IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

The Applications of Artificial Intelligence and Machine Learning In the field of Cyber Security

Nishanth Vaidya

UG Student, Department of Computer Science

Sambhram Institute of Technology, Bengaluru

Nikhil Bharadwaj

PG Student, Department of Computer Science

Syracuse University, New York

Abstract*:* Machine learning has been adopted in a wide range of domains where it shows its superiority over other algorithms. These methods can also be integrated in cyber detection systems with the goal of supporting replacing the first level of security analysts. Although the automation of detection and analysis is an still a distant goal, the efficiency of machine learning in cyber security must be evaluated with the due diligence. We present an analysis, addressed to security specialists, of machine learning techniques applied to the different types of cyber attacks. The goal is to assess the current maturity of these solutions and to identify their main limitations that prevent an immediate adoption of those machine learning cyber detection schemes. Our conclusions are based on an extensive review of the literature as well as on experiments performed on real enterprise systems and network traffic in different conditions.

IndexTerms- **machine learning, deep learning, cyber security.**

1. INTRODUCTION

The appeal and pervasiveness of machine learning (ML) is growing rapidly. Existing methods are being contiguously improved, and their real world applications expand daily. These achievements have led to the adoption of machine learning in several domains, such as computer vision, medical analysis, gaming and cyber security etc. [1]. In certain scenarios, machine learning techniques represent the one of the best choice over traditional rule-based algorithms and even people in general [2]. This trend is also affecting the cyber security field where some detection systems are being upgraded via ML components [3]. Although devising a completely automated cyber defence system isn't yet an attainable objective, first level operators in Network and Security Operation Centres (NOC and SOC) may benefit from detection and analysis tools based on the concepts of machine learning. This paper is specifically addressed to security operations and aims to accurately assess the current maturity of these solutions, their drawbacks and how they can be overcome.

Our study is based on the literature survey's reviews and on original experiments performed on real, large enterprises and network traffic data. Other academic papers compare ML solutions for cyber security by considering one specific application (e.g.:[4], [3], [5]) and are typically oriented to AI experts rather than to security operators. In the evaluation, we leave out the commercial products based on machine learning (or on the abused AI term) because vendors do not reveal their algorithms and tend to overlook issues and limitations. Existing studies on this issue have primarily focused on estimating future process loads and traffics, and optimizing controls to reach required levels of efficiency in addition to meet specifically defined criteria. Whereas the focus of these models, based on estimation and control theories, is predominantly on the continuous control traffic mode, this experiment avoids optimization involvement as the process load/traffic manager actively learns and progresses at an unchanging rate. Therefore, the prime interest is whether a high index/load process thread can be masked or not, even if it causes a danger, or alternatively, if it can be simply be ignored. This aspect of cyber security falls into the category of multitasking and load management control methods. The learning system used in this experiment is based on algorithmic comparison and also pre-emptive task selection, delivered through a selective task scheduling approach that considers load management routine as a lone task.

2. MACHINE LEARNING METHODS

Machine learning includes a large variety of paradigms in continuous evolution, presenting weak boundaries and cross relationships. Furthermore, different views and applications may lead to different outcomes. Hence, we cannot refer to one fully accepted taxonomy from literature, but we prefer to propose an original taxonomy able to capture the differences among the myriad of techniques that are being applied to cyber detection, as shown in the figure below

Categories of Machine Learning:

Machine Learning Techniques are classified into the following parts:

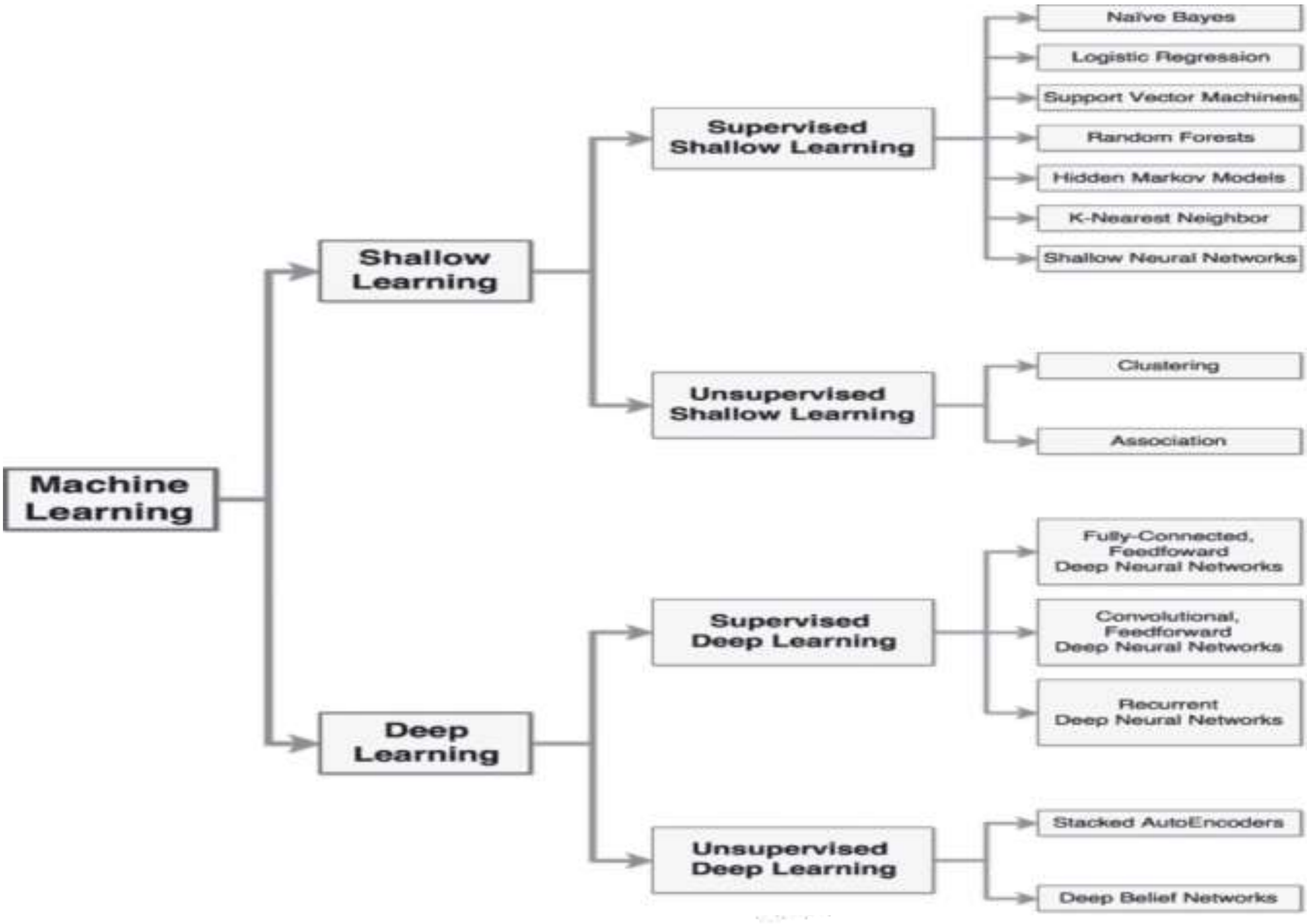

Figure 1: Machine learning categories

A. Shallow Learning

2.1. Supervised SL Algorithms

Supervised learning is the framework where the information and yield is acquired for future preparation. In this, there are two sorts of learning assignment being regression and classification. Some of the most common algorithms are Support Vector Machines (SVM), k-Nearest Neighbours (k-NN) and Artificial Neural Networks (ANN), Genetic algorithms and Decision Trees (DT).

- **Naïve Bayes (NB).** These algorithms are probabilistic classifiers which make the a-priori assumption that the features of the input dataset are independent from each other. They are scalable and do not require huge training datasets to produce appreciable results.
- **Logistic Regression (LR).** These are categorical classifiers that adopt a discriminative model. Like Naïve Bayes algorithms, Logistic Regression methods make the a-priori independency assumption of the input features. Their performance is highly dependent on the size of the training data.
- **Support Vector Machines (SVM).** These are non-probabilistic classifiers that map data samples in a feature space with the goal of maximizing the distance between each category of samples. They do not make any assumption on the input features, but they perform poorly in multi-class classifications. Hence, they should be used as binary classifiers. Their limited scalability

might lead to long processing times

2.2. Unsupervised SL Algorithms

Unsupervised learning involves taking in findings from the datasets including data without marked responses. Right now, there are two learning tasks being Association and Clustering. To find the associations of the objects in a database, Association learning was proposed by Rakesh Agarwal. The most regular count that was used in association rule is Apriori and grouping is used to assemble relative kind of data sets.

- **Clustering.** These group data points that present similar characteristics. Well known approaches typically include algorithms like k-means and *hierarchical* clustering. Clustering methods have a scalability that is somewhat limited, but they represent a feasible solution that is used as a preliminary phase before adopting a supervised algorithm or for anomaly detection purposes.
- **Association.** The aim is to identify unknown patterns between data, making them suitable for prediction purposes. However, they tend to produce an excessive output of not necessarily valid rules, hence they must be combined with accurate inspections by a human expert.

B. Deep Learning

All DL algorithms are based on Deep Neural Networks (DNN), which are large neural networks organized in multiple layers capable of autonomous representation learning.

2.3. Supervised DL algorithms

- **Fully-connected Feedforward Deep Neural Networks (FNN).** They are a variant of DNN where every neuron is connected to every other neuron in the previous layer. FNN do not make an assumption on the input data and provide a flexible and general-purpose solution for classification, at the expense of high computational costs.
- **Convolutional Feed Forward Deep Neural Networks (CNN).** They are a variant of DNN where each neuron receives its input only from a subset of neurons of the upper or previous layer. This characteristic makes CNN effective at analysing spatial data, but their performance decreases when applied to non-spatial data. CNN have a lower computation cost than FNN.
- **Recurrent Deep Neural Networks (RNN).** A variant of DNN whose neurons can send their output also to previous layers; this design makes them harder to train than FNN. They excel as generators of sequential data, especially their recent variant, the *long short-term memory*.

2.4. Unsupervised DL Algorithms

It is affiliated to how programming specialists naturally decide the perfect conduct for particular setting, so as to amplify it’s presentation. Fortification sends the reward input for the operator to get familiar with its working. It comprises of two learning errands being Classification and Control. Some of the applications are computer played board games, self-driving cars and robotic arms,. Most commonly used algorithms are DBN and SAE algorithm.

- **Deep Belief Networks (DBN).** They are modelled via a composition of *Restricted Boltzmann Machines* a class of neural networks with no output layer. DBN can be used for pre-training tasks because they excel in the function of feature extraction. They require a training phase, but with datasets that need not be labelled.
- **Stacked Auto-encoders (SAE)**. They are composed by multiple *Auto-encoders*, a class of neural networks where the number of input and output neurons is the same. SAE excel at pre-training tasks similarly to DBN, and achieve better results on small datasets.

3. LITERATURE SURVEY

[1] M. I. Jordan and T. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, year 2015.

A. Buczak and E. Guven, [3] “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Communications Surveys*, year 2015.

Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, Alessandro Guido, Mirco Marchetti [4] “On the Effectiveness of Machine and Deep Learning for Cyber Security” 2018 10th International Conference on Cyber Conflict

F. Pierazzi, G. Apruzzese, M. Colajanni, A. Guido, and M. Marchetti, [7]“Scalable architecture for online prioritization of cyber threats,” in *International Conference on Cyber Conflict”*, year 2017.

Rajesh N, T Maneesha, Shaika Hafeez, Hari Krishna[6] states that Naïve Bayes are more accurate when compared to Decision Tree Algorithm

4. MACHINE LEARNING BASED METHODS FOR CYBER SECURITY

A. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is one of the most robust and accurate methods in all machine-learning algorithms. It primarily includes Support Vector Classification (SVC) and Support Vector Regression (SVR). The SVC is based on the basis of decision boundaries. A decision boundary separates a set of instances having different class values between two groups. The SVC supports both binary and multiclass classifications. The support vector is the closest point to the separation hyperplane, which determines the optimal separation hyperplane. In the classification process, the mapping input vectors located on the separation hyperplane side of the feature space fall into one class, and the positions fall into the other class on the other side of the plane. In the case of data points that are not linearly separable, the SVM uses appropriate kernel functions to map them into higher dimensional spaces so that they become separable in those spaces [28]. Kotpalliwar et al. [29] choose two representative datasets "Mixed" and "10% KDD Cup 99" datasets. The RBF is used as a kernel function for SVM to classify DoS, Probe, U2R, and R2L datasets. The study calculates parameter values related to intrusion-detector performance evaluation. The validation accuracy of the "mixed" dataset and the classification accuracy of the "10% KDD" dataset were estimated to be 89.85% and 99.9%, respectively. Unfortunately, the study did not assess accuracy or recall except for accuracy. Saxena et al. [23] proposed a SVM approach for building IDS. The study used two feature reduction techniques namely Information Gain and BPSO. The 41 attributes reduced to 18 attributes. The classification performance was reported as 99.4% on the DoS, 99.3% on Probe or Scan, 98.7% on R2L, and 98.5% on the U2R. The method provides a good detection rate in the case of a Denial of Service (DoS) attack and achieves a good detection rate in the case of U2R and R2L attacks. However, the precision of Probe, U2R and R2L is 84.2%, 25.0% and 89.4%, respectively. In other words, the method provided by the essay leads to a higher false alarm rate. On the basis of a short sequence model, Xie et al. [35]applied a class SVM algorithm to ADFA-LD. Due to the short sequence removes duplicate entries, and between the normal and abnormal performed better separability, so the technology can reduce the cost of computing at the same time to achieve an acceptable performance limits, but individual type of attack mode recognition rate is low.

B. K-NEAREST NEIGHBOR

The kNN classifier is based on a distance function that measures the difference or similarity between two instances. The standard Euclidean distance d(x, y) between two instances x and y is defined as :

$$d(x,y)=\sqrt{\sum_{k=1}^{n}(x_k-y_k)^2}$$

where, xk is the k th featured element of instance x, yk is the k th featured element of the instance y and n is the total number of features in the dataset. Assume that the design set for kNN classifier is U. The total number of samples in the design set is S. Let C = {C1 ,C2 ,…CL} are the L distinct class labels that are available in S. Let x be an input vector for which the class label must be predicted. Let yk denote the k th vector in the design set S. The kNN algorithm is to find the k closest vectors in design set S to input vector x. Then the input vector x is classified to class Cj if the majority of the k closest vectors have their class as Cj [36]. Rao et al.[37] used Indexed Partial Distance Search kNearest Neighbor (IKPDS) to experiment with various attack types and different k values (i.e., 3, 5, and 10). They randomly selected 12,597 samples from the NSl-KDD dataset to test the classification results, resulting in 99.6% accuracy and faster classification time. Experimental results show that IKPDS, and in a short time Network Intrusion Detection Systrms(NIDS), have better classification results. However, the study of the test indicators of the experiment is not perfect; it did not consider the precision and recall rate. Another study [42] that had been used KNN for intrusion detection on the same KDD Cup 99 dataset in an approach similar to that of Vishwakarma et al.[41]. The main difference is that the kNN, SVM, and pdAPSO algorithms are mixed to detect intrusions. The experimental results show that mixing different classifiers can improve classification accuracy. The statistical results show that the classification accuracy is 98.55%. Other than accuracy, the study did not count other indicators.

C. DEEP BELIEF NETWORK

Deep Belief Network (DBN) is a probabilistic generative model consisting of multiple layers of stochastic and hidden variables. The Restricted Boltzmann Machine (RBM) and DBN are interrelated because composing and stacking a number of RBMs enables many hidden layers to train data efficiently through activations of one RBM for further training stages [56]. RBM is a special topological structure of a Boltzmann machine (BM). The learning model expresses the correlation between units by weighting. In the study, Ding and Yuxin [57] apply Deep Belief Nets (DBNs) to detect malware. They use PE files from the internet as samples. DBNs use unsupervised learning to discover multiple layers of features that are then used in a feed-forward neural network and fine-tuned to optimize discrimination. The unsupervised pre-training algorithm makes DBNs less prone to overfitting than feedforward neural networks initialized with random weights. It also makes it easier to train neural networks with many hidden layers. Because the DBNs can learn from additional unlabeled data, in the experiments, the DBNs produce better classification results than several other widely used learning techniques, outperforming SVM, KNN, and decision tree. The accuracy of the method is approximately 96.1%, but other specifications are not mentioned

5.CONCLUSIONS

This paper presents a literature review of ML and AI methods for network security. The paper, which has mostly focused on the last few years, introduces the latest applications of ML and AI in the field of intrusion detection. Unfortunately, the most effective method of intrusion detection has yet to be established. Each approach that implements an intrusion detection system has its own advantages and disadvantages, a point apparent from the discussion of comparisons among the various methods. Thus, it is difficult to choose a particular method to implement an intrusion detection system over the others methods. Datasets for network intrusion detection are very important for training as well as testing systems. The ML and DL methods do not work without representative datasets, and obtaining such a dataset is difficult and also time-consuming. However, there are many problems with the existing public dataset's, such as uneven data, outdated content etc. These problems have largely limited the development of research in this field. Network information updates very fast, which brings to the AI and ML model training and use with difficulty, the models need to be retrained long-term and quickly. So incremental learning and lifelong learning will be the focus in the study of this field for the time being.

6.REFERENCES

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, 2015.

[2]Yang Xin, Lingshuang Kong, Zhi Liu "Machine Learning and Deep Learning Methods for Cybersecurity"2017

[3]A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, 2015.

[4] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Comput. Netw., vol. 51, no. 12, pp. 3448–3470, 2007

[5] Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, Alessandro Guido, Mirco Marchetti "On the Effectiveness of Machine and Deep Learning for Cyber Security" 2018 10th International Conference on Cyber Conflict

[6]] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "Review: A survey of intrusion detection techniques in Cloud," J. Netw. Comput. Appl., vol. 36, no. 1, pp. 42–57, 2013.

[7] F. Pierazzi, G. Apruzzese, M. Colajanni, A. Guido, and M. Marchetti, "Scalable architecture for online prioritization of cyber threats," in *International Conference on Cyber Conflict (CyCon)*, 2017.

[8] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," in *IEEE International Conference on Platform Technology and Service (PlatCon)*, 2016.

[9] P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An analysis of Recurrent Neural Networks for Botnet detection behavior," in *IEEE Biennial Congress of Argentina (ARGENCON)*, 2016.

[10] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, "Large-scale malware classification using random projections and neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[11] G. D. Hill and X. J. Bellekens, "Deep Learning Based Cryptographic Primitive Classification," *arXiv preprint*, 2017.

[12] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[13] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *IEEE National Aerospace and Electronics Conference (NAECON)*, 2015.

[14] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *EAI International Conference on Bio-inspired Information and Communications Technologies*

(formerly BIONETICS), 2016.
[15] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *International Journal of Security and Its Applications*, 2015.
[16] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "DL4MD: A Deep Learning Framework for Intelligent Malware Detection," in *International Conference on Data Mining* (DMIN), 2016.
[17] G. Tzortzis and A. Likas, "Deep belief networks for spam filtering," in *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 2007.
[18] G. Mi, Y. Gao, and Y. Tan, "Apply stacked auto-encoder to spam detection," in *International Conference in Swarm Intelligence*, 2015.
[19] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *IEEE International Conference on Computing, Networking and Communications (ICNC)*, 2014.
[20] S. Ranjan, *Machine learning based botnet detection using real-time extracted traffic features*, Google Patents, 2014.
[21] B. Rahbarinia, R. Perdisci, A. Lanzi, and K. Li, "Peerrush: mining for unwanted p2p traffic," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2013.
[22] A. Feizollah and e. al, "A study of machine learning classifiers for anomaly-based mobile botnet detection," in *Malaysian Journal of Computer Science*, 2013.
[23] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throwaway traffic to bots: detecting the rise of DGA-based malware," in *USENIX Security Symposium*, 2012.
[24] T. Chakraborty, F. Pierazzi, and V. Subrahmanian, "Ec2: Ensemble clustering and classification for predicting android malware families," *IEEE Transactions on Dependable and Secure Computing*, 2017.
[25] C. Annachhatre, T. H. Austin, and M. Stamp, "Hidden Markov models for malware classification," *Journal of Computer Virology and Hacking Techniques*, 2015.
[26] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Sethumadhavan, and S. Stolfo, "On the feasibility of online malware detection with performance counters," in *ACM SIGARCH Computer Architecture News*, 2013.
[27] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *ACM Proceedings of the Anti-Phishing Working Groups*, 2007.
[28] G. Xiang, J. Hong, C. P. Rose and, L. Cranor, "Cantina+: A feature-rich machine learning framework for detecting phishing web sites," *ACM Transactions on Information and System Security (TISSEC)*, 2011.
[29] G. Apruzzese, M. Marchetti, M. Colajanni, G. Gambigliani Zoccoli, and A. Guido, "Identifying malicious hosts involved in periodic communications," in *IEEE International Symposium on Network Computing and Applications (NCA)*, 2017.
[30] F. S. Tsai, "Network intrusion detection using association rules," *International Journal of Recent Trends in Engineering*, 2009.

[31] F. Bisio, S. Saeli, L. Pierangelo, D. Bernardi, A. Perotti, and D. Massa, "Real-time behavioral DGA detection through machine learning," in *IEEE International Carnahan Conference on Security Technology (ICCST)*, 2017.
[32] Y. Ye, D. Wang, T. Li, D. Ye, and Q. Jiang, "An intelligent PE-malware detection system based on association mining," *Journal in computer virology*, 2008.
[33] W.-F. Hsiao and T.-M. Chang, "An incremental cluster-based approach to spam filtering," *Expert Systems with Applications*, 2008.
[34] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Systems with Applications*, 2014.
[35] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic Analysis of Malware Behavior Using Machine Learning," *Journal of Computer Security*, 2011.
[36] H. S. Anderson, J. Woodbridge, and B. Filar, "DeepDGA: Adversarially-Tuned Domain Generation and Detection," in *ACM Workshop on Artificial Intelligence and Security*, 2016.
[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014.
[38] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Applications*, 2009.
[39] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: a survey," *Journal of Big Data*, 2015.
[40] A. Khan, B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for textdocuments classification," *Journal of advances in information technology*, 2010.

[41] B. Ingre, A. Yadav, and A. K. Soni, "Decision Tree Based Intrusion Detection System for NSL-KDD Dataset," in International Conference on Information and Communication Technology for Intelligent Systems., 2017, pp. 207–218.

[42] A. J. Malik and F. A. Khan, "A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection," Clust. Comput., no. 3, pp. 1–14, 2017.

[43] N. G. Relan and D. R. Patil, "Implementation of network intrusion detection system using variant of decision tree algorithm," in International Conference on Nascent Technologies in the Engineering Field, 2015

[44] K. Rai, M. Syamala, Devi Professor, and A. Guleria, “Decision Tree Based Algorithm for Intrusion Detection,” vol. 07, no. 4, pp. 2828–2834, 2016.

[45] M. Modinat, A. Abimbola, B. Abdullateef, and A. Opeyemi, “Gain Ratio and Decision Tree Classifier for Intrusion Detection,” Int. J. Comput. Appl., vol. 126, no. 11, pp. 975–8887, 2015

[46] C. Azad and V. K. Jha, “Genetic Algorithm to Solve the Problem of Small Disjunct In the Decision Tree Based Intrusion Detection System,” vol. 7, no. 8, pp. 56–71, 2015.

[47] S. Puthran and K. Shah, “Intrusion Detection Using Improved Decision Tree Algorithm with Binary and Quad Split,” in International Symposium on Security in Computing and Communication, 2016, pp. 427–438.

[48] A. O. R. G. Jimoh, “Anomaly Intrusion Detection Using an Hybrid Of Decision Tree And K-Nearest Neighbor,” vol. 2, no. 1, pp. 67–74, 2015.

[49] M. A. Iniesta-Bonillo, R. Sánchez-Fernández, and D. Jiménez-Castillo, “Sustainability, value, and satisfaction: Model testing and cross-validation in tourist destinations,” J. Bus. Res., vol. 69, no. 11, pp. 5002–5007, 2016.

[50] A. Ammar, “A Decision Tree Classifier for Intrusion Detection Priority Tagging,” J. Comput. Commun., vol. 3, no. 4, pp. 52–58, 2015.

[51] R. Selvi, S. S. Kumar, and A. Suresh, “An Intelligent Intrusion Detection System Using Average Manhattan Distance-based Decision Tree,” Adv. Intell. Syst. Comput., vol. 324, pp. 205–212, 2015.

[52] D. Moon, H. Im, I. Kim, and J. H. Park, “DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks,” J. Supercomput., vol. 73, no. 7, pp. 2881–2895, 2017.

[53] S. Jo, H. Sung, and B. Ahn, “A Comparative Study on the Performance of Intrusion Detection using Decision Tree and Artificial Neural Network Models,” vol. 11, no. 4, pp. 33–45, 2015.

[54] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, “A survey of deep learning-based network anomaly detection,” Clust. Comput., pp. 1– 13, 2017.

[55] Y. Ding, S. Chen, and J. Xu, “Application of Deep Belief Networks for opcode based malware detection,” in International Joint Conference on Neural Networks, 2016, pp. 3901–3908.

[56] M. Nadeem, O. Marshall, S. Singh, X. Fang, and X. Yuan, “SemiSupervised Deep Neural Network for Network Intrusion Detection,” KSU Proc. Cybersecurity Educ. Res. Pract., Oct. 2016