Cyber Situation Awareness Perception Model for Computer Network

Olofintuyi Sunday Samuel
Mathematical Science Department
Achievers University, Owo, Nigeria

***Abstract*—With the increase in cyber threats, computer network security has raised a lot of issues among various companies. In order to guide against all these threats, a formidable Intrusion Detection System (IDS) is needed. Various Machine Learning (ML) algorithms such as Artificial Neural Network (ANN), Decision Tree (DT), Support Vector Machine (SVM), Naïve Bayes, etc. has been used for threat detection. In light of the novel threats, there is a need to use a combination of tools to accurately enhance intrusion detection in computer networks, this is because intruders are gaining ground in the cyber world and the side effects on organizations cannot be quantified. The aim of this work is to provide an enhanced model for the detection of threats on the computer network. The combination of DT and ANN is proposed to accurately predict threats. With this model, a network administrator will be rest assured to some extent based on the prediction of the model. Two different supervised machine algorithms were hybridized in this research. NSL-KDD dataset was deployed for the simulation process in WEKA environment. The proposed model gave 0.984 precision, 0.982 sensitivity and 0.987 accuracy.**

Keywords—Situation awareness; intrusion detection system; artificial neural network based decision tree; decision tree; classification

I. Introduction

There has been an increase in cyber threats which has caused damages all over the globe, this is not far-fetched from the fact that there is increase in the usage of computer networks and the tremendous applications usage especially after the advent of the internet of things [1]. Data encryption, user's authentication, hardware and software firewalls are some of the approaches used so far for threats detection. Regrettably, these methods may not be able to fully protect cyber threats from computer networks [2]. For example, firewalls can only monitor exchange of data between networks and no signal is given for internal attack. An accurate machine learning algorithm is obviously needed to build a formidable security system. Intrusion Detection System (IDS) helps to identify any form of abnormalities in behavior on a computer network [3]. Based on the user's perspective, IDS are of different types which are Network Based Intrusion Detection System (NIDS) and Host Based Intrusion Detection System (HIDS) [1]. In NIDS, threats are scanned across the networks while in HIDS abnormalities and inconsistency are only checked for in the operating system. Basically, there are two approaches for threat detection which are signature-based IDS and anomaly-based IDS. Signature based IDS helps to detect known threat because of the previous record in the database while anomaly-based IDS can detect some new set of threats [4]. Recently, there is high demand for protection against the various types of cyber-attacks. This is because the number of users of computer networks are increasing every day. With this, cyber-threats such as Denial Of Service (DOS), probe, User to Root (U2R) and Root to Local (R2L) are also gaining ground in cybersecurity [5]. According to [6], the activity of intruders cost an organization to loss about 8 billion dollars. The security community were able to detect about 50 million malwares in 2010, 100 million unique malwares were also detected in 2010. The number of executable malwares detected in 2019 skyrocketed to about 900 million unique malware and the number keeps increasing everyday, because of this fact, there is a need to build a formidable cybersecurity that can be adopted by organization in order to reduce their loss [7]. The author in [8] in his first work on Situation Awareness (SA), classified SA into three sections which include the following; perception, comprehension and projection. The "perception" which is always the first phase of SA model deals with detecting events (malicious and threat free) in the environment. The second phase of the model is the comprehension phase, this phase is connected to a trained database of various events, after which the perception relays it information to the comprehension, the comprehension phase then judges the event whether it is a malicious event or not. After that, the comprehension phase then relay the information to the projection phase which is finally feedback to the network administrator. This work is set to modify the perception phase by introducing two supervised machines into the perception phase in order to enhance security, this will help the network administrator to forestall attacks. Recently, there has been inspiring attempt at forestalling network attacks using one, two or more hybridized machine learning algorithms. The author in [9] uses ANN for threat detection on a computer network and compares the result with SVM. The author in [10] uses DT for threats classification on the computer networks, [11] adopts SVM and [12] introduces clustering algorithms. The results of these studies have been promising. However, there is still need for improvement. In this work, two supervised machine learning algorithms were used in the perception phase of the SA. The algorithms used are Decision Tree (DT) and Artificial Neural Network (ANN). DT is a classifier that has tree like structure and use historical dataset for its prediction. Artificial neural network is also a classifier which has three layers; input layers, hidden layers and the output layer. The data to be classify are allocated to each neuron in the input layer which feeds the hidden layers and then finally passed to the output layer. The algorithms were simulated in WEKA environment. WEKA is a simulating tool which contains various machine learning algorithms which are used for classification, data pre-processing, clustering, regression and so on [13]. The remaining section of this paper is as follows: Section II discusses the background to the study, section III discusses the methodology applied with the dataset used. The next section immediately that discusses the simulation results derived from the experiment. The final section talks about the conclusion and the possible future work.

II. Literature Review

In the field of data science and machine learning, one of the best techniques for building prediction models is classification [14]. Various classification algorithms have been proposed by researchers which are used in building a predictive model. Some of the works done so far in machine learning for prediction include the followings: The author in [15] proposed Deep Neural Network (DNN) for the detection of cyber security threats in Internet of Things (IoT) network. The dataset used was obtained from google code jam. After the experimental results, the proposed algorithm showed a better classification performance. The author in [16] deployed a bi-directional recurrent neural network for prediction of cyber-attack. Real world cyber-attack dataset was used to validate the proposed model. The proposed model gave a better prediction accuracy than statistical prediction model. The study by [17] proposed an intrusion detection system based on back propagation neural network. The authors developed an algorithm to classify four types of attacks namely; DOS, Probe, U2R and R2L. The algorithm was evaluated using the KDD99 dataset. The study obtained a detection rate of 0.99 and false alarm rate of 0.03 with a data size of 500. With two hundred increase in data size, the results obtained were still in the range above for detection rate and false alarm rate. The author in [18] proposed C4.5 Decision tree algorithm for the prediction of credit card risk. After data preprocessing and evaluation, the proposed algorithm gave 73.1% accuracy prediction. The accuracy of the C4.5 decision tree was improved to 75.1% by applying bagging ensemble algorithm. The author in [19] proposed a new technique based on soft computing in which a classifier called neuro-fuzzy was used for droppage of packet in MANETS. An Intrusion Detection System (IDS) was used for classification of threats for mobile ad-hoc network, this is because encryption and authentication are not considered as a very good solution to combat with the threat. The proposed IDS use neuro-fuzzy classifier. Matlab toolbox with Qualnet was used as a simulating tool. The result of the simulation showed that the proposed system efficiently detected dropping of packet while attack has low false positive rate and high true positive rate. The author in [20] presented a study on fuzzy rules for intrusion detection system. The work makes use of fuzzy set theory and the analysis of the function of genetic algorithm with fuzzy rule were done for intrusion detection system. In their result, they were able to gain maximum detection of DOS. This approach will be very useful, if the rules can be updated from time to time in order to meet up with the new attacks. The author in [21] presented real time intrusion detection with fuzzy, genetic and Apriori algorithm. The combination of these approaches was necessitated because several machines learning algorithms have been used for intrusion detection but a satisfactory result was not achieved. KDD 99 dataset was used in implementing the proposed model and their evaluation rate were detection speed, false alarm rate and attack types. After their experiment setup, it was observed that their proposed model gives a better outcome. The author in [22] proposed a work on a system for intrusion detection in which two machines were used which are support vector and genetic network programming. The database used in this work was classified into two namely positive kernel and negative kernel. Where the positive kernel was used in creating the rules. The experimental result shows that the combination of support vector machine and genetic network programming has increase the performance of the detection rate of intrusion detection system and also reduces the false positive rate. In the research conducted by [23], an artificial neural network based sequential classifier was used for the detection of false positive, false negative, true positive and true negative respectively. From the result obtained after the experiment, the introduction of sub classifier increases the accuracy of the proposed algorithm compared to individual model. The author in [24] uses five different classifiers in the proposed model to reduce the incidence of false negative rate. The five classifiers that were used include Naïve baye, multilayer perception, decision tree, random forest and K-nearest neighbor. 90.31% accuracy were achieved after the experimental result.

III. Methodology

Generally, Situation Awareness model has three phases which this model agrees with, but due to the inaccuracies in detection rate of threats, two algorithms were hybridized in the perception phase. Fig. 1 depicts the proposed model.

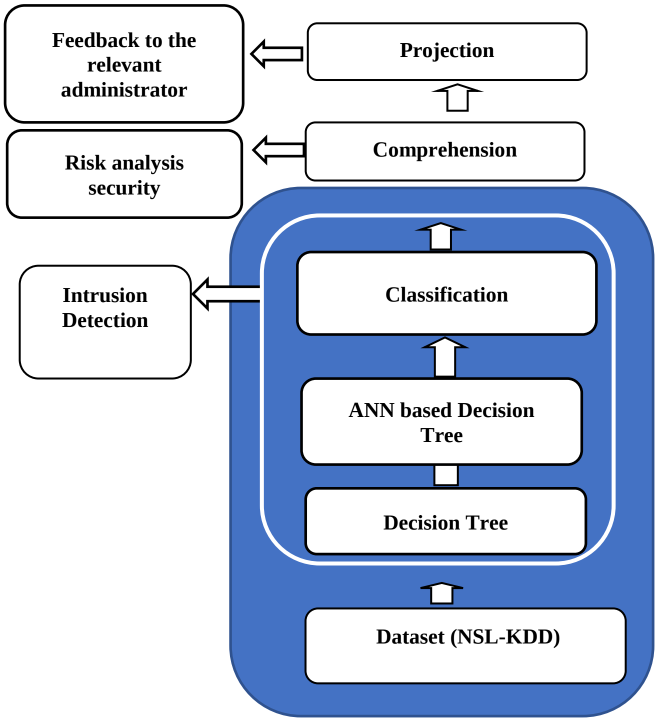

Fig. 1. Proposed Model.

The proposed model has three sections which are the perception phase, comprehension phase and projection phase. The perception phase which deals with detection of events in the environment is made up of two machine algorithms which are Decision Tree and Artificial Neural Network.

A. *Dataset*

NSL-KDD dataset was used in the simulation of the two algorithms used. The dataset is an online dataset which was derived from the Defense Advance Research Project Agency (DARPA). NSL- KDD dataset has forty-one features as gotten online. The dataset is in numeric form ranging from different numbers. But the dataset was all converted to 0's and 1's. NSL-KDD dataset consists of various categories which are:

1) Denial of Service (DOS): DOS is a group of attack, in which they keep the computing memory busy because of this, the memory no longer has time to attend to legitimate request. Example includes: Apache2, Mail bomb, Process table, Smurf, Udpstorm. Back, Land, Teardrop, Ping of death and SYN Flood.

2) User to root: These are group of attacks in which they approach a system as a normal or legitimate user of the system, meanwhile they are intruder. Once they get access to the system, they then explore the system vulnerabilities. Examples are Xterm, perl, loadmodule and fdformat.

3) Root to local: These are group of attacks in which they send packet of data to the network which they do not have access to. With this, they tend to gain access and explore the system vulnerabilities. Examples are FTP write, Imap, Xlock, Dictionary, Phf and Guest.

4) Probing: are also one of the categories of attack where by an attacker approach a system and then gain access to the system which later explore the vulnerabilities of the system. Examples are Saint, satan, Mscan, Ipsweep and Nmap. Table I below shows the 41 attributes of the dataset.

B. *Decision Tree*

Decision tree uses historical dataset for it prediction. It is structured as a tree where the node servers as feature and the edges are features value. The dataset used servers as input into the algorithm, it also considers the attribute of the dataset which helps in predicting the classes of threat. As used in the proposed model, decision tree was used to separate the dataset which has 41 attributes into the "attack group" and the normal group". The main reason why decision tree was firstly used was to classify each of the class of event. All threats are classified as 0 while the normal class was classified as +1. This was done in the excel environment and was then reopened in the notepad which was finally saved in arff format. The dataset must be in arff format so that the simulation tool (WEKA) will accept the dataset for simulation.

C. *ANN based Decision Tree Model (Proposed Model)*

In the second sub-phase of the model, ANN based decision tree is an excellent approach to resolve the problem of multiclass. Hybridizing different models enhances the performance compared to individual models, this is because hybridization reduces the weakness of the individual models. ANN based decision tree was used to classify the threat group into the various categories. ANN is a neural network of neurons which are inter-connected. Basically, ANN has three sections which are the input layer, hidden layer and output layer. The input layer takes the output features from decision tree which are then sent to the hidden layer and finally gets to the output layer. Since the classes of threats are classified into four categories from the dataset gotten, once the output gives 0001 it is classified as "user to root" if the output reads 0010 then it is probe. Furthermore, if the output displays 0100 it is classified as "Remote to local" and if it reads 1000 it then means it is "Denial of service". In this study, the input neurons are represented by each threat feature variables determined by Xi = {X_1, X_2, X_3 …X_n} where i is the number of variables (input neurons). The effect of the synaptic weights, W_i on each input neuron at layer j is represented by the expression:

$$Z_j = W_{1j}X_1 + W_{2j}X_2 + \ldots W_{3j}X_3 + b \quad (1)$$

TABLE I. LIST OF 41 FEATURES OF THE DATASET

Feature Index	Feature name	types
1	Duration	continuous
2	Protocol type	Symbolic
3	service	Symbolic
4	Flag	Symbolic
5	Scr_bytes	continuous
6	Dst_bytes	Continuous
7	Land	Symbolic
8	Wrong fragment	Continuous
9	Urgent	Continuous
10	Hot	Continuous
11	Num_failed login	continuous
12	Logged_in	Symbolic
13	Num_compropmised	Continuous
14	Root_shell	Continuous
15	Su_attempted	Continuous
16	Num_root	Continuous
17	Num_file creation	Continuous
18	Num_shell	Continuous
19	Num_access file	Continuous
20	Num_outbound_cmds	Continuous
21	Is_host_login	symbolic
22	Is_guest_login	symbolic
23	count	Continuous
24	Srv_count	Continuous
25	Serror_rate	Continuous
26	Srv_serror_rate	Continuous
27	Rerror_rate	Continuous
28	Srv_rerror_rate	Continuous
29	Same_srv_rate	Continuous
30	Diff_srv_rate	Continuous
31	Srv_diff_host_rate	Continuous
32	Dst_host_count	Continuous
33	Dst_host_srv_count	Continuous
34	Dst_host_same_srv_rate	Continuous
35	Dst_host_diff_srv_rate	Continuous

36	Dst_host_same_src_port_rate	Continuous
37	Dst_host_srv_diff_host_rate	Continuous
38	Dst_host_serror_rate	Continuous
39	Dst_host_srv_rate	Continuous
40	Dst_host_srv_serror_rate	symbolic
41	Dst_host_serror_rate	symbolic

Equation (1) is sent to the activation function (sigmoid/logistic function) and applied in order to limit the output to a threshold [0, +1]. The difference between the expected output (p) and the actual output (y) is derived using the squared error measure (E):

$$E = (p – y)^2 \tag{2}$$

The output (p) of a neuron depends on the weighted sum of all its inputs as indicated in (1). In this research work, the gradient descent algorithm is applied in order to minimize the error and hence find the optimal weights that satisfy the problem. Derivative of the square error function needs to be calculated with respect to the network's weight. In order to cancel the exponential of 2 when differentiating, ½ is required which is used to redefine the square error function.

$$E = 1/2\ (p – y)^2 \tag{3}$$

Every neuron j, is defined by the output O_j

$$O_j = Q\left(net_j\right) = Q\sum_{k=1}^{n} W_{ij}\, X_i \tag{4}$$

The input net_j to a neuron is the weighted sum of outputs O_j of the previous neurons. The number of input neurons is n and the variable W_{ij} denotes the weight between neurons i and j. Table II: The activation function φ is in general non-linear and differentiable, thus, the derivative of the (1) is:

$$\frac{\partial \varphi}{\partial z} = \varphi(1-\varphi) \tag{5}$$

TABLE II. PATTERN OF THREATS DETECTION

S/N	Attack Group	Different Attacks	Output
1	Denial of service attack	Black, Land, neptune, smurf, teardrop	1000
2	Remote to local attack	ftp write, guess password, imap, multihop	0100
3	user to root attack	buffer overflow, loadmodule, perl, rootkit	0010
4	Probes	satan, ipsweep, nmap, portsweep	0001

The partial derivative of the error (E) with respect to a weight W_{ij} is done using the chain rule twice as follows:

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial O_j}\frac{\partial O_j}{\partial net_j}\frac{\partial net_j}{\partial W_{ij}} \tag{6}$$

The last term on the left hand side can be calculated from (4), thus:

$$\frac{\partial net_j}{\partial W_{ij}} = \frac{\partial}{\partial W_{ij}}\left(\sum_{k=1}^{n} W_{ij}\, X_i\right) = X_i \tag{7}$$

The derivative of the output of neuron j with respect to its input is the partial derivative of the activation function (logistic function) shown in (5).

$$\frac{\partial O_j}{\partial net_j} = \frac{\partial}{\partial net_j}\varphi\left(net_j\right) = \varphi\left(net_j\right)\left(1-\varphi\left(net_j\right)\right) \tag{8}$$

The first term is evaluated by differentiating the error function in (3) with respect to y, so if y is in the outer layer such that y = O_j, then:

$$\frac{\partial E}{\partial O_j} = \frac{\partial E}{\partial y} = \frac{\partial}{\partial y}\frac{1}{2}(p-y)^2 = y - p \tag{9}$$

Considering E as a function of the inputs of all neurons, receiving input from neuron j and taking the total derivative with respect to a recursive expression for the derivative is obtained:

$$\frac{\partial E}{\partial O_j} = \sum_{i\in L}\left(\frac{\partial E}{\partial net_i}\frac{\partial net_i}{\partial O_j}\right) = \sum_{i\in L}\left(\frac{\partial net_i}{\partial O_J}\frac{\partial O_i}{\partial net_i}W_{ij}\right) \tag{10}$$

Thus, the derivative with respect to O_j can be calculated if all the derivatives with respect to the outputs O_j of the next layer – the one closer to the output neuron – are known. Putting them all together:

$$\frac{\partial E}{\partial W_{ij}} = \delta_j X_i \tag{11}$$

With:

$$\delta_j = \frac{\partial E}{\partial 0_j}\frac{\partial 0_j}{\partial net_j} = \{\sum\nolimits_{i\varepsilon L}(0_j - P_j\}\varphi(net_j)(1 - \varphi(net_i) if\ j\ is\ an\ output\ neuron$$

$$\delta_j w_{ji)}\,(net_j)\ (1-(net_i))\ if\ j\ is\ an\ inner\ neuron$$

Therefore, in order to update the weight W_{ij} using gradient descent, one must choose a learning rate ∝. The product of the learning rate and the gradient is equal to change in weight added to the old weight.

$$\Delta W_{ij} = \propto \frac{\partial E}{\partial W_{ij}} \tag{12}$$

Equation (12) is used by the back-propagation algorithm to adjust the value of the synaptic weights attached to the inputs at each neuron in (1) with respect to the inner layer of the multi-layer perceptron. Table II depicts the pattern of threat detection.

D. Event Handler Flowchart

The event handler flowchart in Fig. 2 shows how events are being treated on a computer network with respect to the proposed model and the data used for this research. The output of the data used for this research work are grouped into five categories. The categories include Probe, Denial of Service (DOS), Root to Local (R2L), User to Root (U2R) and Normal. At the third stage of the event handler, the event is being cross-checked whether it belongs to the classes of Root to Local, if it does, it is then disseminated to the network administrator but if the event does fall under root to local, it then moves to the fourth stage. At this stage, it is also being ascertained whether the event belongs to User to Root, if it does, the information is passed to the network administrator and if it is not under User to Root, then it is a normal event.

E. Performance Evaluation

The proposed model was evaluated using the following metrics:

1) Sensitivity: Talks about event that are truly positive and are been announced as positive.

$$Sensitivity = \frac{TP}{TP + FN} \tag{13}$$

2) Precision: Describe events that are negative and are been announced as negative event.

$$Precision = \frac{TP}{TP+FP} \tag{14}$$

3) Accuracy: describes the general effectiveness of the proposed model.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{15}$$

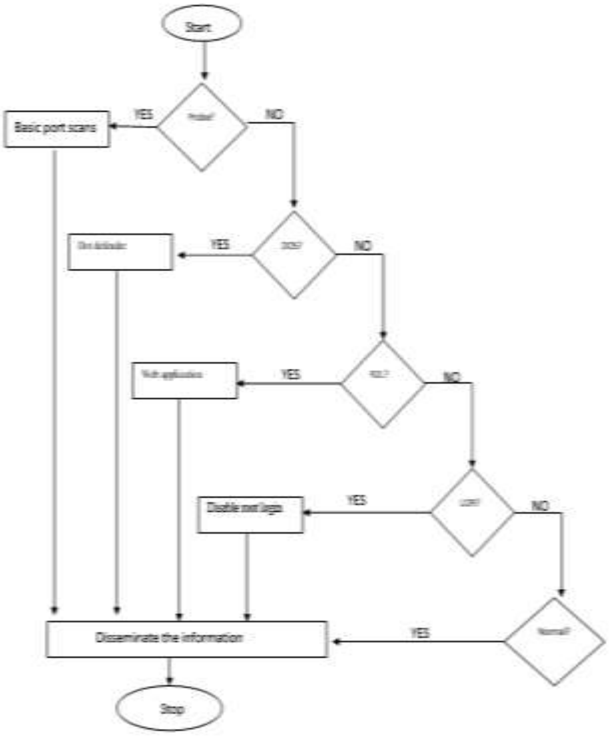

Fig. 2. Event Handler Flowchart.

F. Experimental Setup

In this research work, WEKA simulating tool was used in simulating the proposed model. Two supervised machines were selected and hybridized from the environment. The dataset used was also preprocessed in the excel environment and was then saved in CSV format. For our simulating tool to recognize the dataset, it must be saved in arff format which was done so that the dataset can be recognized. During the experiment process, the dataset was trained and tested. 10-fold cross validation was selected during the process. With the 10-fold cross validation, the dataset was divided into ten different samples where nine out the partitioned dataset was used for training and the testing was done with the remaining one. In a nutshell, the forty-one features given in the NSL KDD dataset was used during the model training.

IV. Result and Discussion

After the simulation of the models, the two machine algorithms used gave different results for different detection rates. The results show the effectiveness of an ensemble classifier over a single classifier. Decision tree classifier gave 58,119 correctly classified instances where the True Positive (TP) is 32052 and the True Negative (TN) is 26067. The incorrectly classified instances are 1,158 instances where the False Positive (FP) derived is 579 instances and the False Negative (FN) is 579 instances. The artificial neural network-based decision tree gave 58,505 correctly classified instances where the TP is 32052 instances, TN is 26453 instances and 772 were incorrectly classified where the FP is 193 instances and FN 579 instances. Fig. 3 depicts the graph that compares the performance of the two algorithms. Since the strength of both machines are different, and both can detect threats of different kinds, (that is) the group of threats both algorithms can detect varies. Table III depicts the experimental results. It was of great advantage combining the two machines for detection.

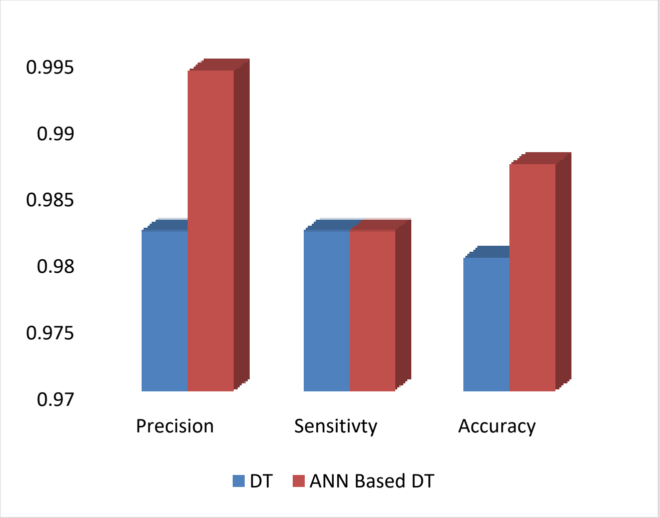

Fig. 3. Performance Compared of the Two Algorithms.

TABLE III. Experimental Results after Simulation

Classifiers	No of instances	TP	FP	TN	FN	Precision	Sensitivity	Accuracy
Decision Tree (DT)	59,277	32052	579	26067	579	0.982	0.982	0.980
ANN based DT	59,277	32052	193	26453	579	0.994	0.982	0.987

V. Conclusion

The effectiveness and efficiency of a machine learning based intrusion detection system for threats detection to companies and other networks users is of great value. Cyber threats range from one category to another categories, because of this fact, some machine learning may not be able to detect some of these threats on a computer network. The study proposed a robust cyber situation awareness model which combines the artificial neural network and decision tree as a detector which provided an improved prediction of intrusion on a computer network and enhances cyber security. After the design and simulation of a prediction model for threats detection, the hybridization of the two algorithms in the perception phase suggests a more secured system than a single classifier. Overall, the two classifiers used which are IDS based are data oriented which helps to detect the various patterns in the NSL-datasets. The ANN base decision tree gave a better accuracy of 98.7% than the decision tree classifier. The difference between the results of the proposed algorithm and decision tree may look insignificant but cannot be overlooked. The network administrator will definitely find this system useful for adequate awareness on a computer network. Future work will focus on how to still improve the accuracy of IDS by introducing more than two ensemble algorithms in the perception phase.

References

[1] H.I. Sarker, A.S.M. Kayes, S. Badsha, H. Alqahtani, P. Waters, and N. Alex "Cybersecurity data science: an overview from machine learning perspective" Journal of Big data, (2020).

[2] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," Journal of Information Security Application, vol. 44, pp. 80–88 (2019).

[3] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. HOU and C. Wang "Machine learning and deep learning methods for cybersecurity," IEEE Access 6, 35365–35381 (2018).

[4] I.H. Sarker, Y.B. Abushark, F. Alsolami, I. Khan "Intrudtree: a machine learning-based cyber security intrusion detection model," Symmetry 12, 754 (2020).

[5] B. Gupta, A. Tewari, A. Jain, and D. Agrawal, "Fighting against phishing attacks: state of the art and future challenges," Neural Computation Application. 28(12):3629–54, 2017.

[6] X. Qu, L. Yang, K. Guo, L. Ma, M. Sun, M. Ke and M. Li, "A Survey on the Development of Self-Organizing Maps for Unsupervised Intrusion Detection" Mobile Network and Application. 10.1007/s11036-019-01353-0, 2019.

[7] Z. Hao, Y. Feng, H. Koide, K. Sakurai " A sequential detection method for intrusion detection system based on artificial neural networks" International Journal of Networking and Computing, Vol. 10, Number 2, pp. 213–226, July 2020.

[8] M. Endsley, "Toward a theory of situation awareness in dynamic systems" In Human Factors Journal, volume 37(1), pages 32–64,1995.

[9] S. S. Olofintuyi, T. O. Omotehinwa, O. H. Odukoya and E. A. Olajubu "Performance comparison of threat classification models for cyber-situation awareness," *Proceedings of the OAU Faculty of Technology Conference,* 2019.

[10] G.Stein, B. Chen, A. Wu and K. Hau "Decision tree classifier for network intrusion detection with ga-based feature selection," *Proceedings of the 43rd annual Southeast regional conference*. *ACM*, 2, 2005.

[11] E.A. Shams and A.A Rizaner "Novel support vector machine based intrusion detection system for mobile ad hoc networks," Wireless Networks, 2018.

[12] W.C. Lin and S.W. Ke "An intrusion detection system based on combining cluster centers and nearest neighbors," Knowledge-based systems, 2015.

[13] S. Hettich, and S.D. Bay, (1999). The UCI KDD Archive. Irvine, CA: University of California, Department of Information and Computer Science.

[14] J. Han, J. Pei and M. Kamber, "Data mining: concepts and techniques," Elsevier, Amsterdam, 2011.

[15] F. Ullah, H. Naeem, S. Jabbar, S. Khalid, M. Latif, F. AL-Turjman, and L. Mostarda, "Cyber security threats detection in internet of things using deep learning approach," IEEE Access. PP.1-10, 2019.

[16] X. Fang, M. Xu, S. Xu and Z. peng, "A deep learning framework for predicting cyber-attacks rates," EURASIP Journal on Information security 2019.

[17] M. S. Mehibs, and H. S. Hashim, "Proposed Network Intrusion Detection System In Cloud Environment Based on Back Propagation Neural Network," Journal of Babylon University, Pure and Applied Sciences, 26(1), 29-40, 2018.

[18] M. A. Muslim, A. Nurzahputra, and B. Prasetiyo, "Improving accuracy of C4.5 algorithm using split feature reduction model and bagging ensemble for credit card risk prediction," in IEEE International Conference on ICT (ICOIACT), 2018, pp. 141–145, 2018.

[19] A. Chaudhary, V. N. Tiwari and A. Kumar, "Design an anomaly based fuzzy intrusion detection system for packet dropping attack in mobile ad hoc networks," IEEE International Conference on Advance Computing (IACC), pp. 256-261, 2016.

[20] P. Sarathi "A Study on Fuzzy Rules for Intrusion Detection" International Journal of Research Studies in Computer Science and Engineering (IJRSCSE) Vol 2, Issue 8, pp 1-9, August 2015.

[21] Rupesh and Balasaheb, "Real Time Intrusion Detection With Fuzzy, Genetic and Apriori Algorithm," International Journal of Advance Foundation and Research in Computer (IJAFRC) Vol 1, Issue 11, ISSN 2348 – 4853, 2014.

[22] P. Kola, C. Suba and A. Kanna, "Network intrusion detection system using genetic network programming with support vector machine," in *Proceeding of the International Conference on Advances in Computing, Communication and Informatics*. Pp. 645-649. Chennai, India, August 3, 2012.

[23] Z. Hao, Y.Feng, H.Koide and K. Sakurai, "A sequential detection method for intrusion detection system based on artificial neural networks. International Journal of Networking and Computing, ISSN 2185-2847 Vol 10, Number 2, pp 213–226, July 2020.

[24] Wu C Phetlasy S, Ohzahata S et al. A sequential classifiers combination method to reduce false negative for intrusion detection system. IEICE Transactions on Information and Systems, 102(5):888–897, 2019.