SEUSL

Sri Lankan Journal of Technology

Research

SLJoT
2022 Sp Issue 01 - 07
ISSN 2773-6970

# Detection of Intrusions in networks using Machine Learning Approach

Himaathri Premakumar[1], Amutha Kunarathinam[1] and Vaishali Ravi [1*]

[1]Dept. of Physical Science, University of Vavuniya., Sri Lanka

[*]Corresponding Author: rvaishali@vau.ac.lk ‖ ORCID: 0000-0002-5463-4071

Received: 1-10-2022 * Accepted: 25/10/2022 * Published Online: 30/11/2022

***Abstract*—** Since the inception of the Internet, data flowing over the communication networks has been subject to cyber-attacks. Intruders are increasingly threatening individuals' privacy because of the widespread usage of the Internet of Things, Social networking, and other major data-generating sources. As a result, researchers are working to develop Intrusion Detection Systems (IDS) that might mitigate the adverse effects of incursions. As technology and user behaviors evolve, attackers employ a variety of tactics to obtain access; as a result, recent research has identified Machine Learning (ML) as a key contributor to anomaly detection. The LightGBM , Support Vector Model (SVM), XGBoost, Random Forest (RF), K Nearest Neighbors (KNN), and Naïve Bayes classifiers are compared in this research to see which performs better in terms of computational time and accuracy. For the evaluation purpose, two datasets are utilized and several metrics such as True Positive Rate (TPR), False Negative Rate (FNR), True Negative Rate (TNR), False Positive Rate (FPR), and F-measure are assessed for the selected models.

***Keywords*—**Intrusion detection, Machine Learning, XGBoost, LightGBM

## I. Introduction

The threat to data privacy and assaults is increasing as a result of developing technological breakthroughs, particularly in social media and cloud computing. In this current communication environment, network and system security plays a major role as network attackers can create many successful attempts to crash the network and systems by intruding them maliciously. To prevent the network and systems from the hacker's intrusions, Intrusion Detection System which is known as IDS are used. Infringements on networks are usually referred to as intrusions, and intrusion detection systems (IDS) are designed to identify and prevent such attacks (Tiwari, Mohit Kumar, Raj Bharti, Akash Kishan, Jai, 2017). The Intrusion Detection monitors the computer system and networks and analyzes for the possibility of an intrusion attack. Intrusion Detection System is an important tool in the cyber security view, which is used to monitor the intrusion and find whether an intrusion attack may happen or not (Gupta and Megha, 2015). Most businesses urge adding intrusion detection systems to their working platforms due to the sharp rise in serious network attacks. With technological advancement, the intrusion of hackers has been increased and a solid way to find out or predict an attack has become a need, which challenges the data transmission of packets over the networks (Megantara, Achmad Akbar, Ahmad, Tohari, 2021). IDS is classified into three main types as network IDS, host IDS and Application IDS, where each type has different parameters to measure the attack. Network IDS monitors network packets to detect intrusion attacks and host IDS monitors a single host either a server or a client. Application IDS finds the intrusion by knowing the high risk applications found (M. Almi'ani, A. A. Ghazleh, A. Al-Rahayfeh and A. Razaque, 2018). IDS uses a couple of approaches to find out whether such an attack has been happened or not.

In this paper, some machine learning algorithms are applied and compared for intrusion detections. Machine learning is a heuristic and statistical approach for problem solving that can train itself with a set of data and execute in future accordingly with the help of models. Machine learning techniques have several algorithms that have different approaches to solve a problem. For intrusion detection, this paper compares the effectiveness of the methods SVM, XGBoost ,Random Forest, KNN, LightGBM, and Naive Bayes.

### *A. Support Vector Machine:*

This method performs both classification and regression by constructing a hyper plane that optimizes clear class boundaries. Due to the feature space, SVM can perform even the complex classifications and the regressions with flexibility and effectiveness (Wang, 2005).

### *B. Random Forest:*

This algorithm also works for both classification and regression. This employs a series of decision trees in which each node carries a decision. This algorithm constructs a

set of decision trees with randomly selected features and the target predictions from each decision tree are counted (Breiman, 2001).

### C. K-Nearest Neighbors:

This is a non-parametric supervised machine learning classification method which is used for regression and classification. It assumes the similarity of the new given data and puts it into the category with closest similarity in the available categories. This is the simplest machine learning algorithm (Guo, Gongde Wang et al, 2003).

### D. Naive Bayes:

This is used for classification only with an assumption of independence among predictors. It acts with the Bayes theorem. For every class a prediction is made with the relevant data points. The class with maximum probability is evaluated as the suitable class (Rish and Irina, 2001). A presence of a particular feature in a class is unrelated to the presence of any other feature.

### E. Extreme Gradient Boosting (XGBoost):

This tree based algorithm is used for either classification or regression problems (Chen, Tianqi and Guestrin, Carlos, 2016).This suggests a prediction model in the form of weak prediction models, mostly decision trees. Similar to other boosting techniques, it constructs in stages and is generalized by enabling optimization of any differentiable loss function. This is a gradient boosting decision tree algorithm which adopts ensemble technique that combines the individual models that are known to be weak learners and produces a better model as a final model (Hu, Ting, Song and Ting, 2019). This is widely used for applied Machine Learning for better performance and speed.

### F. LightGBM

This is also a gradient boosting algorithm that can work for classification and regression (Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu, 2017). This is similar to XGBoost that is known to be fast and high performance (McCarty, D.A.; Kim, H.W.; Lee, H.K., 2020).The difference is where the LightGBM splits its tree leaf-wise while other decision tree algorithms split depth-wise which then chooses the highest yielding leaf. It is based on decision tree algorithms and used for ranking, classification and other machine learning tasks. Scalability and performances are the major two criteria for the developing method. This algorithm applies the techniques called Exclusive Feature Bundling (EFB) and Gradient-Based One-Side Sampling (GOSS). These techniques aid this algorithm to run faster while maintaining a high level of accuracy (Ustuner, M.; Balik Sanli, F, 2019).

Machine Learning has been utilized for the intrusion detection over the recent decades that improves and strengthens the system with the efficacy. Since, the severity of attacks developing, an up-to-date IDS should be developed.

## II. Methodology

Intrusion Detection Systems are used to monitor the packets transfer in a network to identify the malicious and take actions accordingly. This monitoring system for finding abnormal activities and unauthorized access should be real time. Therefore, a real time system should be used, that is a machine learning model. To train a model, a set of steps need to be followed such as data collection, date preprocessing, feature selection, training and testing the model, and validating it (M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, 2017).The flow diagram of the IDS is shown in Figure 1 below.

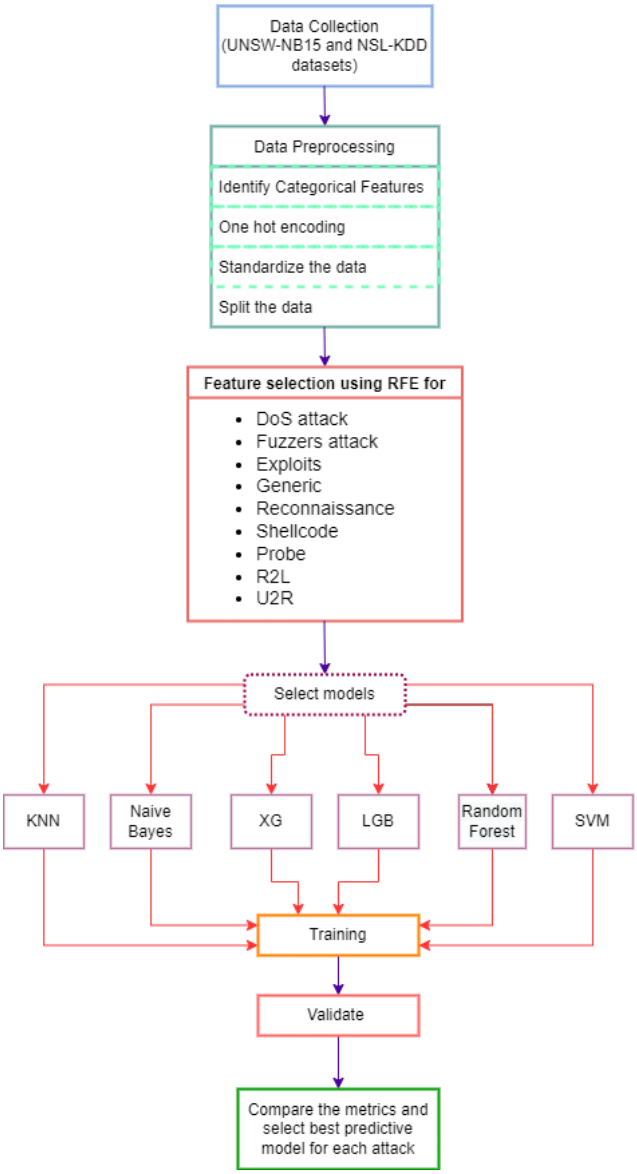


Figure 1: Flow diagram of the IDS

### A. Data Collection

Datasets are a collection of instances that all share a common attribute. To teach machine learning algorithms how

to execute various tasks, training datasets must be given into the algorithm first, followed by validation datasets (or testing datasets) to check that the model correctly understands the data. Machine learning data analysis use algorithms to improve itself over time, but good data is required for these models to function properly.

According to the studies, researchers use KDDCup99, NSL-KDD, ISCX 2012 and UNSW-NB15 to detect signature based and anomaly threats. To train and test the model, two datasets are selected considering efficiency of models relies on quality of data. One is the UNSW-NB15 dataset which was recently created for modern attacks (Moustafa, Nour, and Jill Slay, 2015) and the other one is the NSL-KDD dataset.

KDDCup99 and NSL-KDD are under the same territory or kingdom (C. H. Low, "NSL-KDD dataset", no date). The NSL -KDD is actually an improved variant of KDDCup99, which was developed ten years back by distributing a variety of attacks and avoiding data redundancy. Moreover, ISCX 2012 dataset has no classification of attacks specifically and this ISCX has created a new dataset as CICIDS2017 (Panigrahi, Ranjit  Borah, Samarjeet, 2018). Therefore we have chosen UNSW-NB15 which contains 47 features.

for nine types of attacks namely DoS, Fuzzers, Analysis, Backdoors, Exploits, Generic, Reconnaissance, Shellcode and Worms where we selected only six attacks which are DoS, Fuzzers, Exploits, Generic, Reconnaissance and Shellcode .In NSL-KDD which has 41 features and 4 attacks: Denial of Service (DoS), Probe, Remote to Local (R2L) and User to Root (U2R).

### *B. Data Pre-processing*

Data preprocessing is important to make more useful of the raw data because the quality data lead to quality decisions of top dimensions. This is the point where the computational activity begins. The datasets have redundant and categorical features that have to be improvised. The features will undergo several processes as follows (Agarwal and Vivek, 2015).

*1) Identify the categorical features :* The features that have categorical values are known as categorical features. There are several categorical features that have to be identified and encoded.

*2) One Hot encoding:* Initially, the dataset should be prepared for training and testing. For that purpose, categorical features must be transformed into binary vectors. As a first step, mapping categorical features to numerical values occur and then they are converted into binary vectors.

*3) Standardize data:* It is important to standardize data in a way where the mean is 0 and the deviation is 1.

*4) Split the dataset:* After performing standardization, according to the number of attack types the UNSW-NB15 dataset and NSL-KDD dataset split into 6 and 4 respectively. There is a need to represent the output variable. If any discrete variable is encountered or any integer categorical value is encountered then it is assigned with a suitable integer (Alasadi and Bhaya, 2017).Non-numerical attributes should be appropriately represented numerically. Appropriate integer values are assigned to discrete variables.

### *C. Feature Selection*

Feature selection is a very crucial part for the machine learning model, because the performance of the model relies on the features selected to train the model. Feature selection is the process where a subset of most relevant features from the whole set is extracted to train the machine learning model.The purpose of the feature selection is to select the informative attributes from the entire set of features and remove the unnecessary surplus data that does not make an impact on the effectiveness of the model. Irrelevant or partially relevant features can negatively impact on model performance (Miao, Jianyu  Niu, Lingfeng, 2016). High accuracy can be achieved by feature selection in comparison to selecting all the features of the model.

Recursive Feature Elimination (RFE) is a wrapper-style feature selection algorithm that also uses filter-based feature selection internally, which means that a special machine mastering algorithm is given and used within the core of the approach, wrapped by RFE, and used to assist choose features (Chen, Xuewen and Jeong, Jong Cheol, 2008). this is in contrast to filter out-based feature choices that rating each function and select the ones functions with the most important or smallest score. For each training model 12 features are selected by using RFE algorithm. But in total, 49 and 41 input features for UNSW-NB15 and NSL-KDD respectively. After twelve features are selected, models are trained for each attack type and then tested. The tested results are validated to find the effectiveness of each model and find the suitable algorithm. Figure 2 represents the features selected for attacks in NSL-KDD Dataset and Figure 3 represents the features selected for attacks in UNSW-NB15 Dataset.

## III. Results and Discussion

After training the model, the testing is performed to check the accuracy of the predictive models. The classification is the process in which the predictive model classifies the packet as whether it is a benign packet or malicious one. For the purpose of producing an effective predictive model, several algorithms are used to train different models and test them for performance. The experimental set up is performed with two different datasets which are UNSW-NB15 and NSL-KDD datasets, and six machine learning algorithms such as Support Vector Machine (SVM), K Nearest Neighbors, Random Forest, XGBoost, Naive Bayes, and LightGBM. The UNSW-NB15 dataset model is used to classify six attacks which are DoS, Fuzzers, Exploits, Generic, Reconnaissance and Shellcode, and the NSL-KDD dataset is used to classify four attacks such as DoS, Probe, R2L and U2R.

The predictive models are built with 12 features from the dataset for each attack that are more suitable for the prediction. This feature selection is performed by RFE. Then the selected features are trained with different algorithms

| DoS | Probe | R2L | U2R |
|---|---|---|---|
| ['src_bytes', 'dst_bytes', 'wrong_fragment', 'num_compromised', 'count', 'same_srv_rate', 'diff_srv_rate', 'dst_host_serror_rate', 'Protocol_type_icmp', 'service_ecr_i', 'flag_S0', 'flag_SF'] | ['src_bytes', 'dst_bytes', 'same_srv_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'dst_host_rerror_rate', 'service_private', 'flag_SF'] | ['duration', 'src_bytes', 'dst_bytes', 'hot', 'logged_in', 'is_guest_login', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'service_ftp_data'] | ['duration', 'src_bytes', 'dst_bytes', 'hot', 'num_compromised', 'root_shell', 'num_file_creations', 'count', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_diff_srv_rate'] |

Figure 2: Feature selected for attacks in NSL-KDD Dataset.

| DoS | Fuzzer | Exploits | Generic | Reconnaissance | Shellcode |
|---|---|---|---|---|---|
| 'Djit', | 'sttl', | 'sttl', | 'smeansz', | 'Sload', | 'ackdat', |
| 'Stime', | 'Sload', | 'dttl', | 'dmeansz', | 'Dload', | 'ct_state_ttl', |
| 'Sintpkt', | 'dtcpb', | 'Sload', | 'tcprtt', | 'smeansz', | 'ct_srv_src', |
| 'Dintpkt', | 'smeansz', | 'Dload', | 'synack', | 'dmeansz', | 'ct_srv_dst', |
| 'tcprtt', | 'dmeansz', | 'smeansz', | 'ackdat', | 'Sintpkt', | 'ct_dst_ltm', |
| 'synack', | 'Djit', | 'dmeansz', | 'ct_state_ttl', | 'tcprtt', | 'ct_src_ltm', |
| 'ackdat', | 'Ltime', | 'Sintpkt', | 'ct_srv_src', | 'ackdat', | 'dstip_149.171.126.12', |
| 'ct_state_ttl', | 'Sintpkt', | 'synack', | 'ct_srv_dst', | 'ct_state_ttl', | 'dsport_9223', |
| 'ct_srv_src', | 'Dintpkt', | 'ct_state_ttl', | 'ct_dst_ltm', | 'ct_srv_src', | 'dsport_9243', |
| 'ct_srv_dst', | 'synack', | 'ct_srv_src', | 'proto_arp', | 'ct_srv_dst', | 'dsport_9424', |
| 'proto_icmp', | 'ct_state_ttl', | 'ct_srv_dst', | 'proto_icmp', | 'dsport_10378', | 'proto_icmp', |
| 'proto_tcp' | 'proto_arp' | 'proto_ospf' | 'proto_ospf' | 'proto_icmp' | 'proto_ospf' |

Figure 3: Feature selected for attacks in UNSW-NB15 Dataset.

and the performance is illustrated in a confusion matrix. On the basis of the confusion matrix, it can be concluded that XGBoost and LightGBM are the most suitable algorithms for intrusion detection. Both algorithms are optimized versions of Gradient boosting library; either one of these algorithms can be used for a better Intrusion Detection System.

The trained models with different algorithms are tested for accuracy and the result for each algorithm differs. Most of the models could obtain more than 99%. But the accuracy for all the attacks were not validated above 99% for all the models except XGBoost and LightGBM. Most of the models were able to accurately detect generic attacks; however Naïve Bayes algorithm is the least accurate algorithm in predicting the attacks. The confusion matrix is shown in "Fig.4" below.

Prediction is performed on the trained model for all the attacks using the selected dataset. The prediction of each model is represented as a confusion matrix. Confusion matrix is the method for evaluating the efficiency of the trained models. By using the confusion matrix, recall, precision, accuracy, and F1-score can be calculated. These metrics are measured based on the following terms.

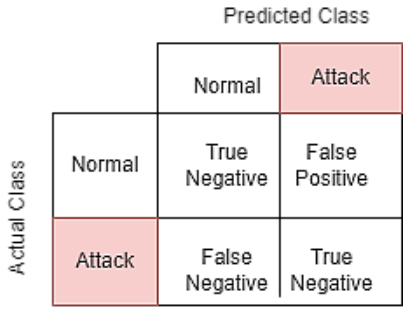


Figure 4: Confusion Matrix

### A. *True Positive (TP):*

This defines the correctly evaluated true values. In other words the actual values are true and the predicted values also true.

### B. *True Negative (TN):*

This term defines the correctly evaluated false values. In other words, the actual values and the predicted values are false.

### C. False Positive (FP):

This defines when the actual value is false but it is predicted as true.

### D. False Negative (FN):

This term is used when the actual is true but predicted as false.

Accuracy is the metric that is most commonly used to measure the performance (Hossin, Mohammad and M.N, Sulaiman. 2015). This is the metric that measures the correctly predicted values out of entire classes. The formula for accuracy is shown in Equation (1).

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{1}$$

With the performance measures of the Accuracy, XGBoost and LGB models showed highest accuracy in all the attacks except for DoS in both the UNSW-NB15 and NSL-KDD data sets. The accuracy for 6 attacks in UNSW-NB15 and 4 attacks in NSL-KDD datasets is shown in Figure 5 and Figure 6 respectively. Since, the Naïve Bayes model yields comparatively very poor accuracy scores, the figures for accuracy have refrained from illustrating the performance of that model. Apart from Naïve Bayes model, other models produces approximately equal values, which are plotted using a line graph to demonstrate and identify the optimal model.

Certainly one of some other overall performance metric this is broadly evaluated is Precision. it's far the ratio of successfully expected high-quality values from the entire predictive tremendous values (Hossin, Mohammad and M.N, Sulaiman. 2015) as follows.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

While analyzing the precision measures, XGBoost and LGB models showed 100% Precision in 4 out of 6 attacks in the UNSW-NB15 data set, where it deviated with only by 0.1% and 0.2% in Dos and Fuzzers attacks respectively .With the NSL-KDD Data set, XGBoost and LGB models achieved the second highest precision for Dos and Probe attacks. Recall is the metric used to measure the correctly predicted positive values from actual true classes (Hossin, Mohammad and M.N, Sulaiman. 2015) is shown below.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Recall of LightGBM and XGBoost was evaluated to be the model which showedhighest performance among all the other models with the NSL-KDD Data set, with all the 4 other attacks except DoS. According to the UNSW-NB15 data set LightGBM and XGBoost models perform better than SVM, KNN, and Naïve models where they provide second highest results for all the 6 attacks.LightGBM performs best among the other algorithms in Exploits, DoS and Generic attacks with a 100% Recall. More false negatives are undetected in XGBoost compared to LightGBM.

$$F1\ score = \frac{2 \times Recall \times Precision}{(Recall + Precision)} \tag{4}$$

Precision and recall are metrics to minimize false positive rate. XGBoost and LightGBM perform the same in all attacks except for Dos and Fuzzer, where all the other attacks yield 100% with UNSW-NB15 data set. Similarly with NSL-KDD data, XGBoost and LightGBM perform better with all the four attacks compared with the other models.

After the comparison of all the 6 Machine Learning algorithms with the UNSW-NB15 and NSL-KDD data sets, the results show XGBoost and LightGBM models perform better in this Intrusion Detection System Analysis. These both models' performance are nearly equal to the RF model in Precision and Recall but, shows highest accuracy measures than RF model.

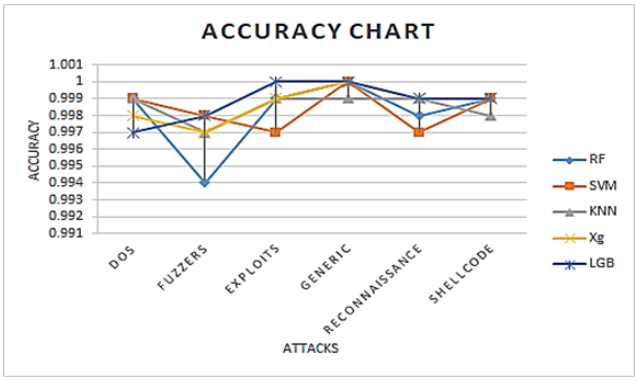


Figure 5: Accuracy obtained for the attacks in UNSW-NB 15 dataset by using Machine Learning Algorithms

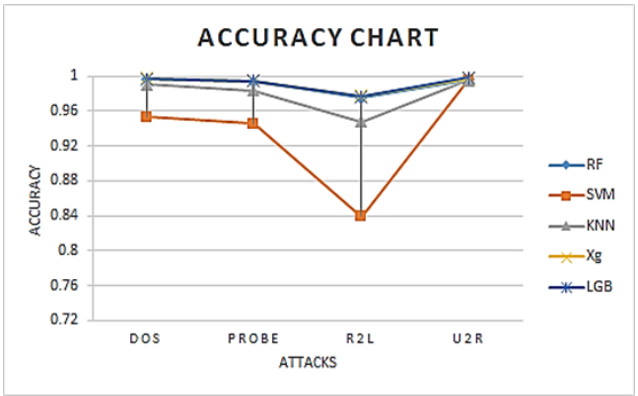


Figure 6: Accuracy obtained for the attacks in NSL-KDD dataset by using different Machine Learning Algorithms.

## IV. Conclusion

Intruders target the computer system and networks by using sophisticated techniques. Due to the advancement in technology, the severities of attacks keep increasing; so, effective IDS has to be developed. Intrusion Detection Systems are real time systems that have to be effective in order to detect the malicious packets, abnormal activities

and protect the network from intruders. In recent history, machine learning is an exponentially developed area for effective performance. There are many algorithms in machine learning, in that this paper discusses the Accuracy, Recall and Precision of these models with comprehensive experiments and analyses to prove that XGBoost and LightGBM are highly effective and showing higher accurate predictions for the attacks.

## REFERENCES


Agarwal, V. (2015). Research on data preprocessing and categorization technique for smartphone review analysis. *International Journal of Computer Applications, 131*(4), 30-36.

Alasadi, S. A., Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences, 12*(16), 4102-4107.

Aljawarneh, S., Aldwairi, M., Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science, 25*, 152-160.

Almi'ani, M., Ghazleh, A. A., Al-Rahayfeh, A., Razaque, A. (2018, April). Intelligent intrusion detection system using clustered self organized map. In 2018 *Fifth international conference on software defined systems (SDS)* (pp. 138-144). IEEE.

Almseidin, M., Alzubi, M., Kovacs, S., Alkasassbeh, M. (2017, September). Evaluation of machine learning algorithms for intrusion detection system. In 2017 IEEE *15th International Symposium on Intelligent Systems and Informatics (SISY)* (pp. 000277-000282). IEEE.

Anwer, H. M., Farouk, M., Abdel-Hamid, A. (2018, April). A framework for efficient network anomaly intrusion detection with features selection. In 2018 *9th International Conference on Information and Communication Systems (ICICS)* (pp. 157-162). IEEE.

Breiman, L. (2001) Random Forests. Machine Learning, 45, 5-32. https://doi.org/10.1023/A:1010933404324

C. H. Low, NSL-KDD dataset, Retrieved from https://github.com/defcom17/NSL-KDD.

Chen, Tianqi Guestrin, Carlos. (2016). XGBoost: A Scalable Tree Boosting System. 785-794. 10.1145/2939672.2939785.

Chen, X. W., Jeong, J. C. (2007, December). Enhanced recursive feature elimination. *In Sixth International Conference on Machine Learning and Applications (ICMLA 2007)* (pp. 429-435). IEEE.

Dina, A. S., Manivannan, D. (2021). Intrusion detection based on Machine Learning techniques in computer networks. *Internet of Things, 16*, 100462.

Elhag, S., Fernández, A., Bawakid, A., Alshomrani, S., Herrera, F. (2015). On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems. *Expert Systems with Applications, 42*(1), 193-202.

Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K. (2003, November). KNN model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 986-996). Springer, Berlin, Heidelberg.

Gupta, M. (2015). Hybrid intrusion detection system: Technology and development. *International Journal of Computer Applications, 115*(9), 5-8.

Hossin, M. (2020). Sulaiman,". A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION EVALUATIONS," IJDKP) *Int. J. Data Min. Knowl. Manag. Process, 5*(2).

Hu, T., Song, T. (2019, October). Research on XGboost academic forecasting and analysis modelling. *In Journal of Physics: Conference Series* (Vol. 1324, No. 1, p. 012091). IOP Publishing.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Khraisat, A., Gondal, I., Vamplew, P. (2018, June). An anomaly intrusion detection system using C5 decision tree classifier. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 149-155). Springer, Cham.

McCarty, D. A., Kim, H. W., Lee, H. K. (2020). Evaluation of light gradient boosted machine learning technique in large scale land use and land cover classification. *Environments*, 7(10), 84.

Megantara, A. A., Ahmad, T. (2021). A hybrid machine learning method for increasing the performance of network intrusion detection systems. *Journal of Big Data, 8*(1), 1-19.

Miao, J., Niu, L. (2016). A survey on feature selection. *Procedia Computer Science, 91*, 919-926.

Moustafa, N., Slay, J. (2015, November). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 *military communications and information systems conference (MilCIS)* (pp. 1-6). IEEE.

Panigrahi, R., Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. *International Journal of Engineering Technology, 7*(3.24), 479-482.

Rao, K. N., Rao, K. V., PVGD, P. R. (2021). A hybrid intrusion detection system based on sparse autoencoder

and deep neural network. *Computer Communications, 180*, 77-88.

Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

Subramanian, S., Srinivasan, V. B., & Ramasa, C. (2012). Study on classification algorithms for network intrusion systems. *Journal of Communication and Computer, 9*(11), 1242-1246.

Tiwari, M., Kumar, R., Bharti, A., Kishan, J. (2017). Intrusion detection system. *International Journal of Technical Research and Applications, 5*(2), 38-44.

Ustuner, M., & Balik Sanli, F. (2019). Polarimetric target decompositions and light gradient boosting machine for crop classification: A comparative evaluation. *ISPRS International Journal of Geo-Information, 8*(2), 97.

Wang, L. (Ed.). (2005). *Support vector machines: theory and applications* (Vol. 177). Springer Science Business Media.

Yerriswamy, T., & Murtugudde, G. (2021). An efficient algorithm for anomaly intrusion detection in a network. *Global Transitions Proceedings, 2*(2), 255-260.

Zaman, S., Karray, F. (2009, January). Features selection for intrusion detection systems based on support vector machines. In 2009 *6th IEEE consumer communications and networking conference* (pp. 1-8). IEEE.