European Journal of Research Development and Sustainability (EJRDS)
Available Online at: https://www.scholarzest.com
Vol. 3 No. 12, December 2022
ISSN: 2660-5570

AINTRUSION DETECTION SYSTEM IN SCADA NETWORK USING MACHINE LEARNING

Hadeel Ali Kadhim
Department of Electrical and computer enginerning
Altinbas University
Istanbul, Turkey
203720154@ogr.altinbas.edu.tr
Abdullahi Abdu Ibrahim
Department of Electrical and computer enginerning
Altinbas University
Istanbul, Turkey
abdullahi.ibrahim@altinbas.edu.tr

Article history:
Received: 6th October 2022
Accepted: 6th November 2022
Published: 14th December 2022

Abstract:
Machine learning is a branch of artificial intelligence based on the idea that systems can learn to identify patterns and make decisions with a minimum of human intervention. In this study, demonstration learning will be used, using neural networks in a prototype of a drone built to perform trajectories in controlled environments. To accelerate the training convergence process, a new training data selection approach has been introduced, which picks data from the experience pool based on priority instead of randomness. An autonomous maneuver strategy for dual-UAV olive formation air warfare is provided, which makes use of UAV capabilities such as obstacle avoidance, formation, and confrontation to maximize the effectiveness of the attack.

Keywords: UAV, ML, DL, ANN

I. INTRODUCTION

A secure computer system must take care to comply with the pillars of cybersecurity, which are: Confidentiality, Integrity and Availability, as shown in Figure 1 [4]. Considering that in SCADA systems, the dominant paradigm since its inception has been the availability of information [2], the other aspects have been neglected, giving rise to a scenario where many of the industrial communications lack encryption and as if it were not enough supervision and control equipment run under obsolete software versions, highly vulnerable to all kinds of malware

Although several techniques and algorithms have been reported in the literature for intrusion detection; most of them can only identify previously known failure patterns using rule-based methodologies; in other solutions that use machine learning algorithms, these are restricted to a certain type of communication protocol. On the other hand, to the best of our knowledge, all the work carried out focuses on the analysis of network traffic, ignoring the internal behavior of the variables within the control devices and therefore the media effects that these could have on the process. Industrial. In SCADA systems, two types of variables are those that allow remote monitoring and manipulation of a plant. The first type is known as supervisory variables and their purpose is to transport information from the plant to the SCADA client. The second type are known as control variables and their purpose is to be able to execute orders from the client application. The proposed solution is aimed at achieving intrusion detection; For this, it must have the ability to recognize the normal behavior patterns of a SCADA system, in such a way that, by identifying an atypical pattern in the behavior of the variables of the control devices, a possible cyberfailure can be recognized

Fig.1.Comparison of malware activity to other cyber threats.

An article published in 2011with the title of "Adware detection and privacy control in mobile devices" by Ideses et .al, tells that the first computer virus was developed for a demonstration by Leonard Adleman (one of the developers of the algorithm RSA1) at a computer security seminar.

In 2011, Cisco publishes an article in which it says that the use of the internet would quadruple until 2015, driven mainly by the growing number of users of mobile devices (tablets and smartphones). The increase in bandwidth, allowing for higher speeds and ease of access, are relevant contributions to this growth. In this scenario, it is essential to create methodologies that allow a correct classification of network malware. Many network management tasks, such as workload characterization, capacity planning, route provisioning, network malware modeling and monitoring depends on the identification and classification of that same malware. Today there is a considerable number of applications that use the internet and that number has been increasing steadily. Multimedia applications, social networks, Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) programs, among others, have become essential in the business world, both at a private and corporate level. In this sense, anomalies in network malware can cause serious problems in their operation, being one of the main concerns for entities that depend on the transmission of data for their businesses. The identification and elimination of these anomalies must be done as quickly as possible or there is a risk of high losses as a result of these anomalies. [Jerome et.al] noticed the changes in the characteristics of the network, such as the increase in the transmitted quantity of packets, is one of the specific types of anomalies and can be caused by several factors, such as physical problems or techniques on the network, such as power cuts or inadequate equipment configurations, up to intentional malicious behavior, such as malicious software failures or unauthorized access attempts. That is why it is essential to find effective network malware classification methods, in order to ensure a quality and at the same time secure service. [Hwan-Hee et.al] explained the main problem in dealing with the analysis of malware on a large-scale network. Techniques for capturing packets and analyzing their contents are generally used to generate statistical data and detect anomalous events. But in a large network that has a considerable number of interconnected devices, this technique becomes unviable, due to the fact that a large processing capacity and space are required to store the data. Traditional classification techniques, such as those based on communication port numbers or packet payload analysis, are also not effective for all types of network malware [Kurniawan] et.al Due to its nature and the fundamentals of many other techniques, the field of qualification of network malware has maintained a great interest on the part of many authors. For example [Lindorfer et.al] mentions in their work that a common technique for identifying network applications on the Internet is through the monitoring of malware based on usage. to better known communication ports, making an analysis of the packet headers to identify malware that is associated with a particular port and, therefore, of an application in [Mas'ud et.al] This process can lead to inaccurate estimates due to the amount of malware carried out by the most diverse applications, since protocols such as HyperText Transfer Protocol Secure (HTTPS), are often used to relay other types of malware , for example, a Virtual Local Area Network (VLAN) over HyperText Transfer Protocol (HTTP). On the other hand, some point-to-point applications avoid using known communication ports, in an attempt to circumvent any block imposed by the system, which can cause even more inaccuracies in the estimates. The captured breath, finding similar patterns that can be used for the classification of some anomaly, allows to create a base with data of comparison with the information resulting from this analysis. [Aziz et.al] refers to the concept of anomaly detection using entropy, where Shannon's entropy is cited. This type of detection and classification is done through the entropy calculation for the following categories: source port, destination port, IP address, source and IP address , destination, from the correlation between them. Entropy values can be compared to classify abnormal behaviors in the analyzed malware [Xiaoyan et.al] During the research carried out for this work, it was detected that there is a relationship between the malware class and its statistical properties that provide information on the distribution of the byte flow of packages and the output of a specific number of specific applications.

[John et.al] also observed that the relevance of the packages used in the studies carried out by [Westyarian et.al] tend to a certain profile size, the observation of this profile allows a more precise selection of the three relevance to these studies. [Aziz et.al] proposed a classification scheme should allow for the description of representative malware

patterns. shows our version of the steps to follow, which serve as guides for carrying out this process. In the area of security, using this information is crucial in identifying anomalies in the network system. Our goal is to understand how the characterization and detection of anomalies in network malware is done

II. SCADA SECURITY SYSTEM

A. *PROPOSED MODEL*

Malicious content on any network is normally detected through several features, on of the most apparent feature is the extensive exchange in information between users, This exchange of information, called network traffic, is carried out with the transmission of a sequence of Transmission Control Protocol, This traffic can be considered normal or anomalous depending on certain parameters such as the high number of bytes transmitted, or a request for access to an unauthorized machine, etc. we propose a feature extraction method that extracts the features of normal activity on SCADA, and abnormal activity in order to feed a classification system, the classification scheme that classifies the activity into normal and abnormal in accordance with the data provided by the feature extraction phase

B. *ATTACK MODEL*

Several classifications of attacks have been proposed according to several criteria such as the severity of the attack, the vulnerabilities exploited by the latter, and the way to proceed. The classification most used in the literature is that adopted by the Massachusetts Institute of Technology (MIT) [21], which includes five classes of attacks that we will present in what follows.

Denial Of Service DOS

A denial of service is an attack in which the hacker makes certain processing or storage resources unavailable or too busy to be able to respond to the requests of legitimate users of a machine. There are several types of denial of service (DOS) attacks [52]. Some attacks like "smurf" perfectly abuse legitimate devices. Other "ping of death" creates malformed packets that confuse the TCP/IP stack of the target machine, the latter will try to reconstruct these packets afterwards. Still other "apache2, dos, syslogd" take advantage of bugs in the network.

User to Administrator Attacks (User To Root U2R)

Are a class of exploit in which the attacker starts with access to a normal user account on the system (obtained by sniffing passwords, using e.g. social engineering), then tries to exploit the vulnerability on this system to gain administrator access. There are several types of U2R attacks [35]. The most common is the "buffer overflow" attack which occurs when a program copies a lot of data into a static buffer memory area without checking if the size of the latter is sufficient. , which will cause an overflow. The overflowed data will be stored in the system stack, thus covering the next instructions that were to be executed. By carefully handling data that overflows onto the stack, an attacker can cause the execution of some arbitrary commands by the operating system that will help it get what it is looking for. Another class of U2R attacks exploits programs that give insight into the environment in which they are running, a good example of such an attack is the "load module" attack. Another class of U2R attacks exploits programs that have poor handling of temporary files. Some U2R attacks use vulnerabilities due to exploitable competitive conditions involving the execution of a single program, two or more programs running simultaneously [25]. Although controlled programming could eliminate all of these vulnerabilities, such bugs are present in every version of UNIX and Microsoft Windows available today. exploitation that will help him get what he is looking for. Another class of U2R attacks exploits programs that give insight into the environment in which they are running, a good example of such an attack is the "load module" attack. Another class of U2R attacks exploits programs that have poor handling of temporary files. Some U2R attacks use vulnerabilities due to exploitable competitive conditions involving the execution of a single program, two or more programs running simultaneously [25]. Although controlled programming could eliminate all of these vulnerabilities, such bugs are present in every version of UNIX and Microsoft Windows available today. exploitation that will help him get what he is looking for. Another class of U2R attacks exploits programs that give insight into the environment in which they are running, a good example of such an attack is the "load module" attack.

Remote To Local R2L Attacks

An R2L attack occurs when a hacker who can send packets to a machine across a network but does not have an account on that machine exploits a vulnerability to gain local user access to that machine. There are several ways in which an attacker can achieve his objective [Kendall, 99]. Some attacks exploit buffer overflow caused by network server software "imap, named, sendmail". The "dictionary, ftp-write, guest and xsnoop" attacks try to exploit weak or misconfigured system security policies. The "xlock" attack uses social engineering, to succeed the attacker must parody human operators to provide their passwords to screensavers which are actually Trojan horses.

Probe Attacks

In recent years, an increasing number of programs have been distributed that can automatically scan a network of computers to gather useful information or to find known vulnerabilities [36]. These network probes are quite useful for a hacker planning a future attack. An attacker with a map of the machines and services available on a network can use this information to find any weak points in the network. Some of these "satan, saint, mscan" scanning tools allow even a novice hacker to examine hundreds or thousands of machines on a network very quickly.

Data Attacks

Data attacks involve someone (user or administrator) who performs certain actions, which he has the possibility to perform on a given workstation, but according to the security policy of the site where he operates, he does not have the permissions sufficient to do so. Often these attacks involve the transfer of secret data files to or from sources where they do not belong

C. DETECTION MODEL

The classification of network traffic (IP packets) is far from new. The increasing number of attacks and the emergence of new threats are all factors that have favored its use. Classification in intrusion detection consists in separating the data collected on a network into several distinct classes, according to their content. Each class therefore groups packages with similar content. The main purpose of this task is to make the IDS able to autonomously determine packets related to an attack class. As shown in the following figure:

1) What is a classifier

A classifier is a set of equations or logical rules built from the labeled input data available from the database known as the learning base. In the case of attack classification, the examples provided to the machine are in the form (Attack, Class). This method of reasoning is called inductive because we induce knowledge (the model) from the input data (the packets) and the outputs (their classes). Thanks to this model, we can then deduce the classes of new data: the model is used to predict. The model is good if it predicts well.

2) Training of learning algorithms in classification

The training of a learning algorithm in classification contains three phases: the initial preparation of the data, the learning and the validation [31].

3) Initial data preparation

Good data preparation is the key preliminary step in a KDD (Knowledge Discovery in Databases) process. Data preparation takes about 60-80% of the time spent in the data extraction process [13]. In our case, this first phase consists of finding an attack sample types which must be representative of the kind of data to be classified.

- Eliminate duplicates and input errors: Redundant data can be troublesome because they will give more importance to repeated values. Conversely, an input error may obscure a repetition.
- Check domain integrity: A check on the domains of values makes it possible to find outliers.
- Process missing information: This is the term used to designate the case where fields contain no data. Sometimes it is interesting to keep these records because the absence of information can be information. On the other hand, the values contained in the other fields may also be useful.

These are the values of the fields in the data records. These data have a type that must be specified. Indeed, most of the methods are sensitive to the manipulated data. For example, some methods are faulted by continuous data while others may be sensitive to the presence of discrete data:

Discrete data: binary or logical data: 0 or 1; Yes or no ; true or false, are good customer data. Enumerative data are discrete data for which there is no a priori defined order, for example: color.

Ordered enumerative data:concern responses to an opinion survey data from the discretization of continuous data

Continuous data:these are integer or real data: age, average income, ... but also data that can take a large number of ordered values. For data mining applications, it is common to transform them into continuous data or ordered enumerative data. We transform a date of birth into an integer age or an ordered enumerative variable corresponding to age groups.

Text data:a text can, for certain applications, be summarized as a NUplet consisting of the number of occurrences in the text of keywords from a predefined dictionary. As they can also be transformed into whole data.

The data preparation stage can include other phases such as data selection which consists in obtaining data in accordance with the objectives that we impose on ourselves and the enrichment which results in the addition of new fields while preserving often the same number of records (object).

a) Data learning or training

Using the training set, the algorithm learns which kinds of attacks end up in which classes.

b) Improvement

In this phase the classifier tries to properly label the data belonging to the validation set. The classification obtained is compared with that previously established to deduce the empirical error.[1]. If the results are conclusive, the classifier is used to process new cases whose classification is unknown. The validation set can be obtained either:

The simplest, by separating the data set D into a training set Da and a validation set Dv. The algorithm is thus trained with the training set and the empirical error is measured on the validation set.

Or, when there is not enough data to separate them into two sets, using cross-validation which is a more reliable approximation. it is necessary to train the algorithm on all the data except those contained in Di, and to measure the empirical error on the set Di. It is therefore necessary to repeat the learning cycle k times, and then to use the average of the empirical errors.

Figure 2 Steps of the proposed classification process

III. EXPERIMENTS

In this section, results obtained using the procedures discussed in the previous chapter are analyzed extensively. Some important performance criteria that were made reference to are: Mean Square Error which is the average squared difference between our estimated values and actual value. MSE can be described as a measure of quality an estimator gives, it is always a non-negative value and values closer to zero are more reliable. Another criterion is the number of epochs which describes the number of passes completed when training through the given dataset. Our feedforward neural network uses iterative algorithms that require as many epochs during training or learning process. Time of training represents the total amount of time a feedforward neural network training algorithm needs to complete one training process.

A. *A. Selection of Optimal Training Algorithm*

There is no specific formula for choosing a training algorithm. There are various factors to consider in choosing an optimal algorithm such as; complexity of the problem, the samples available, accuracy, computation time and resources, and whether the network is used for regression analysis or discriminant analysis that estimates dependent and independent variable relationships. These various properties associated with these training algorithms determines if the training function used is fast, whether it can be used for really complex problems and whether it requires abundant memory.

Figure 4.6 displays the performance plot of the network using the Resilient Back propagation (RP) training function. Training was completed in after 13 epochs, a Mean Square Error (MSE) of 0.4993 was generated.

Figure 1 to 6, shows the performance plot of using the training functions mentioned earlier, from figure 1 it can be seen that Levenberg-Marquardt (LM) gives the better optimization result. The Levenberg-Marquardt (LM) was used as our optimal training function because of its speed, and it does not require large computing and processing memory.

B. *B. Determination of Optimum Network Architecture*

After determining the optimal training function needed for our network, modifications are made to the number of neurons in the hidden layer of our network architecture, to compare and analyse the optimal architecture needed to generate efficient results. In this section we will determine the optimal number of neurons required in the hidden layer to attain an optimal network for training. Different architecture was considered with varying number of neurons, also it is known that an increase in the number of hidden neurons is likely to cause overfitting

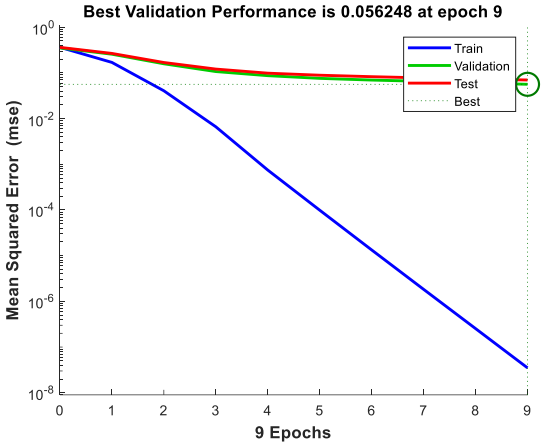

Figure 1: Performance Plot of {48-55-24} Network with Training Function Trainlm.

Figure 1 displays the performance plot of the network using optimal training function and 55 hidden layer neurons. Training was completed in 55 seconds at 9 Epochs, validation curve shows a validation performance value of 0.056248 at 9 Epochs. Mean Square Error (MSE) of 0.0252 was generated.

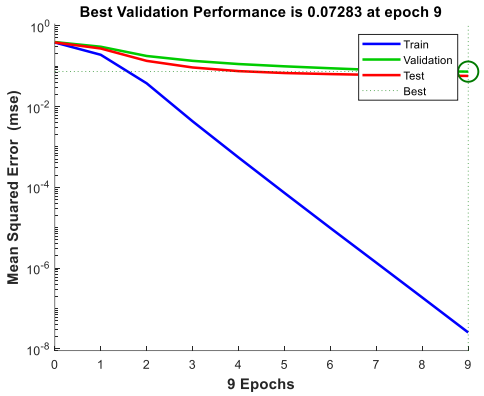

Figure 2: Performance Plot of {48-60-24} Network with Training Function Trainlm.

Figure 2 displays the performance plot of the network using optimal training function and 60 hidden layer neurons. Training was completed in 56 seconds at 9 Epochs, validation curve shows a validation performance value of 0.07283 at 9 Epochs. Mean Square Error (MSE) of 0.0260 was generated.

Table 1: Results of Different Network Architecture

Number of Hidden neurons	MSE	Epochs	Time(sec)
5	0.2184	12	1
8	0.1888	12	2
10	0.1562	12	2
15	0.1062	11	8
20	0.0706	13	10
25	0.0631	17	21
30	0.0563	11	17
35	0.0360	11	25
40	0.0336	11	34
45	0.0290	10	38
50	0.0289	9	42
55	0.0252	9	55
60	0.0260	9	56

From table 1 the following observations have been made from comparing the various values for performance criteria:

- The number of epochs becomes constant at a value of 9 epochs after 50 neurons.
- Mean Square Error (MSE) reduces as the number of hidden neurons increases.
- The time taken increases with increase in the number of neurons.

- It is obvious that the network topology with 50 hidden neurons in the hidden layer has the better optimization results among the four modifications even if it takes 42 seconds to train.

C. C. Testing Optimality Of {48-50-24} Network Architecture

The trained network displays a performance criterion known as the Mean Square Error which can be measured with respect to our dataset testing samples. It serves as a measure of how efficient the network will perform with real world datasets. The receiver operating characteristic plot (ROC) is another way to determine how well the neural network has matched the data samples. The relationship between false positive and true positive concentrations is established when the performance limit ranges from 0 to 1When markers appear farther left and up, they represent the fewer false positives that need to be accepted to generate a high true positive rate. The best classifiers are characterized by markers going from the bottom left corner, to top left corner, to the top right corner, or something close to that.

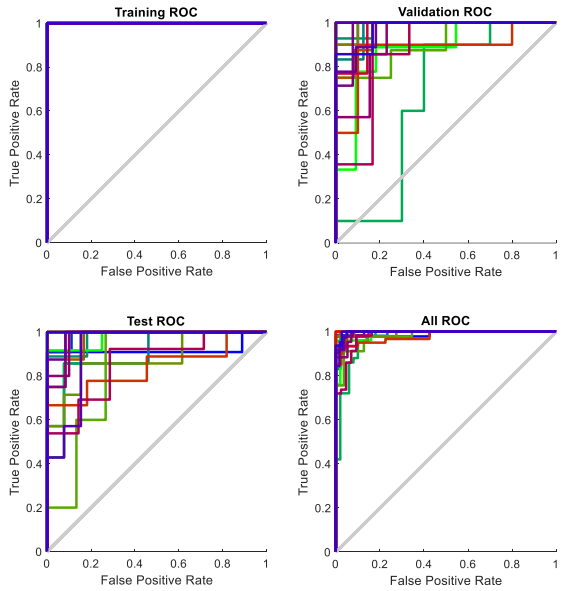

Figure 3: ROC Plot of {48-50-24} Network

Figure 3 shows that the neural network architecture is efficient.

1) D. VALIDATION OF {48-50-24} NETWORK ARCHITECTURE

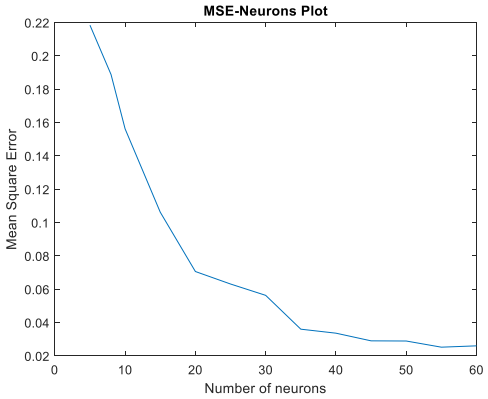

Figure 4: Graph showing relationship between MSE and Number of hidden layer neurons.

Figure 4 shows that the Mean Square Error monotonically reduces as the number of neurons in the hidden layer increases. This validates the network topology with 50 hidden neurons in the hidden layer as the better optimization results.

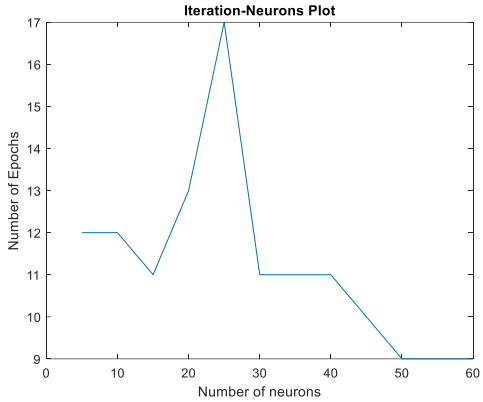

Figure 5: Graph showing relationship between Number of epochs and Number of hidden layer neurons.

Figure 5displays the maximum iteration reached for each modification of number of neurons in the hidden layer. From the figure 25 neurons modification has the highest iteration at 17 epochs while the 50 neurons modification stopped after 9 epochs. This validates the network topology with 50 hidden neurons in the hidden layer as the optimal network architecture.

D. *E. Performance Evaluation*

The performance of the overall system was evaluated by determining overall accuracy, false-positive rate (FPR) and false-negative rate (FNR). The Positive and Negative labels are compared with predicted labels gotten from the neural network classification for evaluating performance. The classification produces four outcomes:

- True positive (TP): This outcome represents a registered MAC address and the network classifies it as positive prediction
- False positive (FP): This outcome represents an attacker's MAC address and the network classifies it as positive prediction
- True negative (TN): This outcome represents an attacker's MAC address and the network classifies it as negative prediction
- False negative (FN): This outcome represents a registered MAC address and the network classifies it as negative prediction

Accuracy is calculated using:

$$\frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

False-positive rate (FPR)

$$1-\frac{TN}{TN+FP} \quad (2)$$

False-negative rate (FNR)

$$1-\frac{TP}{TP+FN} \quad (3)$$

Table 2 shows four scenarios and their corresponding TP, FP, TN, FN values with 20 sampled MAC addresses in each case.

Table 2: Scenarios and corresponding TP, FP, TN, FN values

Evaluation metric	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**
TP	14	15	14	17
FP	2	2	3	1
FN	3	3	3	1
TN	1	0	0	1

The results from these evaluation metrics performed on four scenarios can be found in table 4.3.

Table 3: Result generated from evaluation metrics.

Dataset	Accuracy	FPR	FNR
1	75.00%	66.66%	17.65%
2	75.00%	100.00%	16.67%
3	70.00%	100.00%	17.70%
4	90.00%	50.00%	5.56%
Average	77.50%	79.20%	14.40%

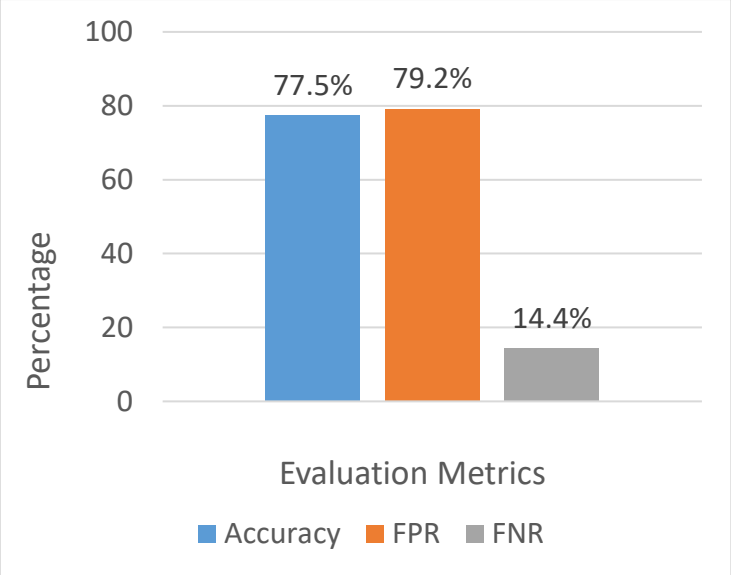

Figure 6: Bar graph representing various evaluation metrics

From table 3and figure 6, the system achieves an overall accuracy of 77.50%. The false positive rate has a value of 79.20%, while the false negative rate value is seen to be 14.40%. The value of the FNR is quite low which is a sign of the effectiveness of our proposed system, this also implies that our proposed system misclassifies unauthorized MAC addresses by a small percentage of 14.40. This in no way exposes the network to attacks

IV. CONCLUSION

The paper was aimed at designing a network authentication mechanism that can detect and prevent cyber-attacks from unauthorized users irrespective of the penetration tool used. In other to achieve our aim we collected MAC Address from trusted sources to ensure the authenticity of our input data and was able to train the neural network to detect the false MAC address using the neural network toolbox (nntool) on MATLAB undergoing different algorithms as explained above. From the results gotten from the training, we can infer that MAC address plays an important role in the securing of our wireless communication

V. REFERENECES

[1] P. Radoglou-Grammatikis, P. Sarigiannidis, I. Giannoulakis, E. Kafetzakis, E. Panaousis, Attacking IEC-60870-5-104 SCADA systems, in: 2019 IEEE World Congress on Services (SERVICES), 2642-939X, 2019, pp. 41–46, doi: 10.1109/ SERVICES.2019.0 0 022 .

[2] A. Elgargouri, M. Elmusrati, Analysis of cyber-attacks on IEC 61850 networks, in: 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT), 2017, pp. 1–4, doi: 10.1109/ICAICT.2017. 86 86 894 .

[3] RISI, Risi online incident database, 2015, http://www.risidata.com/Database .

[4] R.J. Robles , M. kyu Choi , E. suk Cho , S. soo Kim , G.-c. P , S.-S. Yeo , Vulnerabil- ities in SCADA and critical infrastructure systems, Int. J. Future Gener. Com- mun. Netw. 1 (1) (2008) 99–104 .

[5] M. Yampolskiy, P. Horvath, X.D. Koutsoukos, Y. Xue, J. Sztipanovits, Taxon- omy for description of cross-domain attacks on CPS, in: Proceedings of the 2Nd ACM International Conference on High Confidence Networked Systems, in: HiCoNS '13, ACM, New York, NY, USA, 2013, pp. 135–142, doi: 10.1145/ 2461446.2461465 .

[6] T.M. Chen, S. Abu-Nimeh, Lessons from stuxnet, Computer 44 (4) (2011) 91– 93, doi: 10.1109/MC.2011.115 .

[7] R. Langner, Stuxnet: dissecting a cyberwarfare weapon, IEEE Secur. Privacy 9 (3) (2011) 49–51, doi: 10.1109/MSP.2011.67 .
[8] R.M. Lee , M.J. Assante , T. Conway , German steel mill cyber attack, Ind. Control Syst. (2014) 1–15 .
[9] S. Trisal, 3 cyber attacks that rocked indus- trial control systems, 2017, https://cyware.com/news/ 3- cyber- attacks- that- rocked- industrial- control- systems- 817fee48 .
[10]D. Bisson, 3 ICS security incidents that rocked 2016 and what we should learn from them, 2016, https://www.tripwire.com/state- of- security/ ics- security/3- ics- security- incidents- rocked- 2016- learn/ .
[11]I.T. Laboratory, National vulnerability database, https://nvd.nist.gov/general .
[12]G. Yadav , K. Paul , Assessment of SCADA system vulnerabilities, in: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automa- tion (ETFA), 2019, pp. 1737–1744 .
[13]M. Henrie, Cyber security risk management in the SCADA critical infrastruc- ture environment, Eng. Manag. J. 25 (2) (2013) 38–45, doi: 10.1080/10429247. 2013.11431973 .
[14]R.I. Ogie, R. I., Cyber security incidents on critical infrastructure and indus- trial networks, in: Proceedings of the 9th International Conference on Com- puter and Automation Engineering - ICCAE '17, 2017, pp. 254–258, doi: 10. 1145/3057039.3057076 .
[15]D. Upadhyay, S. Sampalli, SCADA (supervisory control and data acquisition) systems: Vulnerability assessment and security recommendations, Comput. Secur. 89 (2020) 101666, doi: 10.1016/j.cose.2019.101666 .
[16]E. Luiijf, M. Ali, A. Zielstra, Assessing and improving SCADA security in the dutch drinking water sector, Int. J. Crit. Infrastruct.Prot. 4 (3) (2011) 124–134, doi: 10.1016/j.ijcip.2011.08.002 .
[17]C.-R. Chen, C.-J. Chang, C.-H. Lee, A time-driven and event-driven approach for substation feeder incident analysis, Int. J. Electr. Power Energy Syst. 74 (2016) 9–15, doi: 10.1016/j.ijepes.2015.07.017 .
[18]H. Alshawish , A. de Meer , Risk mitigation in electric power systems: where to start? Energy Inform. 2 (1) (2019) 34 .
[19]G.D. Gonzalez Granadillo , J. Garcia-Alfaro , E.Y. Alvarez Lopez , M. El Barbori , H. Debar , Selecting optimal countermeasures for attacks against critical sys- tems using the attack volume model and the rori index, Comput. Electr. Eng. 47 (2015) 13–34 .
[20]G. Yadav, P. K., Patchrank: ordering updates for SCADA systems, in: 2019 24th IEEE ETFA, 2019, pp. 110–117, doi: 10.1109/ETFA.2019.8869110 .
[21]G. Yadav, P. Gauravaram, A.K. Jindal, Smartpatch: a patch prioritization frame- work for SCADA chain in smart grid, MobiCom '20, Association for Computing Machinery, New York, NY, USA, 2020, doi: 10.1145/3372224.3418162 .
[22]A .A . Cárdenas , S. Amin , S. Sastry , Research challenges for the security of con- trol systems, in: Proceedings of the 3rd Conference on Hot Topics in Security, in: HOTSEC'08, USENIX Association, Berkeley, CA, USA, 2008, pp. 6:1–6:6 .
[23]C. Neuman , Challenges in security for cyber-physical systems, in: Workshop on Future Directions in Cyber-physical Systems Security, 2009, pp. 1–4 .
[24]A.C.F. Chan, J. Zhou, On smart grid cybersecurity standardization: issues of designing with nistir 7628, IEEE Commun. Mag. 51 (1) (2013) 58–65, doi: 10. 1109/MCOM.2013.6400439 .
[25]C. Schuett, J. Butts, S. Dunlap, An evaluation of modification attacks on pro- grammable logic controllers, Int. J. Crit. Infrastruct.Prot. 7 (1) (2014) 61–68, doi: 10.1016/j.ijcip.2014.01.004 .
[26]Z. Basnight, J. Butts, J. Lopez, T. Dube, Firmware modification attacks on pro- grammable logic controllers, Int. J. Crit. Infrastruct.Prot. 6 (2) (2013) 76–84, doi: 10.1016/j.ijcip.2013.04.004 .
[27]R. Zhu, B. Zhang, J. Mao, Q. Zhang, Y. an Tan, A methodology for determining the image base of arm-based industrial control system firmware, Int. J. Crit. Infrastruct.Prot. 16 (2017) 26–35, doi: 10.1016/j.ijcip.2016.12.002
[28]NIST, National institute of standards and technology, 2017, https://www.nist. gov/ .
[29]I. Garitano , R. Uribeetxeberria , U. Zurutuza , A review of SCADA anomaly de- tection systems, in: E. Corchado, V. Snášel, J. Sedano, A.E. Hassanien, J.L. Calvo, D. S´ l e ¸ za k (Eds.), Soft Computing Models in Industrial and Environmental Ap- plications, 6th International Conference SOCO 2011, Springer Berlin Heidel- berg, Berlin, Heidelberg, 2011, pp. 357–366 .