this generality allows greater flexibility, freedom, accuracy and compactness when representing real world situations. All the usual properties of Boolean algebra can be extended to FL, and probability's degree of belief in a Boolean variable becomes a fuzzy variable's degree of truth. It provides a principled way to encode expert knowledge or heuristics into algorithms that mimic a human's approach to solving challenging problems by weighing different sources of information and making judgments based on a preponderance of the evidence [20].

Chimphlee et al. [21] proposed an IDS based on rough sets theory and fuzzy c-means. Rough sets theory has been used as a selection tool to discover data dependencies and reduce the number of attributes contained in a training dataset. Fuzzy c-means have been used to avoid hard boundaries between normal and intrusion classes by allowing representation of concepts that could be considered to be in more than one category. Proposed system achieved a total of 93.45% OA on KDD'99 dataset, while reduced number of features resulted with enhanced performance. Dickerson et al. [22] developed FIRE IDS which uses simple data mining techniques to process the network input data and expose metrics that are particularly significant to anomaly detection. These metrics are then evaluated as fuzzy set for every observed feature and later used to detect network attacks.

F. K-nearest neighbor

K-nearest neighbor (kNN) is one of the most simple and traditional non-parametric supervised learning algorithms. It stores all available cases and classifies new ones based on a similarity measure (e.g. distance function) calculated between feature vectors. Classification is being performed by a majority vote of object's *k* nearest neighbors, where *k* is typically chosen to be a small positive integer (e.g. 3). For example, if k is chosen to have a value 1, then object is assigned to the class of its nearest neighbor. It has to be noted that if k is chosen to be considerably large, it will result with greater classification time.

Liao et al. [23] proposed an kNN IDS, combined with signature verification, achieving DR of 100% for known attacks and 75% for novel attacks when tested on DARPA 1998 dataset. Li et al. [24] achieved DR of 99.7% (k=50) on KDD'99 dataset by using proposed kNN based IDS. Ma et al. [25] proposed a kNN IDS based on similarity as a quantitative measure for distance, achieving 90.28% DR (k=1) on KDD'99 dataset.

G. Naive Bayes

Naive Bayes (NB) is a supervised learning algorithm based on applying Bayesian theorem with the “naive” assumption of independence between every pair of features. Classification is produced by combining prior probabilities and likelihood, to form a posterior probability using the so-called Bayes' rule. Despite its simplicity, NB can often outperform more sophisticated classification algorithms in both speed and accuracy [26]. Also, they are particularly suited when dimensionality of the input is high [27].

Panda et al. [28] achieved better results on KDD'99 dataset with a system based on NB than on IDS based on back-propagation neural network. Amor et al. [29] achieved 91.52% OA on KDD'99 dataset. Farid et al. [30] proposed a hybrid IDS based on NB and decision tree, achieving 99.63% DR on KDD'99 dataset.

H. Decision tree

Decision tree (DT) is a supervised learning algorithm based on flowchart-like structure in which internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test and each leaf node represents a class label. Decision is taken after computing all attributes. The paths from root to leaf represent classification rules. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features (i.e. attribute). DT is simple to understand and to visualize, making its explanation for the resulting value easily explainable by Boolean logic.

Abbes et al. [31] compared DR of rule-based Snort and IDS based on DT on a custom dataset based on collected RPC protocol traffic. Snort detected only three, while proposed IDS detected all 46 different forms of attacks. Bouzida et al. [32] made comparison of DT with and without principal component analysis (PCA), a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. They reduced computation time on dataset KDD'99 by a factor of approximately thirty, with slight loss of OA from 92.60% to 92.05%. In comparison of NN and DT for intrusion detection, Bouzida et al. [33] concluded that while NNs are very interesting for generalization and very poor for new attacks detection, DTs have proven efficiency in both generalization and new attacks detection.

I. Support vector machine

Support vector machine (SVM) is a supervised learning algorithm based on concept of decision planes that define decision boundaries. An SVM constructs the hyperplane, or a set of hyperplanes in a high-dimensional space, that separates all data points of one class from those of the other class. The best hyperplane for an SVM is the one with the largest margin between the two classes. It often happens that classes can't be linearly separable. For this reason, the original finite-dimensional space can be mapped into a much higher-dimensional space, using what is called the “kernel trick”, presumably making the separation easier [34].

Laskov et al. [35] concluded that the best performance is achieved by the non-linear methods, such as SVM, MLP and rule-based methods. Khan et al. [36] presented results where their solution utilizing dynamically growing self-organizing tree (DGSOT) enhanced the training time of SVM on DARPA 1998 dataset from 17.34h to 13.18h and improved OA from 57.6% to 69.8%. Chen et al. [37] achieved OA of 86.79% on KDD'99 dataset by using rough set theory (RST) to preprocess data and reduce the number of dimensions before forwarding it to SVM for intrusion detection. Mulay et al. [38] concluded that the integration of DT and SVM model gives better results than the individual models.

J. Random forest

Random forest (RF) is a supervised learning algorithm that is based on a collection (ensemble) of tree predictors, rather than a single classification tree, where to grow each tree a random selection is made from the examples in the training set. Each tree gives a classification and we say that the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest), with basic principle that a group of “weak learners” can come together to form a “strong learner”.

Kim et al. [39] concluded that their approach based on RF has been able not only to show high DR, but also to figure out stable output of important features simultaneously. Also, they stated that performance of RF based IDS turns out to be comparable to that of SVM. Zhang et al. [40] achieved average DR of 92.58% on original KDD'99 dataset and 99.86% on balanced dataset, where minority classes have been over-sampled and majority classes down-sampled to increase the DR of minority intrusions.

K. Self-organizing map

Kohonen's self-organizing map (SOM) is an unsupervised learning algorithm producing a low-dimensional (typically two-dimensional), discretized representation of the training samples' input space, simply called a map. It consists of components called nodes or neurons. Each node is associated with a weight vector of the same dimension as the input data vectors and a position in the map space. Nodes in a 2-D layer learn to represent different regions of the input space where input vectors occur. In addition, neighboring neurons learn to respond to similar inputs, thus the layer learns the topology of the presented input space. They provide a way of representing multidimensional data in much lower dimensional spaces, where this process of reducing the dimensionality of vectors is essentially a data compression technique known as the vector quantization [41].

Srinivasan et al. [42] proposed an agent-based SOM IDS called SAPID, designed especially for wireless networks. They achieved an average DR of 97% on an undeclared dataset. Lichodzijewski et al. [43] achieved DR of 94% on DARPA 1998 dataset with conclusion that hierarchically built unsupervised neural network approach is able to produce encouraging results. Rhodes et al. [44] analyzed SOM's potential for intrusion detection. Their experiments shown that even single SOM, when trained on normal data, will detect anomalous behaviors. Also, the ratio by which normal and intrusive packets differed has been greater than an order of magnitude. Their conclusion was that IDS based on SOM should be particularly powerful because it never needs to be told what intrusive behavior looks like.

V. Literature Overview

For literature overview purposes quantitative analysis was done in form of scientific publication search for topic related terms. Each chosen term was inquired in combination with term “intrusion detection” to narrow down results to only the respective field. Google Scholar was used to perform searches through scientific literature, mainly because of its simplicity and openness. Total number of publications (excluding citations and patents) was recorded depending on different search objectives. Main goal of presented results is the outline of current trends provided in a comprehensible format.

In Fig. 1 trends for terms “artificial intelligence” and “machine learning” are shown for period 2010-2014 in form of a number of intrusion detection related publications for each particular year. Comparison is performed to the auxiliary category “other” (e.g. signature-based), where those two terms have been excluded from results by using excluding search operator “-”. From given results it is visible that the usage of AI, and ML as its most prominent sub-field, have an increasing tendency. Also, it can be concluded that ML constitutes the main area of AI used in intrusion detection field of study.

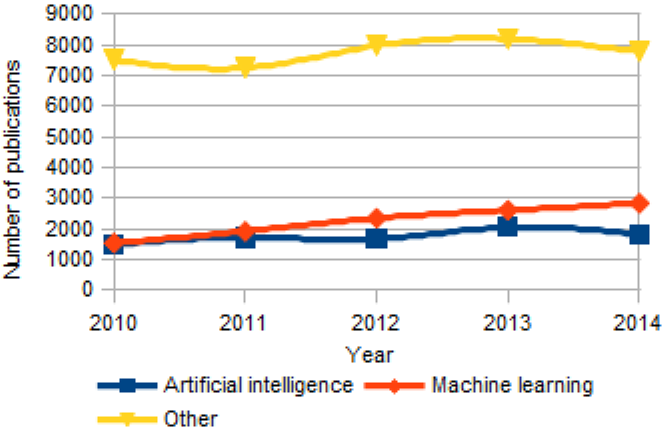

Fig. 1. Intrusion detection trends for AI and ML in comparison with other methods

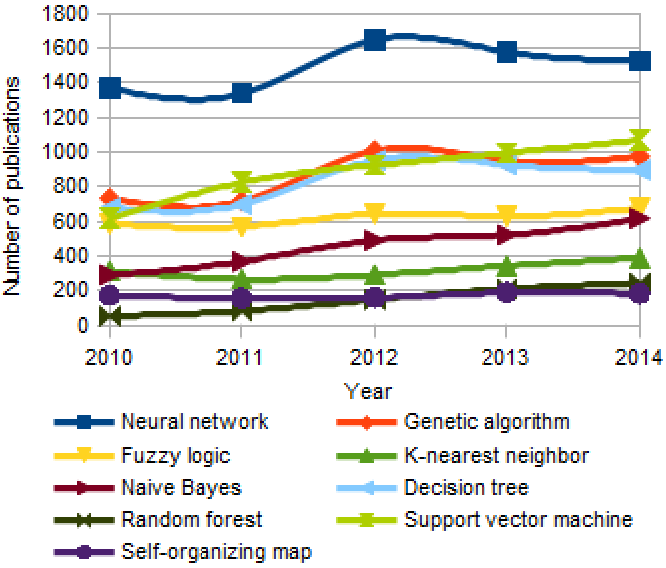

Fig. 2. Intrusion detection trends for different AI algorithms

In Fig. 2 trends for different AI algorithms are shown for period 2010-2014. It is clear that NN is the most popular AI algorithm used for intrusion detection compared to other algorithms, but with slight decrease in popularity over the last couple of years. After it there are GA and SVM, where SVM is constantly increasing in popularity. Out of the rest, SOM seems to be the least popular, which could be explained by the lack of an intuitive approach to solution of intrusion detection problem. It is clear from observed trends that AI composes a substantial part of the intrusion detection study, becoming more and more popular tool of choice in this field.

VI. Conclusion

As faster and more effective countermeasures are needed to cope with the ever-growing number of network attacks, AI comes as a natural solution to the problem. Albeit briefly, this paper discusses the foundations of AI in intrusion detection field. Most popular AI algorithms are presented along with the review of related studies, showing the maturity of the subject. At the end, results for the quantitative analysis of scientific literature are given, from where it has been concluded that AI plays a substantial role in the intrusion detection field of study, with NN being the most popular selection.

Further research in this area is necessary as there are very promising results that are obtainable from such algorithms, especially in area of their combined use.

References

[1] “Application Vulnerability Trends Report: 2014,” Cenzic Inc., Campbell, CA, 2014.

[2] “Cisco 2014 Annual Security Report,” Cisco Systems Inc., San Jose, CA, 2014.

[3] L. Wenke, S. J. Stolfo, P. K. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop, and J. Zhang. “Real time data mining-based intrusion

detection," in *Proc. DARPA Information Survivability Conference & Exposition II, 2001, DISCEX'01*, vol. 1, pp. 89-100. IEEE, 2001.

[4] S. D. Sekar. (2014). *What is the difference between Artificial Intelligence, Machine Learning, Statistics, and Data Mining*. [Online]. Available: http://shakthydoss.com/what-is-the-difference-between-artificial-intelligence-machine-learning-statistics-and-data-mining/

[5] P. Kazienko, E. Lughofer, and B. Trawiński, "Special Issue on Hybrid and Ensemble Methods in Machine Learning," *New Generation Computing 29*, no. 3, pp. 241-244, 2011.

[6] (2014) DARPA Intrusion Detection Data Sets webpage on The MIT Lincoln Laboratory website. [Online]. Available: http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/

[7] (2014) KDD Cup 1999: Computer network intrusion detection webpage on SIGKDD website. [Online]. Available: http://www.sigkdd.org/kdd-cup-1999-computer-network-intrusion-detection

[8] H. G. Kayacık, and N. Z. Heywood, "Analysis of three intrusion detection system benchmark datasets using machine learning algorithms," in *Proc. Intelligence and Security Informatics*, pp. 362-367, Springer Berlin Heidelberg, 2005.

[9] W. Lee, S. J. Stolfo, and K. W. Mok, "Mining in a data-flow environment: Experience in network intrusion detection," in *Proc. of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 114-124, ACM, 1999.

[10] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM transactions on Information and system Security* 3, no. 4, pp. 262-294, 2000.

[11] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Applications* 36, no. 10, pp. 11994-12000, 2009.

[12] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications,* 2009.

[13] C. Grosan and A. Abraham, *Intelligent systems: A modern approach*, vol. 17, Springer, 2011.

[14] A. K. Ghosh, and A. Schwartzbard, "A Study in Using Neural Networks for Anomaly and Misuse Detection," in *Proc. USENIX Security,* 1999.

[15] R. P. Lippmann, and R. K. Cunningham, "Improving intrusion detection performance using keyword selection and neural networks," *Computer Networks* 34, no. 4, pp. 597-603, 2000.

[16] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," in *Proc. Research in Security and Privacy, 1992 IEEE Computer Society Symposium on*, pp. 240-250, IEEE, 1992.

[17] W. Li, "Using genetic algorithm for network intrusion detection," in *Proc. of the United States Department of Energy Cyber Security Group* (2004): 1-8., 2004.

[18] S. Mukkamala, A. H. Sung, and A. Abraham, "Modeling intrusion detection systems using linear genetic programming approach," *Innovations in Applied Artificial Intelligence*, pp. 633-642. Springer Berlin Heidelberg, 2004.

[19] P. G. Teodoro, J. D. Verdejo, G. M. Fernández, and E. Vázquez. "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & security* 28, no. 1 (2009): 18-28, 2009.

[20] J. K. Williams, J. Craig, A. Cotter, and J. K. Wolff, "A hybrid machine learning and fuzzy logic approach to CIT diagnostic development," *AMS Fifth Conference on Artificial Intelligence Applications to Environmental Science*, 2007.

[21] W. Chimphlee, A. H. Abdullah, M. N. M. Sap, S. Srinoy, and S. Chimphlee, "Anomaly-based intrusion detection using fuzzy rough clustering," *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, vol. 1, pp. 329-334. IEEE, 2006.

[22] J. E. Dickerson, and J. A. Dickerson, "Fuzzy network profiling for intrusion detection," in *Proc. Fuzzy Information Processing Society, NAFIPS, 19th International Conference of the North American*, pp. 301-306, IEEE, 2000.

[23] Y. Liao, and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security* 21, no. 5, pp. 439-448, 2002.

[24] Y. Li, and L. Guo, "An active learning based TCM-KNN algorithm for supervised network intrusion detection," *Computers & security* 26, no. 7, pp. 459-467, 2007.

[25] Z. Ma, and A. Kaban, "K-Nearest-Neighbours with a novel similarity measure for intrusion detection," in *Proc. Computational Intelligence (UKCI), 13th UK Workshop on*, pp. 266-271, IEEE, 2013.

[26] H. Zhang, "The optimality of naive Bayes," *AA* 1, no. 2, 2004.

[27] S. Caballé, A. Abraham, T. Daradoumis, and A. A. J. Perez. *Computational intelligence for technology enhanced learning*, vol. 273, Springer, 2010.

[28] M. Panda, and M. R. Patra, "Network intrusion detection using naive bayes," *International journal of computer science and network security* 7, no. 12 (2007): 258-263, 2007.

[29] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive bayesian networks in intrusion detection systems," in *Proc. Workshop on Probabilistic Graphical Models for Classification, 14th European Conference on Machine Learning (ECML) and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), 23rd September, in Cavtat–Dubrovnik, Croatia.*, p. 11, 2003.

[30] D. M. Farid, N. Harbi, and M. Z. Rahman, "Combining naive bayes and decision tree for adaptive intrusion detection," *arXiv preprint arXiv:1005.4496*, 2010.

[31] T. Abbes, A. Bouhoula, and M. Rusinowitch, "Protocol analysis in intrusion detection using decision tree," in *Proc. Information Technology: Coding and Computing, ITCC 2004, International Conference on*, vol. 1, pp. 404-408, IEEE, 2004.

[32] Y. Bouzida, F. Cuppens, N. C. Boulahia, and S. Gombault, "Efficient intrusion detection using principal component analysis," in *Proc. 3éme Conférence sur la Sécurité et Architectures Réseaux (SAR), La Londe, France,* 2004.

[33] Y. Bouzida, and F. Cuppens, "Neural networks vs. decision trees for intrusion detection," in *Proc. IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM), Tuebingen, Germany*, pp. 28-29, 2006.

[34] B. Scholkopf, "The kernel trick for distances," *Advances in neural information processing systems (2001),* pp. 301-307, 2001.

[35] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: supervised or unsupervised?," in *Proc. Image Analysis and Processing–ICIAP 2005*, pp. 50-57, Springer Berlin Heidelberg, 2005.

[36] L. Khan, M. Awad, and B. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," *The VLDB Journal—The International Journal on Very Large Data Bases* 16, no. 4, pp. 507-521, 2007.

[37] R. C. Chen, K. F. Cheng, Y. H. Chen, and C. F. Hsieh, "Using rough set and support vector machine for network intrusion detection system," in *Proc. Intelligent Information and Database Systems, ACIIDS 2009, First Asian Conference on*, pp. 465-470, IEEE, 2009.

[38] S. A. Mulay, P. R. Devale, and G. V. Garje, "Intrusion detection system using support vector machine and decision tree," *International Journal of Computer Applications*, no. 3, pp. 40-43, 2010.

[39] D. S. Kim, S. M. Lee, and J. S. Park, "Building lightweight intrusion detection system based on random forest," in *Proc. Advances in Neural Networks-ISNN 2006*, pp. 224-230, Springer Berlin Heidelberg, 2006.

[40] J. Zhang, and M. Zulkernine, "Network Intrusion Detection using Random Forests," in *Proc. PST*, 2005.

[41] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, "Som pak: The self-organizing map program package," *Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science,* 1996.

[42] T. Srinivasan, V. Vijaykumar, and R. Chandrasekar, "A Self-organized Agent-based architecture for Power-aware Intrusion Detection in wireless ad-hoc networks," in *Proc. Computing & Informatics, ICOCI'06, International Conference on*, pp. 1-6, IEEE, 2006.

[43] P. Lichodzijewski, A. N. Z. Heywood, and M. I. Heywood, "Dynamic intrusion detection using self-organizing maps," in *Proc. The 14th Annual Canadian Information Technology Security Symposium (CITSS)*, 2002.

[44] B. C. Rhodes, J. A. Mahaffey, and J. D. Cannady, "Multiple self-organizing maps for intrusion detection," in *Proc. of the 23rd national information systems security conference*, pp. 16-19, 2000.