2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS) | 978-1-6654-4442-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/TELSIKS52058.2021.9606424

Next-Generation Network Intrusion Detection System (NG-NIDS)

Yazan Alnajjar, Jinane Mounsef

Abstract—This paper introduces the Next-Generation Network Intrusion Detection System (NG-NIDS) with intelligent capabilities based on the Artificial Neural Networks (ANN) and Machine Learning (ML) algorithms. The results have been achieved by training the model on a benign as well as malicious traffic. The proposed NG-NIDS achieved 99.9% accuracy of detecting the malicious traffic, which reflects the fact that this design is accurate and reliable.

***Keywords*—Anomaly detection, Behavioral network security, Intrusion detection, ANN cyber security, Artificial intelligence security.**

I. Introduction

Network Intrusion Detection System (NIDS) is essential in the network infrastructure components. It can be a device or software application that monitors a network or systems for malicious activity or policy violations [1]. Intrusion Detection systems falls under one of the following two categories, Host-Based Intrusion Detection System (HIDS) and Network-Based Intrusion Detection system (NIDS). The HIDS is a software version installed on the workstations to detect the compromise and malicious activities on the host level, while the NIDS is monitoring detecting the malicious traffic on the network level. The NIDS has two types, Anomaly-Based and Signature-Based detection [2], the most commonly used, as of today, is the Signature-Based version. The Anomaly-Based detection (NG-NIDS), which is the topic of this paper, is still not well known and few network infrastructure vendors started showing the interest in this area.

Secure Networks showed that attacks, which exploited fundamental TCP/IP problems insertion, evasion, and Denial-of-Service attacks, were able to elude NIDS detection. Dan Kaminsky recently showed he could send a series of fragmented packets to a NIDS that, based on the time and the operating system platform that they arrive at, reassemble into an attack for that platform that is not recognized by the NIDS [3].

The NIDS by itself provides alerts of compromise which can be fed into security information and event management (SIEM). The former is a subsection within the field of computer security, where software products and services combine security information management (SIM) and security event management (SEM). They provide real-time analysis of security alerts generated by applications and network hardware [4].

Yazan Alnajjar is with the Faculty of Electrical Engineering, University of Rochester Institute of Technology, Silicon Oasis, Dubai, United Arab Emirates (e-mail: ywa3420@rit.edu).

Jinane Mounsef is professor with the Faculty of Electrical Engineering, University of Rochester Institute of Technology, Silicon Oasis, Dubai, United Arab Emirates (e-mail: jmbcad@rit.edu).

SIEM systems can have thousands of correlation rules. Some of them are simple, and some are more complex. Once a correlation rule is triggered the system can take appropriate steps to mitigate from a cyber-attack. Usually, this includes sending a notification to a user and then possibly limiting or even shutting down the system.

The Correlation-Engine of the security information and event management system (SIEM) might be an interesting area for the Machine Learning (ML) applications, as the SIEM is centralizing the events and logs from all connected devices over the network, as it has the comprehensive picture of what is happening over the entire network deeply to the device level.

II. Related Work

Cyber-Security is a major concern for business owners and individuals. This is known in this domain as a priority triangle Availability, Integrity and Confidentiality (AIC), hence, this area has been an active area for research in the last few years, which helped to introduce the added value and the capabilities of the Artificial Intelligence (AI) tools in such a critical domain.

In [5], Jian and Goyal, from the Center of Development and Advance Computing in India, have proposed an adaptive intrusion detection system based on the immunity. Their model was built in such a way similar to the human immune system; it can detect the anomalies after performing the Deep Packet Inspection (DPI) for SSH attacks, then it generates a new signature to push it to the existing signatures-databases with new entries. The model achieved good results between 95% - 99% accuracy, however it defers from our proposed approach in the way it can handle the newly identified attacks. Our proposed method is curated for anomalies detection without the need to generate a new signature.

In [6], Sharafaldin et al. have prepared a testing lab environment with many workstations in order to collect a dataset while exciting several cyber-attacks. Their dataset has been used to train several Intrusion Detection System models. Their testing included the K-Nearest Neighbor algorithm (kNN), Multilayer Perceptron (MLP) and Random Forest (RF) algorithms. They have achieved relatively a very good accuracy between 96% - 98%. Their approach differs from our approach mainly on the model design, as we are using Artificial Neural Networks (ANN) to build the model, which results in a better performance and higher accuracy.

In [7], Islam and Ashiqur Rahman have explored the Anomaly Intrusion Detection System in Wireless Sensor Networks. Their study focused mainly on the physical layer intrusion detection, where they have considered the first layer in the OSI model during the research. The performed attacks

differ from our model, as they are applicable to the physical layer only. For instance, the Black-Hole attack is common over the wireless networks. As result of their study, they have defined 2 selective rules for the categorization. On the other hand, their model can be an additional integration to our model in order to detect the physical intrusion attempts, as well.

In [8], Ravale et al. have studied the Feature Selection Based Hybrid Anomaly Intrusion Detection System using K-Means and RBF-Kernel function. Their model achieved a high precision of 93.33% for DOS attacks.

In [9], Yang and Yilai investigated the Anomaly Intrusion Detection based on SVM. Their approach can select learning vector samples while training and effectively reduce the number of training samples and training time while getting a higher detection rate classifier with smaller samples dataset.

III. Use Cases and Domain of Work

A. Intrusion Detection System

The function of an Intrusion Detection System (IDS) is to detect the attempts that compromise the operation of a system. This will cause the system to operate in a manner that breaches the design requirements. This could take the form of a compromise to the confidentiality, availability and integrity of the system and could inappropriately control the stored data.

The traditional NIDS relies mainly on the signatures database [2]. Therefore, the user must keep the database updated, which is by itself a challenge for the Signature-Based (NIDS) that will not detect attacks without signature. Moreover, NIDS will not detect the variation of attacks after changing the signature of the attack.

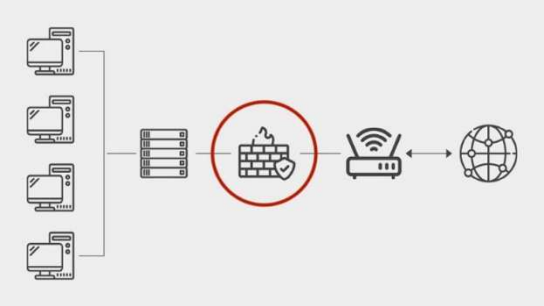

Fig. 1. Network Intrusion Detection System (NIDS).

The basic architecture and installation of the NIDS is normally as shown in **Fig**. 1, where it is configured as a passive device to monitor the inbound and outbound traffic over the network. The traditional NIDS is normally delivered as a built-in additional function in the commercial firewalls; however, some vendors still deliver it as a standalone device with tapping interface to capture the traffic.

B. Machine Learning (ML)

ML is the study of computer algorithms that improve automatically through experience [10]. It is seen as a subset of artificial intelligence (AI), where learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do.

For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step [11].

C. Anomaly Detection

In data mining, anomaly detection, also known as outlier detection, is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data [12]. Typically, the anomalous items represent an issue such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are referred to as outliers, novelties, noise, deviations and exceptions [13].

D. Artificial Neural Network (ANN)

Artificial neural networks (ANNs), are computing systems vaguely inspired by the biological neural networks that constitute brains [14].

ANN is based on a collection of connected units or nodes, called artificial neurons, which loosely model the neurons in a biological brain. Each connection can transmit a signal to other neurons. An artificial neuron receives a signal, then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by a non-linear function of the sum of its inputs [14].

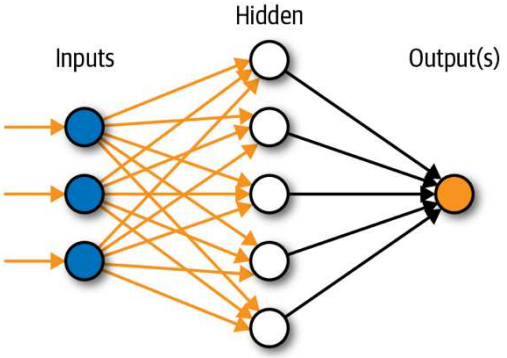

Fig. 2. Artifical Neural Network (ANN).

Fig. 2 shows a simplified ANN with one hidden layer, 3 inputs and 1 output. The hidden layer consists of 5 neurons which represent the weight values.

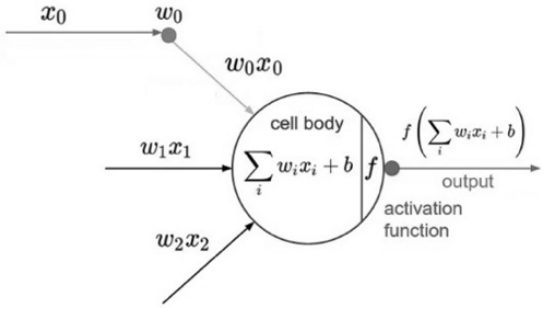

Fig. 3. Artificial neuron activation function.

Fig. 3 shows the artificial neuron activation function, which is calculated by multiplying the input values by the weight value of that specific neuron as shown in (1).

$$f = \sum_i w_i \cdot x_i + b \quad (1)$$

$\mathbf{w_i}$ corresponds to the weight, which is defined after training the ANN, $\mathbf{x_i}$ is the input, **b** is the bias and **f** corresponds to the activation function.

ANNs have been used in several computer security domains, including the analysis of software design flaws [15] and computer virus detection [16]. ANN approaches to detection of multiple types of network attacks have also been shown to be effective [17], though their application to the detection of shellcode was not considered.

This paper studies the Next-Generation Network Intrusion Detection System (NG-NIDS) based on training an ANN model, which will demonstrate the capabilities and the added value of applying ML tools in the cyber security domain.

IV. Dataset Used for Training, Testing and Validation

The availability of datasets related to this field is rare, as such datasets are internal and cannot be shared due to privacy issues. Also, as network behaviors are normally changing, in addition to intrusions evolution over the time, it has become necessary to move from static and one-time datasets towards more dynamically generated datasets, which not only reflect the traffic compositions and intrusions of that time but are also modifiable and extensible [6].

Training and testing dataset used for the proposed NG-NIDS model was generated by a researcher's team at University of New Brunswick (UNB) [6] as Pcap trace log, while exciting several attacks in different methods and several operating systems. These attacks include DDos, XSS, SQL Injection and Brute Force attacks.

Several tools have been used to execute these attacks. The table below lists some of the tools used for each attack type:

TABLE I
Tools used of each attack type from Kali Linux machine.

Attack Type	Tools Used
Brute Force Class-1	Hydra, Medusa, Ncrack, Metasploit, Nmap NSE scripts
XSS Class-2	Damn Vulnerable Web App (DVWA)
SQL Injection Class-3	Damn Vulnerable Web App (DVWA)
DDos Class-4	High Orbit Ion Cannon (HOIC)

In order to capture the traffic information while executing the attacks in Table I, CICFlowMeter-V3 (JAVA based tool) has been used to capture and extract 83 features from the monitored traffic. These features have been used later as an input the ANN, as explained in Section B.

To use this dataset for the proposed NG-NIDS model, it has been labeled for each attack type including the information about the workstations such as: Duration, Number of packets, Number of bytes, Length of packets in addition to other information captured at the same time.

The final dataset consists of 412,791 entries, which have been subdivided into 3 groups, as 60% training data, 20% validation data and 20% testing data.

V. Detecting Malicious Activities

A. Web Vulerabilities and Attacks

The majority of web application attacks occur through cross-site scripting (XSS) and SQL injection attacks [18], which typically are made possible by flawed coding and failure to sanitize application inputs and outputs.

According to the security vendor Cenzic [19], the top vulnerabilities in March 2012 include the following as shown in Fig. 4:

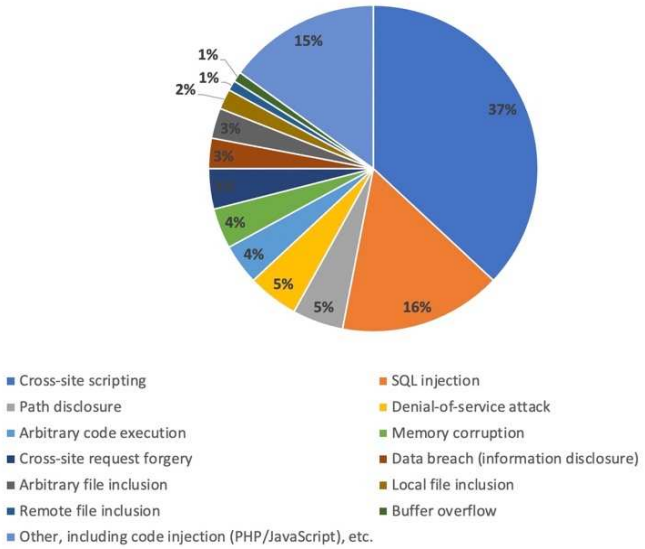

Fig. 4. The top vulnerabilities in March 2012 [19].

The ten most critical web application security risks include Injection, Broken authentication, Sensitive data exposure, XML external entities (XXE), Broken access control, Security misconfiguration, Cross-site scripting (XSS), Insecure deserialization, Using components with known vulnerabilities and Insufficient logging and monitoring [20].

B. Next-Generation NIDS Approach

This section will explain the proposed approach for the NG-NIDS based on the ANN and Anomaly-Based detection, This simulation has been carried out using MATLAB R2019b and the dataset [6] described in Section IV.

The model is designed to read 83 input fields for several information about the traffic, then feeding them to a hidden layer consisting of 40 neurons in order to predict traffic types and classify them. The designed model classifies the traffic into 5 categories, where classes 1 to 4 represent the malicious traffic and class 5 represents the benign traffic as follows:

- Class-1 - Brute Force
- Class-2 - XSS
- Class-3 - SQL Injection
- Class-4 - DDos
- Class-5 - Benign Artificial Neural

Fig. 5 shows the designed Artificial Neural Network (ANN) which has been used for the training and testing.

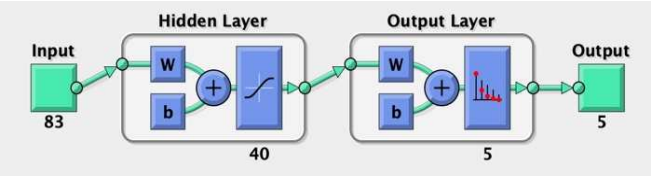

Fig. 5. Network design.

C. Results

The results are summarized in Table II and show a good Anomaly-Detection performance. Table II shows the sensitivity and precision of the designed model after testing the trained ANN on our test dataset.

TABLE II
MODEL QUALITY PARAMETERS FOR DETECTING MALICIOUS TRAFFIC

Attack Type	Quality Parameter	Value
Brute Force Class-1	Sensitivity	80.3 %
	Precision	82.9 %
XSS Class-2	Sensitivity	61.3 %
	Precision	51.5 %
SQL Injection Class-3	Sensitivity	99.7 %
	Precision	100 %
DDos Class-4	Sensitivity	100 %
	Precision	100 %

Based on the results shown in Table II, the model can detect almost all of the malicious traffic, despite that in some few cases, it did not classify them correctly in the right malicious traffic class. However, it was able to classify them as a malicious traffic.

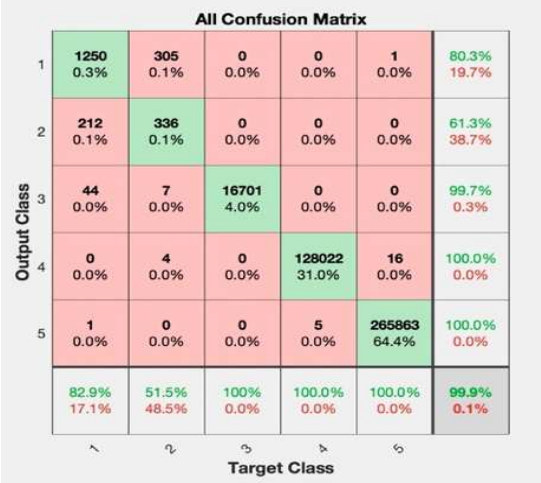

Fig. 6. Confusion matrix of the model.

Fig. 6 shows the confusion matrix for the trained model, which reflects a good performance to detect the malicious traffic. The overall accuracy of the model is 99.9%.

Such model makes a perfect integration with the security information and event management system (SIEM), where the SIEM has a complete overview over the network. Also, the proposed model can be configured to trigger a preventive action where it will not only detect the malicious traffic, but will also block it. Such system, known as Network Intrusion Prevention System (NIPS).

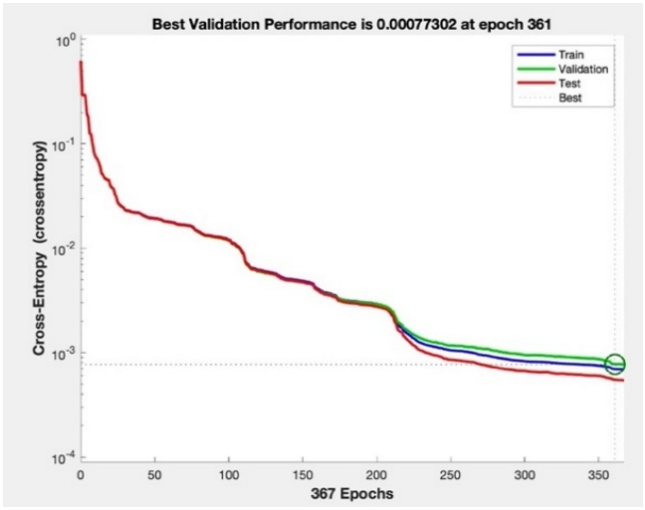

Fig. 7. Model Performance for 367 epoch.

Fig. 7 shows the network performance validation given targets and outputs. Minimizing cross-entropy leads to a good classifier. The obtained entropy is 0.00077302 at 361 epochs for the proposed model, where the cross-entropy is calculated as shown in (2):

$$CE = - T \cdot \log(y), \quad (2)$$

where T is the targeted output and y corresponds to the actual model output.

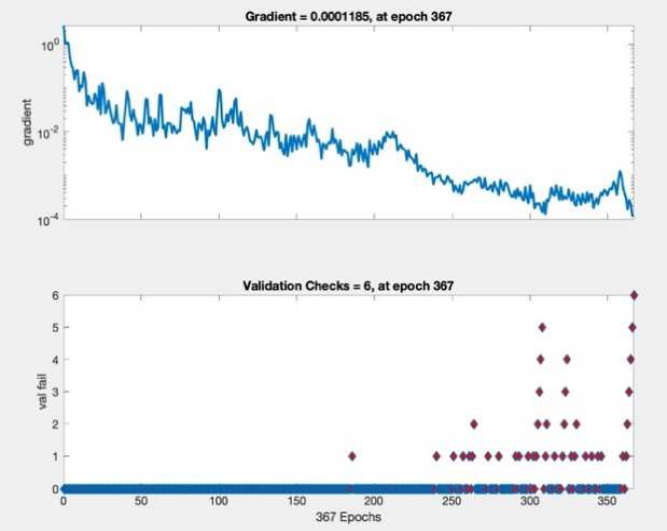

Fig. 8. Gradiant value change during training.

Fig. 8 shows the variation in gradient coefficient with respect to the number of epochs. The final value of the gradient coefficient is 0.0001185 at epoch 367. Minimizing the gradient coefficient reflects a better ANN performance.

Fig. 9 shows the Receiver Operator Characteristic (ROC) of the model, which is a metric used to check the quality of classifiers. It represents the ability of the model to differentiate between classes. The ROC curve can be plotted by defining the True Positive Rate (TPR), which is the correct positive among all positive samples and the False Positive Rate (FPR) which is the incorrect positive results occur among all negative samples. Maximizing the area below the curve represents a higher model quality. Fig. 9 illustrates a high model performance, which is also observed by calculating the model accuracy of 99.9%, as shown in the confusion matrix (Fig. 6).

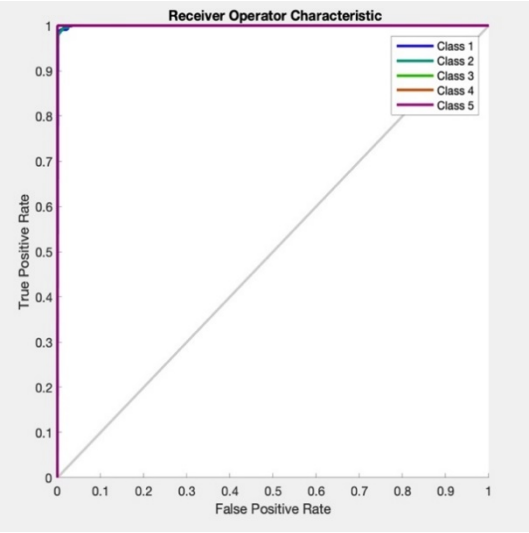

Fig. 9. Reciver Operator Charachteristic (ROC).

VI. Conclusion

NIDS are essential devices for modern network infrastructure. They are used to alarm and notify network administrators (NA), information security department (ISD) and the incident response team (IRT) when a cyber-attack is detected. However, the existing modifications to traditional cyber-attacks can evade the NIDS easily. On the other hand, the threat vector is becoming more critical, as the attackers are continually creating new tools and improving their existing ones. At the end, the hacker does not have to be highly skilled to make a minor change in the hacking tools in order to change the generated signature.

The proposed Anomaly-Based detection has demonstrated a high efficiency to detect the attacks without having to monitor the signatures with a high accuracy of 99.9%. This can reduce the risks of the latest critical vulnerabilities, which is also known as Zero-Day Vulnerability, where the equipment manufacturers did not release yet to date the official patch to fill the gap.

Network Security is crucial for system owners and users, where data privacy and network integrity are critical decision factors for most industries. Recently, the digital evolution has drawn the attention of the industrial domain and the critical infrastructure, which is normally operating on offline networks with zero-trust connectivity within the same organization. Although the digital evolution helped these industries to interact with the outside world and use the latest available technologies in the market in their domain, it also brought a high Cyber-Security concern, which is still a crucial part for the system integrity and safety.

Anomaly-Based detection works better with more information provided about the network. In other words, more information collected will result in a better trained model and more efficient ANN. The security information and event management system (SIEM) can take this approach to the next level. SIEM normally centralizes the logs and events for every component in the network (Switches, Routers, Firewalls Workstations, Servers, …) and therefore, the same concept can even provide a better protection performance despite that the NIDS are still required with prevention functions to trigger the firewall and block certain traffic.

The challenge that might be faced for such a model is based on the way the ANN normally works, as the ANN for the NG-NIDS is similar to a black-box for the end-user. In other words, some users prefer the full view of how a certain device works. However, in most cases, the proof-in-use plays a great decision factor.

We cannot rely on training the ANN one time only, as the proposed NG-NIDS model has to be retrained frequently with the new updated datasets, especially for the dynamic network environments, for example, modifying the network by adding or removing workstations, switches and routers, which will require retraining for the model in order to adapt and update the parameters of the NG-NIDS.

In conclusion, a future improvement of the proposed model can introduce another hidden layer, where the first layer can run in offline to keep training the ANN with new actual collected data captured from the live traffic, while the second layer will use the optimized parameters generated by the fist layer and apply them to the actual live traffic. This approach might help in many factors, as it can reduce the downtime for the NG-NIDS, while it is being trained with a new dataset. This can be essential for production network users. Another advantage will be to minimize the manual human intervention to configure such devices, especially for the dynamic network where devices are being added or removed from the network.

References

[1] T. Mitchell, *Machine Learning*, New York, 1997.

[2] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion Detection System: A Comprehensive Review”, *J. Netw. Comput. Appl*., vol. 36, no. 1, pp. 16-24, 2013.

[3] T. H. Ptacek, and T. N. Newsham, *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*, Secure Networks, Inc., Calgary Alberta, 1998.

[4] C. Kubecka, *28c3: Security Log Visualization with a Correlation Engine*, 2011.

[5] P. Jain, and S. Goyal, “An Adaptive Intrusion Prevention System Based on Immunity”, *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, pp. 759-763, 2009.

[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorba-ni, *Intrusion Detection Evaluation Dataset (CIC-IDS2017)*, Canadian Institute of Cybersecurity, 2018.

[7] M. S. Islam and S. A. Rahman, “Anomaly Intrusion Detection System in Wireless Sensor Networks: Security Threats and Existing Approaches”, *International Journal of Advanced Science and Technology*, vol. 36, no. 1, pp. 1-8, 2011.

[8] U. Ravale, N. Marathe, and P. Padiya, “Feature Selection Based Hybrid Anomaly Intrusion Detection System using K Means and RBF Kernel

Function", *Procedia Computer Science*, vol. 45, no. 39, pp. 428-435, 2015.
[9] Y. Xie and Y. Zhang, "An Intelligent Anomaly Analysis for Intrusion Detection Based on SVM", *International Conference on Computer Science and Information Processing (CSIP)*, pp. 739-742, 2012.
[10] "ANN", ANN - File Exchange - MATLAB Central. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/976-ann. [Accessed: 15-Dec-2020].
[11] E. Alpaydin, *Introduction to Machine Learning*, (Fourth ed.), MIT, pp. 13-18, 2020.
[12] A. Zimek and E. Schubert, *Outlier Detection, Encyclopedia of Database Systems*, Springer New York, pp. 1-5, 2017.
[13] V. J. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies", *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85-126, 2004.
[14] Y.-Y. Chen, Y.-H. Lin, C.-C. Kung, M.-H. Chung, I.-H. Yen, "Design and Implementation of Cloud Analytics-Assisted Smart Power Meters Considering Advanced Artificial Intelligence as Edge Analytics in Demand-Side Management for Smart Homes", *Sensors*, vol. 19, no. 9, 2019.
[15] A. Adebiyi, J. Arreymbi, and C. Imafidon, „A Neural Network Based Security Tool for Analyzing Software", *Doctoral Conference on Computing, Electrical and Industrial Systems*, Springer, pp. 80-87, 2013.
[16] G. Liu, F. Hu, анд W. Chen, „A Neural Network Ensemble Based Method for Detecting Computer Virus", *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, vol. 1, pp. 391-393, 2010.
[17] J. Wu, D. Peng, Z. Li, L. Zhao, and H. Ling, „Network Intrusion Detection Based on a General Regression Neural Network Optimized by an Improved Artificial Immune Algorithm", *PLOS ONE*, vol. 10, no. 3, pp. 1-13, 2015.
[18] J. Fonseca, M. Vieira, and H. Madeira, „Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks", *Dependable Computing*, 2007.
[19] Cenzic, Inc., 2012 Trends Report: Application Security Risks, 11 March 2012.
[20] Open Web Application Security Project, OWASP Top 10 - 2017: The Ten Most Critical Web Application Security Risks, 2017.