A Three-Tier Model for Intrusions Classification on a Computer Network

Sunday Samuel Olofintuyi

Department of Computer Science, College of Natural and Applied Science, Achievers University, Owo, Ondo State, Nigeria.

email: Olofintuyi.sundaysamuel@gmail.com

Received: 24 December 2022 | Accepted: 22 May 2023 | Early access: 22 Jun 2023

Abstract - *Activities of cyber attackers are on the rampage; this is because there is an increase in the usage of computer related applications. Attackers have caused reputational and economic damages to network administrators, companies and industries based on the information they have stolen. To curb all these activities, a formidable Intrusion Detection System (IDS) is needed to guide against all the numerous cyber-attacks. The research work solely aimed at reducing the accessibility of cyber threats by bringing its operations to as minimal as possible because of the adverse effects they have had in the past. This research proposed a three-tier IDS which classifies the various attacks into their various groups. The proposed model consists of Bayes Network (BN), Support Vector Machine (SVM), and Artificial Neural Network (ANN). NLS KDD 99 dataset was used for simulating the proposed three-tier IDS in the WEKA environment. The effectiveness and efficiency of the proposed model was based on recall, precision, and accuracy. The proposed three-tier model gave the following results: recall: 0.993; precision: 0.979; accuracy: 0.986.*

Keywords: Classification, Intrusion Detection System, Cyber-attacks, Machine Learning, Cyber Security.

1 Introduction

The advent of Internet of Things (IoT) has generated more activities of cyber-attacks on the network and the tremendous usage of computer applications by user has also increase cyber threats on a computer network. Companies, industries, and various sectors of the economy have suffered a serious setback because of the devastating effects of cyber-threats. Also, measures have been put in place by companies, industries, research institutes, government, and network administrators to curb all these cyber-attacks but all effort seem not enough (Olofintuyi, 2021). In 2010, there were about 50 million malwares, and in less than 3 years, the number of malwares has increased to about 100 million. Unexpectedly, by the year 2019, the number of malwares have skyrocketed to about 900 million cyber threats (Sarker et al., 2020). Morgan (2021) predicted in his work that by the year 2021, the global crime rate will cost about $6 trillion USD and $10.5 trillion by the year 2025. Recently, First America records that about 900 million records were compromised and the account of American Medical Collection Agency (AMCA) was hacked and the attackers were able to gain access to their account and record for almost a year (Hao et al., 2020). Hardware and software firewalls, user's authentication and data encryption are some of the mechanisms that have been adopted to curb the activities of intruders. Unfortunately, all these mechanisms seem not robust enough for effective and efficient guidance against these cyber-threats (Mohammadi et al., 2019). For instance, a firewall only gives signal when communication takes place between two or more networks, but it doesn't give any signal for any form of internal attacks. With this, a more secure, robust, and accurate Machine Learning (ML) based IDS is needed for the safety of the system. IDS is a system that detects inconsistencies, attacks, irregularities, infectious activities, and any form of abnormalities on a network such as Root to Local (R2L), Denial of Service (DoS), User to Root (U2R) and probe (Olofintuyi & Omotehinwa, 2021). IDS is also suitable for classifying the various threats into their respective classes using either Machine Learning or statistical methods. The following metrics: True Negative (TN), False Negative (FN), True Positive (TP) and False Positive (FP) were used to evaluate the aforementioned algorithms. Obtaining optimum results for all the metrics at the same time seem impossible because each metric is dependent on each other. The big challenge comes in when striking the balance between them (Hao et al., 2020). The various classes of threats can be identified and grouped into their various classes by a data driven IDS. This is possible when IDS analyzes the

patterns in the cyber threat and then categorizes them into various classes. ML algorithms are needed to build data driven IDS. However, different ML predicts based on their context hence, each algorithm classifies threats on the network to different groups based on their context (Alqahtani et al., 2020). Based on the pertinent reasons, a three layers model has been proposed for classifying the dataset into threats and benign to reduce the false negative and overall increase the accuracy. Efficiency and effectiveness of the proposed three-tier model is evaluated based on Recall, Precision, and Accuracy. The next section discusses the review of literature, methodology used, result obtained and conclusion.

2 Literature Review

Activities of cyber threats differ on the computer network because of this: an IDS system is needed to classify each threat to their respective classes (Stallings, 2003). Generally, IDS can be broadly classified into two types which are Anomaly Based Intrusion Detection System (AIDS) and Signature Intrusion Detection System (SIDS) (Lin et al., 2013). AIDS detects threats based on the new pattern established by the system. AIDS generates a new model with a new pattern which is capable of detecting any unknown threat (Buczak & Guven, 2016). Operation of SIDS is quite different from AIDS. SIDS classifies threats based on known patterns. SIDS cross checks the pattern of the threat on the network against the patterns of the event known and then classifies each activity to their respective groups. SIDS is effective and efficient when it comes to classifying known attacks but it is ineffective at classifying unknown attacks. A good example of SIDS is an expert system developed in mid-1960 (Liao, 2005). Machine learning and statistical methods have been the major approaches used for classifying threats under the anomaly-based intrusion detection system. Operation of statistical methods is based on assumption whether a particular situation is normal or abnormal. Also, there is inconsistency in the assumption made with the statistical method and because of this, the parameters are not easily determined (Zhao, 2020). Machine learning algorithms such as Support Vector Machine (SVM) (Shams & Rizaner, 2018), ANN (Olofintuyi et al., 2019) BN (Sarker et al., 2020) and clustering (Lin & Ke, 2015) have played a vital role in IDS but there are some loopholes in their operation and this is not far-fetched from the fact that each classifier predicts base on their context (Sarker, 2019; Olofintuyi & Olajubu, 2021).

Vapnik and Corinna (1995) were the first to propose SVM, and since then, many other researchers have used it for threat classification on the computer network. Aslahi- Shahri (2006) proposed a hybrid model of support vector machine and genetic algorithm for threat detection on the network. KDD99 dataset was used for model simulation and an accuracy of 97.3% was derived. Also, to solve the problem of low detection rate, Pozi et al., (2016) proposed a hybridized approach for classifying threats. The hybridized model consists of SVM and genetic programming and accuracy of 89.28% was achieved. Horng et al., (2011) presented a novel hybridized model which consisted of SVM and hierarchical clustering. KDD99 dataset was also used and an accuracy of 95.7% was achieved.

Another powerful machine learning algorithm used is Decision Tree (DT). DT uses a sequence of decisions to classify events into their respective classes. DT adopts a tree-like approach for classification. Rahman et al., (2010) uses DT for threats classification on the computer network; KDD99 dataset was also used for model simulation and an accuracy of 98% was derived. Sahn and Mehtre (2015) also used J48 for threats classification; Kyoto 2006+ dataset was used for model simulation. After the experiment, 97.2% accuracy was achieved.

Gang (2010) achieved 96.71% accuracy on NSL-KDD data using neural network and clustering algorithms. Also, Mansour et al., (2012) presented a recurrent neural network for intrusion detection; the performance evaluation was based on KDD dataset and an accuracy of 94.1% was obtained. Long Short Term Memory Recurrent Neural Network (LSTM-RNN) was proposed by Jihyun et al. (2016); LSTM-RNN was used to classify the event on the network and 93.93% accuracy was achieved using the KDD dataset.

3 Methodology

A sequential three-tier model is proposed because a single classifier is limited when it comes to detecting the entire negatives. The proposed workflow was adopted to reduce the False Negative (FN) and then improve the overall accuracy. Also, it was adopted to know the effectiveness of increasing classifiers on the network. NLS-KDD 99 dataset was used in model building and simulation in Waikato Environment for Knowledge and Analysis (WEKA). The three-tier model consists of Bayesian Network (BN), Support Vector Machine (SVM), and Artificial Neural Network (ANN). Firstly, NSL-KDD 99 is fed into the proposed model, BN classifies the event as either threats or benign. Benign from the first classifier are reclassified to detect the false positive and false negative and overall improve the accuracy. Once classified as threat, the administrator tagged such traffic as threat. But we are more concern about the benign because there are still some elements of threats in it. This is so because

of the weakness of the classifiers. This approach is adopted because of the context of each machine learning algorithm; bearing in mind that FN must be at the minimal level to protect the network administrator. The benign output is forwarded to SVM which also classifies the event into threat or benign. The output from SVM is fed into ANN. At this stage, ANN re-classifies the threats and benign to reduce the FN in the dataset. Figure 1 depicts the flowchart for the proposed model.

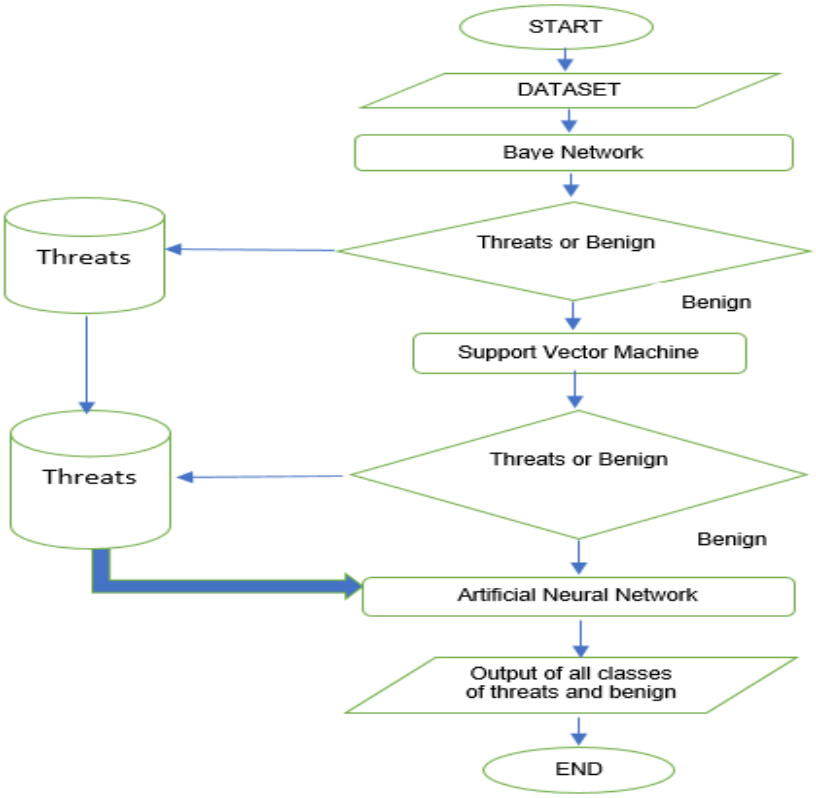

Figure 1: Flowchart of a Three-Tier Model

3.1 Dataset

NSL KDD 99 dataset was obtained online, the dataset is an extract from KDD99 dataset. NSL KDD99 dataset does not contain any redundant and irrelevant features. The dataset has forty-one (41) attributes and the dataset has five classes which are; DoS, Probe, U2R, R2L and benign. Table 1 depicts the forty-one features of the dataset used.

DOS: DoS is the first group of threats considered in this research work. The aim of this threat is to shut down the network so that intended users which are legitimate will not have access to it. Traffic is used to flood the target to get this task accomplished. Examples include SYN Flood, Ping of death, Back, Land, Process table, Mail tomb and Apache 2.

Probe: The first line of action of the probe is to obtain vital information from the network after which it launches its attacks. Examples include Mscan, Nmap, Satan, saint and Ipsweep.

Root to Local: The system becomes vulnerable when packets of data are sent by the attackers and the end user accepts it. Examples include Xlock, Dictionary, Imap, FTP Write and Guest.

User to Root: U2R gained access to the system in disguise to be legitimate users. These groups of threat explore the vulnerabilities of the system once they have gained access to the system. Examples include Perl, Xtem, Loadmodule and Fdformat.

Table 1: The forty-one features of the dataset

No	Feature name	Types	NO	Feature Name	Types	NO	Feature name	Types
1	Duration	Continuous	15	Su_attempted	Continuous	29	Same_srv_rate	Continuous
2	Protocol type	Symbolic	16	Num_root	Continuous	30	Diff_srv_rate	Continuous
3	service	Symbolic	17	Num_file creation	Continuous	31	Srv_diff_host_rate	Continuous
4	Flag	Symbolic	18	Num_shell	Continuous	32	Dst_host_count	Continuous
5	Scr_bytes	Continuous	19	Num_access file	Continuous	33	Dst_host_srv_count	Continuous
6	Dst_bytes	Continuous	20	Num_outbound_ cmds	Continuous	34	Dst_host_same_srv_rate	Continuous
7	Land	Symbolic	21	Is_host_login	Symbolic	35	Dst_host_diff_srv_rate	Continuous
8	Wrong fragment	Continuous	22	Is_guest_login	Symbolic	36	Dst_host_same_src_port_ra te	Continuous
9	Urgent	Continuous	23	count	Continuous	37	Dst_host_srv_diff_host_rat e	Continuous
10	Hot	Continuous	24	Srv_count	Continuous	38	Dst_host_serror_rate	Continuous
11	Num_failed login	Continuous	25	Serror_rate	Continuous	39	Dst_host_srv_rate	Continuous
12	Logged_in	Symbolic	26	Srv_serror_rate	Continuous	40	Dst_host_srv_serror_rate	Symbolic
13	Num_compro pmised	Continuous	27	Rerror_rate	Continuous	41	Dst_host_serror_rate	Symbolic
14	Root_shell	Continuous	28	Srv_rerror_rate	Continuous			

3.2 Bayes Network (BN)

BN is the first algorithm used on the three-tier model. NSL-KDD 99 dataset was fed into the algorithm, and the algorithm then classifies the dataset into two categories as either threat or benign. Threats are malicious activities that aim to intrude into the network and steal vital information while benign are activities that are not harmful to the computer network.

3.3 Support Vector Machine (SVM)

This algorithm classifies the output from the BN. Although BN has classified the output as benign, because of the different content of how each algorithm classifies, SVM is used to re-classify the output again as either threat or benign. SVM classifies each point in the space into various categories using a hyper-plane.

3.4 Artificial Neural Network (ANN)

ANN is the third algorithm used to classify the dataset into threat or benign. ANN has basically three components which are the input, hidden, and output layers. Output from SVM and database of threats serves as input into the input layer of ANN. The hidden layer performs its operation by using sigmoid activation function. The output layer classifies each group into their respective classes as depicted in Table 2.

Table 2: Threat/Benign classification based on their group

S/N	Attacks/Benign	Different attacks	Output
i	Denial of service attack	Mail bomb, Ping of death, Land, SYN, Process table Flood, Back and Land.	00001
ii	Root to local	Xlock, Guest, Dictionary, write, Imap and FTP	00010
iii	User to root	Xterm, Fdformat, Loadmodule and Perl.	00100
iv	Probes	Mscan, Saint, Ipsweep, Satan and Nmap	01000
v	Benign		10000

3.5 Performance Evaluation

The proposed three-tier model was validated after experimental simulation with the following metrics:
False Positive: This classifies events that are negative as positive wrongly.
False Negative: This metric misclassifies positive events as negative.
True Positive: Report events that are positive correctly
True Negative: Report events that are negative correctly
Recall: Completeness and quantity of the model are being measured by this parameter. Equation 1 depicts the formula for recall.

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

Precision: This described the exactness and quality of the proposed model. Equation 2 depicts the formula for precision.

$$Precision = \quad (2)$$

Accuracy: This described the effectiveness of the proposed model. Equation 3 depicts the formula for accuracy.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (3)$$

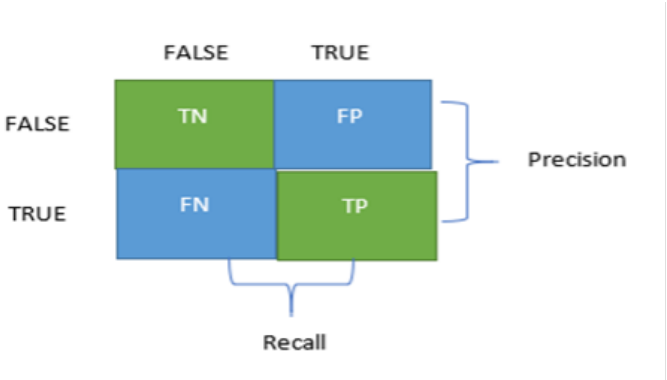

Figure 2: Confusion matrix for evaluation

3.6 Experimental Setup

The whole experiment was carried out in a WEKA environment. The dataset (NSL-KDD 99) used has no redundant and irrelevant features because it is an extract from KDD 99 dataset. The dataset was firstly saved in CSV format and later converted to arff format. This is done because that is the format WEKA recognizes in order to accept the dataset for simulation. 10-fold cross validation was applied during simulation, where the dataset was partitioned into ten samples. 9 of the samples were used for model training while the remaining one was used for testing. Finally, the performance of the three-tier model was based on how the model was able to correctly classify instances. In the experimental setup, three different classifiers were selected and combined so that we can produce low FN. In adding a new classifier to the first, we carefully select a model that produces low FP and FN so that we will be able to achieve a low overall FN for the three-tier model. By adding new classifiers, we expect that each added classifier should improve on the limitation of the first and overall improve the accuracy. The proposed model adopts three classification steps. Firstly, the model was created and classification was done by the models. In the training phase, 10-fold cross validation was used to avoid overfitting and the best model was selected based on the turning parameters. Immediately after the training phase, the classification begins. For each of the classifier, the testing data is fed into it. The intention of the researcher is to reduce FN by detecting the negatives from each benign classified. The positive output from BN is forwarded as input into SVM. Also, the positive output from SVM is fed into ANN which does the final classification.

4 Results and Discussion

The first algorithm used for classification in the first layer is BN. The algorithm correctly classified 50,588 instances and wrongly classified 8689 instances. The following results were obtained from the first layer after simulation: recall: 0.790; precision: 0.871; accuracy: 0.8534. For the second layer, SVM algorithm was used, and it classified 58119 instances correctly and 1158 incorrectly. The following results were obtained from the second layer: recall: 0.978; precision 0.978, accuracy: 0.9804. ANN was used for the third layer and 58505 instances were correctly classified while 772 were incorrectly classified. The following results were obtained for the third layer: recall: 0.993; precision: 0.979; accuracy: 0.9869.

From Table 3, the first layer (BN) produced an FN of 5599 instances and an accuracy of 85.34% which is not too satisfactory for an administrator. And because of this, another classifier (SVM) was introduced which reduced the FN drastically to 579 instances and produced an accuracy of 98.04 %. To further improve the FN and accuracy, another classifier (ANN) was used, and the final FN gave 193 instances while 98.69 % accuracy was achieved. This shows that using a combination of classifiers can drastically improve the FN and accuracy as compared to a single classifier. The three-tier model reduces the FN from 5599 instance for the first layer to 579 instances in the second layer and 193 instances in the third layer of the model. It is suspected that the difference in the FN between the first layer and the second layer is 5020 instances. This is suspected to be so because the first layer actually does the classification as threat or benign. It is benign that is passed to the second classifier to check and reclassify if there is other malicious traffic in the benign so that it can reclassify. Finally, ANN does the final classification and gave an improvement of 386 FN as compared to the second layer and 5406 FN as compared to the first layer. Table 3 depicts the results of the different three layers. From the table, layer 3 of the model gave a better accuracy compared to the respective two layers. Figure 3 depicts the bar chart of the three-tier model.

Table 3: Evaluation Table for the three-tier model

Layers	Instances	TN	TP	FN	FP	Recall	Precision	Accuracy
Layer 1	59,277	29543	21045	5599	3090	0.7902	0.8712	0.8534
Layer 2	59,277	32054	26065	579	579	0.9781	0.9781	0.9804
Layer 3	59,277	32054	26451	193	579	0.9933	0.9792	0.9869

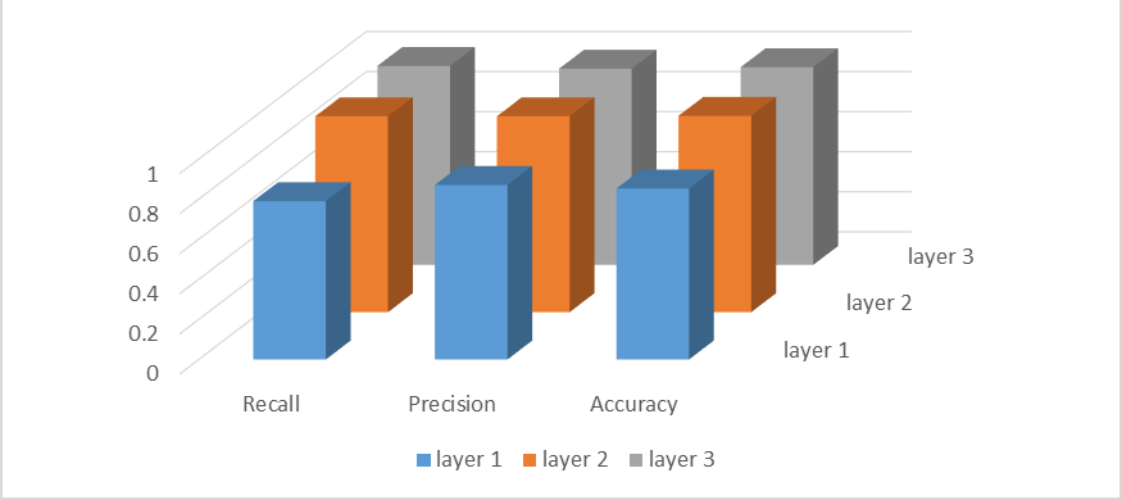

Figure 3: Chart to depict the evaluation results for the three-tier model

5 Conclusions

Detection rate of each algorithm differs and this is based on the context of each of the algorithms. Also, there are various categories of cyber-attacks on the network which some of them have outplayed some of the mechanisms put in place to curb them. It is with this reason, that the study proposed a three-tier model. The proposed three-tier model will reduce activities of threats in companies, industries, IT offices and government parastatal if fully deployed. Finally, the proposed model gave 97.92% precision, 99.33% recall. And 98.69% accuracy.

References

Alqahtani, H., Sarker, I. H., Asra K., Syed Md. Minhaz, H., Sheikh I., & Sohrab H. (2020). Cyber Intrusion Detection Using Machine Learning Classification Techniques. *Springer Nature Singapore*, CCIS 1235, pp. 121–131.

Aslahi-Shahri, M. (2016). A hybrid method consisting of genetic algorithm and support vector machine for intrusion detection system. *Neural computing and applications*, 27(6):1669-1676.

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE* Communications Surveys and Tutorials.

Chen, Y. H., Horng, S. J., & Su, M., Y. (2011). A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert systems with Applications*, 38(1):306-313

Gang, M. (2010). A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert systems with applications*, 37:9.

Harbi, N., Rahman, C. M. & Farid, D. M (2010). Attacks classification in adaptive intrusion detection using decision tree. *World academy of science, engineering and technology*, 39:86-90.

Hao, Z., Feng, Y., Koide, H. & Sakurai, K. (2020). A sequential detection method for intrusion detection system based on artificial neural networks. *International Journal of Networking and Computing*, 10:213-226

Liao, S H. (2005). Expert system methodologies and applications|. A decade review from 1995 to 2004. *Expert systems with applications*, 28(1):93-103.

Lin, C. H., Liao, H. J., & Lin, Y. C. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16-24.

Lin, W. C. & Ke, S. W. (2015). An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge-based system.

Mohammadi, S., Mirvaziri H., Ghazizadeh-Ahsaee, M. & Karimipour, H. (2019). Cyber intrusion detection by combined feature selection algorithm. *Journal of Information Security and Application*, 44:80-88

Morgan, S (2021). Cyberwarfare in the suite. Cyber security magazine. Publish by cybersecurity ventures.

Mustapha, N., Pozi, M., & Sulaiman, M. (2016). Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming. *Neural Processing Letters*.

Olofintuyi, S.S. (2021). Cyber Situation Awareness Perception Model for Computer Network. *International journal of advanced computer science and application*. 12(1):392-397.

Olofintuyi, S.S. & Olajubu, E.A (2021). Supervised Machine Learning Algorithms for Cyber-Threats Detection in the Perception Phase of a Situation Awareness Model. *International Journal of Information Processing and Communication*, 11(2): 61-74.

Olofintuyi, S.S. & Omotehinwa, T.O. (2021). Performance Evaluation of Supervised Ensemble Cyber Situation Perception Models for Computer Network. *Computing, Information Systems, Development Informatics and Allied Research Journal*. 11(2):1-14.

Olofintuyi, S.S., Omotehinwa, T. O., Odukoya, O.H. & Olajubu, E. A. (2019). Performance comparison of threat classification models for cyber-situation awareness. Proceedings of the OAU Faculty of Technology Conference, 305-309.

Ozgur, A. & Erdem, H. (2016). A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. Peer Preprints, 4.

Sahu, S. & Mehtre, B. M. (2015). Network intrusion detection system using J48 decision tree[c]. International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2023-2026.

Sarker, H. Abushark. Y., Alsolami, F. & Khan, A. (2020). Intrudtree: a machine learning-based cyber security intrusion detection model. Symmetry, 12:754-761.

Sarker, H. (2019). A machine learning based robust prediction model for real-life mobile phone data. Internet of Things, 5:180-193.

Shams, E. A., & Rizaner, A. A. (2018). A novel support vector machine-based intrusion detection system for mobile ad hoc networks. Wireless Networks.

Stallings W. (2003). Cryptography and network security: principles and practices.

Thu, H. L., Kim, J., & Kim, J. (2016). Long short term memory recurrent neural network classifier for intrusion detection. 2016 International Conference on Platform Technology and Service (PlatCon).

Vladimir, V. & Corinna, C. (1995). Support-vector networks. Machine learning, 20(3):273-297.

Zahra, J., Mansour, S., & Ali, F. (2012). Intrusion detection using reduced-size recurrent neural network based on feature grouping. Neural Computing and Applications, 21:6.

Zhao, H., Feng, Y., Koide, H., & Sakurai, K. (2020). A sequential detection method for intrusion detection system based on artificial neural networks. *International Journal of Networking and Computing*, 10:213-226.