UTILIZING ENSEMBLE LEARNERS HELP PREVENT UNAUTHORIZED ACCESS INTO IOT NETWORKS

N Venkateswaran[1], R Jegadeesan[2], Dava Srinivas[3], N Umapathi[4], G Karthick[5]

Article History: Received: 11.04.2023 **Revised:** 27.05.2023 **Accepted:** 13.07.2023

Abstract

Utilizing intrusion detection systems is necessary to safeguard information systems from attacker attacks. Many publicly accessible open-source assault datasets have been released in recent years so that academics and also researchers can evaluate the performance of different detection classifiers. These datasets contain a full collection of exemplary network features. This study, researchers look at the problem of Network-Based Intrusion-Detection System (NIDS) by employing the Bot-IoT dataset from the network based Internet of Things (IoT) to evaluate its usefulness of seven distinct Ensemble Learning Classifiers in terms of detection efficiency (ELCs). The outcomes of our trial demonstrated that CatBoost was the ELC that performed the most effectively with Effectiveness, Positive predictive value, F-Measure, Training and Test Time, despite the fact that all ELCs had excellent classification metric scores.

Keywords: Ensemble Learning, Network Based Intrusion Detection, CatBoost, Machine Learning techniques, Internet of things (IoT), Ensemble Classifiers.

[1]Assoc.Professor, Department of CSE, Jyothishmathi Institute of Technology and Science, Karimnagar, Telangana.
[2]Professor and Head, Department of CSE, Jyothishmathi Institute of Technology and Science, Karimnagar, Telangana.
[3]Assoc.Professor, Department of CSE, Jyothishmathi Institute of Technology and Science, Karimnagar, Telangana.
[4]Professor and Head, Department of ECE, Jyothishmathi Institute of Technology and Science, Karimnagar, Telangana.
[5]Assoc.Professor, Department of ECE, Jyothishmathi Institute of Technology and Science, Karimnagar, Telangana.

Email: [1]venkywn@gmail.com, [2]ramjaganjagan@gmail.com,[3]dr.dsrinivas@jits.ac.in, [4]nrumapathi@gmail.com, [5]karthick.sgs@gmail.com.

DOI: 10.31838/ecb/2023.12.s3.669

1. Introduction

Internet of Things (IoT) is one of the newest networking trends (IoT). It can be defined as the "interconnection of things" with constrained computational capabilities. Without the need for computer-to-computer or person-to-person communication, It is capable of internet data transmission and reception. [1].

Because of the rapid expansion of information technology, huge amounts of data are becoming more and more ingrained in our daily lives. According to a Cisco prediction, IP traffic will increase from 122 EBs(Exabytes) monthly in 2018 to 415 EBs monthly in year 2022 [2]. A rise in network traffic has resulted in an increase in the variety and quantity of cyber attack-related threats. An unauthorized try to use threats, impair, harm, severely affect, violate, or otherwise adversely impact the information assets of another party is popularly referred to as a "cyber attack." Network intrusion detection systems (NIDS) are becoming a common tool used by enterprises to safeguard their network security. Even though they are necessary, security methods like firewall systems, virus scanning, encryption techniques, and identity authentication are insufficient to shield networks

and PCs from today's dangers. The aforementioned security steps, a Machine Learning (ML) techniques based strategy and intrusion detection systems (IDS) can all work together to solve these issues [3].

Along with the regular IoT, it's important to discuss resource-constrained IoT devices. Despite the fact that these gadgets use IoT based systems or applications, they are battery-operated, compact, low-power gadgets with a variety of design compromises. They also have restricted computational and storage capacities. Simply put, endnote sensors are devices with limited resources that are utilised for a specific purpose. These gadgets have limited storage and processing power, restricted energy due to energy-vulnerable batteries, and other resource limitations. They also communicate without human engagement, via weak radio networks, and over low-power lossy links [4].

In order to identify latent data anomalies, Network traffic is continuously tracked and analysed by an IDS. Depending on their approach to detection, IDSs can be categorised into two groups [5]. The first kind of intrusion detection recognises and categorises network intrusion attempts using specified attack signatures. Therefore, this method is barely able to identify fresh and new attacks [6]. Anomaly-based detection, on the other hand, can uncover risks that were previously unnoticed by applying machine learning techniques to search network data for anomalies. A typical behaviour or circumstances are called anomalies.

Decades of research [7]–[10] have concentrated on improving the precision and effectiveness of IDSs. Due to its promising reliability, The majority of IDS research nowadays is focused on anomaly-based IDS, It has been widely employed.

In recent years, Machine_Learning (ML) techniques for intrusion detection have included Random Forest , Neural Networks, Decision Tree and Support Vector Machine (SVM). Each algorithm, however, has benefits and drawbacks. Classifiers that are effective at spotting one type of attack might not be effective at spotting another. There are several restrictions regardless of the methods for pre-processing data or choosing features for the classifiers, according to several recent research studies [11]–[13]. To improve its effectiveness, IDS's ML architecture is continuously developed into classifiers that are more complicated. An illustration of how to increasing the accuracy and coherence of intrusion detection is the use of Ensemble Learning Classifiers (ELCs) for classification. The strategy has gained more traction than using a single classifier. By combining them, tt creates a strong learner who can make up for the shortcomings of the subpar classifiers. Since they produce superior performance than single classifiers, ELCs are preferable solutions.

To evaluate the effectiveness of multiclass classification, IoT researchers employed the

Bot-IoT dataset [14]. Seven Ensemble Learning Classifiers are evaluated for their performance in this work. The ELCs used in this experiment are CatBoost, SVM, SGB, Random Forest, LightGBM, and XGBoost. This study's two main objectives are speed and detection efficacy. Security systems were among the Internet of Things (IoT) applications examined in our research and also home automation, intelligent buildings, and intelligent transportation systems. As comparison to conventional network-based applications, the security protection provided by the IoT is still insufficient. On the IoT dataset network categories, we evaluate the detection capabilities of the five ELCs [15]. According to the results of our investigation, Comparing CatBoost to other ELCs, it can deliver the best classification results. CatBoost fared better than the other ELCs in our investigation since it trained and tested the dataset the quickest.

The remainder of this journal entry is divided into the following sections. The second section gives a brief introduction of several recent NID ensemble ML classifier efforts. In Part III, we introduce the IoT sample dataset, ensemble learning classifiers, research methodology, and experiment design. Section IV focuses on comparing the performance results of different ELCs. Section V's final section, which finishes this investigation, has some significant observations.

Related Works

Data mining as well as machine learning are combined into one process using the "Ensemble Learning" approach [16]. To categorise weak learners, this study used ensemble learners, a group of poor learners' individual learning approaches. Classifiers with limited learning capacity are brought together and collectively educated to enhance classification results. [17]. As ELCs have generated optimal outcomes that are more spectacular than those of single learning classifiers, the vast majority of researchers and academics involved in ML-related research are currently using them in their particular work. By utilising ensemble learning approaches, In order to provide more safety through improved intrusion detection, ensemble learning techniques may be improved. Using free research datasets, the authors Verma et al. [18] conducted research comparing the results of single learning and ensemble learning classifiers. The output of the ensemble learners was substantially superior to that of the single learning classifiers. Though this study uses a wide variety of ELCs, its main goal is to do a comparative comparison. In the sections that follow, we look at some cutting-edge studies that used ELCs on the network datasets in their work. For classification in NID, Bansal and Kaur [19] investigated tuning based on XGBoost. The XGBoost intrusion detection classifier was used since it is reliable and effective. Distributed Denial of Service, XG_Boost, MLP, Ada_Boost, NB, SVM, NN and KNN investigators also looked towards classifiers for binary as well as multiclass classification techniques using the dataset [CICIDS-2017] (DDoS attack types only). The XG_Boost classifier correctly classified 91.37% and 99.55% of binary classes. The standard grade rate was high in both categories. For the KDD99 dataset, Obeidat et al[20] .'s evaluation of seven supervised ML models for multiclass classification to identify assaults. In the analysis, only 60K procedurally generated KDD99 test sets were used. With 93.79% accuracy, Random Forest outperformed the other options in the categorization research. J48, a Decision Tree classifier variation, scored higher on precision (93.12%) than Random Tree (90.59%). The author Larriva-Novo et al. [21] used the seven highest performers as their base learners in their examination of a class of independent learners' performance on the universal bench mark UNSW-NB15 dataset. The learners are combined for the final classification employing the XGBoost technique as the meta-learner. Using the Synthetic Minority Oversampling Method (SMOTE), the dataset records are additionally balanced for the evaluation, and duplicated features are eliminated using the Kendall’s Rank correlation coefficient feature selection method. The study demonstrates enhanced performance following SMOTE dataset record balancing. Rajadurai and Gandhi [22] used the popular NSL-KDD based dataset to prove that ensemble classifiers are effective and are also capable of reliably detecting network intrusions. The proposed ensemble classifier's basic learner is composed of RF and gradient boosting classifiers. The ensemble classifier successfully classified 91.16% of the cases as a result. The classifier's capacity to identify abnormalities decreased in other network categories, nevertheless, due to low recall and detection rates. Shi et al. provide a stacked ensemble learner using a feature selection approach. The KDD99 and NSL-KDD bench mark datasets are utilised for testing and the Extreme Tree Classifier and QDA are integrated to deliver their learning outcomes. Their test results show that the recommended learner outperformed other classifiers and consistently performed well on both datasets. Furthermore, the implementation of feature selection reduces the learner's development time. But for assessing the ELC, non-IoT datasets were utilised. The majority of the research has been concentrated on categorising non-IoT datasets, with numerous algorithms reporting significant calculation durations and high false-positive rates. In some studies, neither the duration of the experiment nor the categorization task type are mentioned. Additionally, we found that most studies [23]–[25] with the Bot-IoT dataset was used to assess the performance of multiple ML classifiers used in binary or other 5-class categories. In this study, we classify cases from the Bot-IoT dataset using a variety of ELCs in order to assess their classification precision as well as their training and testing timeframes.

Detection of Iot Network Intrusion Using Ensemble Learner

We classify cases from the Bot-IoT dataset using a variety of ELCs in order to assess their classification precision as well as their training and testing timeframes.

A. Pre-processing and the Dataset

For our analyses, we use the Bot-IoT dataset [13] developed by Koroniotis et al. The dataset is made up of four Excel CSV files including the testing and training occurrences. Examples bench mark dataset are given the category labels as "attack" (Network_Based classes), named as "category" (five based network classes), and named as "subcategory" are 11 network classes. The dataset also contains forty three (43) network based features. The Table I display the distribution of the network based benchmark dataset samples employed in this investigation. In the Cyber Range Lab located at UNSW Canberra, the bench mark dataset Bot-IoT was created utilising actual and simulated IoT network traffic as well as other types of attacks. A realistic test bed infrastructure with common and uncommon botnet discrepancies was created to accomplish this, in order to gather enormous volumes of networked data (Network Denial-of-Service, Information Acquisition and Identity Theft). DDoS-HTTP (DDH), Data Exfiltration (DEx), OS Fingerprint (OSFP), DoS-HTTP (DHTTP), Keylogging (KLG), and the variants' subclass anomaly types were service-based scans (SES). The majority of network data is composed of DDoS-UDP (DDU), DoS-UDP (DU), DoS-TCP (DTCP), and NDoS-TCP (NDDT), with the remainder being consisting of all additional network-based data.

Table-I Synopsis of The Dataset Occurrences

Main_Category	Sub_Category	Occurrences
Denial-of-Service (DoS)	DU DTCP DHTTP	1032961 615800 1485
Distributed DoS	DDU DDT DDH	577876 347751 988
Information Acquisition	SES OSFP	64281 17679
Identity Theft	KLG DEx	72 7
Non Attack	Non attack	476
Class Distribution Assault		Non Attack: 476 (0.021%)

	Abnormality 2,660,001 (99.98%)
Total Amount of All Occurrences	2,659,376

During preprocessing, we found that not every feature was required for network classification. Flags, proto feature, Daddr, pkSeqID feature, Saddr, and State in particular were deleted. The research found that flags, proto, and state features that all store the identical content as their respective flags, proto, and state numbers. Daddr_, pkSeqID feature & Saddr were discarded given that they are device-focused.

The subcategory name is converted into integer values between 0 and 10 to cover all 11 network categories, due to the fact that the network dataset's occurrences are all included in the scope of our analysis. Similar to the approach taken by Churcher et al. [24], the dataset was similarly divided into 80 and 20 percent, with 80% of the data being implemented for training and 20 percent being used to test the classifiers. Moreover, we apply min-max scaling normalisation, scaling up to an interval of 0 to 1, to lessen the skewness in the feature data.

B. Ensemble-Learning-Classifiers (ELCs)

Combining weak learners to produce strong learners is the fundamental tenet of ensemble learning classifiers [27]. At the first level of the ELC's two-tier classification system, base learners classify cases. After that, the meta-learner looks for and incorporates the results of the fundamental learners. The second-level classifier corrects the first-level losses before generating the final classification [28]. Three ensemble learning-based categorization algorithms are bagging, boosting, and stacking. In this paper, we concentrate on bagging and enhancing ELCs variations. Here is a brief summary of a few well-known ELCs.

1) AdaBoost: An iterative classifier called AdaBoost [29] combines a number of weak classifiers to produce a single robust classifier. This classifier's fundamental principle involves training several weak classifiers with the same training data samples. It modifies the sample weight before utilising the new data to train the subsequent weak classifier based on the results of each and every training session and the efficiency of the prior general classification techniques. AdaBoost classifier calculate the efficiency of the verdict's weak classifiers and combines them into a strong classifier. The iterative process comes to a conclusion when a specific set of requirements is met.

2) LightGBM: LightGBM is a Gradient-Boosted Decision Trees (GBDT) classifier that uses feature grouping and the gradient-based one-side sampling approach (GOSS) that is mutually exclusive (EFB) [28]. The lengthier training hour for the earlier GBDT classifier is mostly spent selecting the proper split point. LightGBM huge utilization of the histogram approach to choose features and choose segmentation points in order to resolve this issue. The initial continuous feature values are binned in this method, and the classifier is constructed utilising these bins. The histogram significantly reduces the amount of time needed to select split points and improves the training and prediction effectiveness of a classifier.

3) Random Forest: It is a collection of classification or regression trees that have not been pruned [30]. Today, In particular for large datasets with varied properties, it is the most precise data mining technique. Multiple categorization trees are generated by the random forest. Each tree is built utilizing a tree based classification classifier and a distinct bootstrap sample inherits from the final data. The forest is then built, and every tree is then given a new object that has to be categorised. The class of the instance is decided by a vote from each tree. For its final classification choice, the forest chooses the class with the most support for a given instance.

4) CatBoost: The open-source machine learning toolkit CatBoost [31], created by Russian search engine Yandex, was released in 2017. It is a member of the same family of boosters as the well-known XGBoost and LightGBM. It works effectively with textual, numerical, and categorical data and has a quick learning curve. The Boosting family classifier's gradient bias and prediction shift issues are addressed by CatBoost, which also increases prediction accuracy, which in turn partially resolves the overfitting issue. Discrete data can also be accurately and effectively analysed using CatBoost. Unlike earlier classifiers, it has functionality for visualising data and GPU support.

5) XGBoost: In 2014, Tianqi Chen developed XGBoost [32], a modified version of GBDT that improves prediction performance and speed. Using Julia, R, Python, Hadoop, Scala, and Python, it is a scalable approach. A number of XGBoost's parameters help to reduce overfitting and boost performance as a whole. As a result, it offers precision, viability, and efficiency. It can be up to nine times faster than classical GBDT and can execute automatically in similarity on Windows and Linux environment.

6) Stochastic_Gradient_Boosting (SG_Boost) : This type of boosting is known as stochastic gradient boosting. At the start of each cycle, a random subset of training data is selected from the full training dataset (without replacement). The whole sample is subsequently omitted in favour of fitting the base learner using the randomly chosen subset.

7) SVM: Using supervised machine learning, classification or regression issues can be resolved. Your data are changed using a procedure known as the kernel trick, and the best output boundary is determined based on these alterations. Python was used to run all experimental simulations on a PC running 64-bit Windows, 6GB RAM memory and an Intel Core series processor with a minimum of 1.99 GHz clock speed are further PC features. Additionally, the Scikit-Learn package was used to generate the ML classifiers. This was completed using the Anaconda Navigator GUI platform.

Evaluation of Performance

The detecting capabilities of the ELCs used in our experiment are covered in this section. The seven ELCs that were examined in this article. Throughout the investigation, performance metrics including Accuracy, Positive Predictive value, Sensitivity, F measure, Testing and Training Time were used.

Table-Ii A Comparison of All Elc's in Detail (Metrics Are In %)

Name of the Classifier	Performance/ Accuracy	Positive Predictive value	Sensitivity	F measure	Train_Time (sec)	Test_Time (sec)
Ada_Boost	99.92	81.73	81.73	81.67	734.23	14.11
Light_GBM	96	43.77	43	43	575.41	66.30
Random Forest	98.93	52.21	53.12	52.62	279.21	9.27
Cat_Boost	99.98	99.88	99.74	99.82	229.42	1.83
XG_Boost	99.98	99.71	99.83	99.76	498.87	24.75
SVM	97.85	61.66	54.14	54.98	296	54.85
SGBoost	98.22	49	49.22	49.45	355.27	39.87

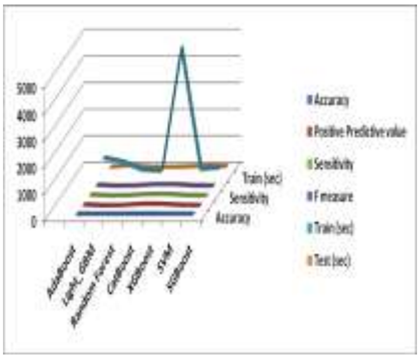

Figure: 1 Comparison of the 7 ELCs' respective times (metrics are in sec).

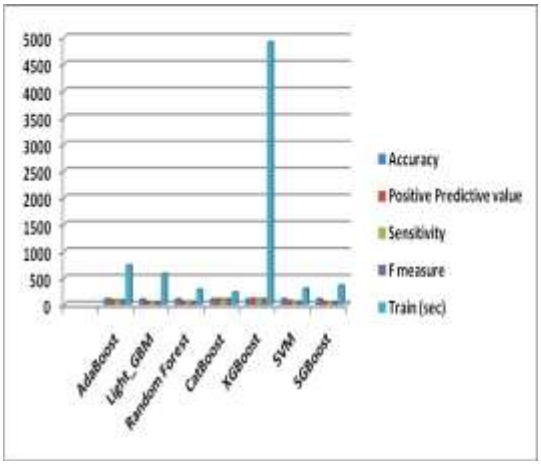

Figure: 2 Testing Time comparisons of ensemble learners (metrics are in sec)

On an IoT dataset, we evaluated the effectiveness of seven machine learning classifiers from ELC to ascertain their effectiveness and detection speed. The five ELCs' performances are summarised in Table II. The results showed that over 95% of the network instances could be successfully classified by all of the assessed classifiers, demonstrating their overall classification accuracy. The CatBoost algorithm can be taught faster than Random Forest, which takes more than four minutes. The XGBoost classifier requires about one hour and 48 minutes to train, but the prediction test only needs 25 seconds.

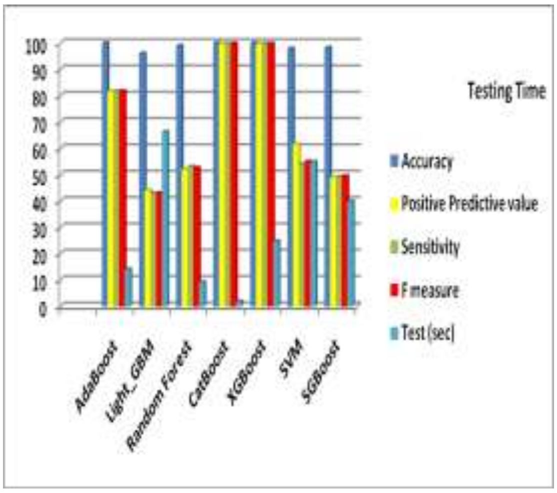

Figure: 3 Testing Time (metrics are in sec)

We come to the conclusion that the CatBoost classifier is the fastest and when it comes to training time, XGBoost is the slowest classifier. The inference time or prediction test time is important since intrusion detection systems frequently operate in real-time. The performance of the entire network will suffer from a classifier with a long prediction time. Despite outperforming LightGBM, the Cat_Boost and XG_Boost classifiers obtain the best level of accuracy. Light GBM, the Model with the lowest scores across all measures with the exception of training time, in terms of classification metrics, performed the worst. CatBoost fared better than the other classifiers in our experiment in terms of training and evaluation times. We chose CatBoost as the best classifier for the benchmark IoT dataset for its rapid testing and training schedules and higher predicted efficiency. In terms of overall performance, CatBoost ELC is superior than cutting-edge ELCs in terms of effectiveness and efficiency when it comes to identifying IoT network breaches. The classifier's resilience, decreased need for intensive hyperparameter tuning, decreased likelihood of overfitting, and ability to deliver cutting-edge classification results are some of its benefits. Additionally, because it can handle categorical information automatically, the pre-processing time is shorter. The classifier beat other ELCs in our trial in terms of classification accuracy, which is explained by these advantages. Therefore, in our opinion, deleting pointless properties as a result of the output variable should shorten testing and training times while keeping detection performance unchanged.

2. Conclusion

This research examined the performance of seven ensemble learning classifiers for anomaly detection on an Internet of Things (IoT) benchmark dataset. According per our findings, CatBoost performs better than seven other ELCs in all eleven categories of the multiclass network. Of the seven ELCs, CatBoost had the quickest training and testing times as well as the highest overall classification metric scores. To further our understanding of the trade-off between the demands placed on computer resources during training and testing, we propose extending our research by testing several feature selection techniques on the dataset. This would improve categorization performance much further. To further assess the effectiveness and efficiency of the seven classifiers, they can also be tested on different IoT datasets that contain a wider variety of assaults.

3. References

1. Albulayhi, K.; Smadi, A.A.; Sheldon, F.T.; Abercrombie, R.K. IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. Sensors 2021, 21, 6432.
2. C. Whitepaper, ”Cisco Visual Networking Index: Forecast and Trends,2017–2022,” Cisco, 2018.
3. Natesan. V and K. Umadevi, "Hybridized wrapper filter using deep neural network for intrusion detection," Computer Systems Science and Engineering, vol. 42, no.1, pp. 1–14, 2022.
4. Y. Tang and S. Chen, “An automated signature-based approach against polymorphic internet worms,” IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 7, pp. 879–892, 2007.
5. M. Hasan, M. M. Islam, M. I. I. Zarif, and M. Hashem, “Attack and anomaly detection in iot sensors in iot sites using machine learning approaches,” Internet of Things, vol. 7, p. 100059, 2019.
6. S. Prabaharan and at el "An Efficient Neuro Deep Learning Intrusion Detection System for Mobile Adhoc Networks." EAI Endorsed Transactions on Scalable Information Systems 9.6 (2022): e7-e7.
7. N. K. Sahu and I. Mukherjee, “Machine learning based anomaly detection for iot network:(anomaly detection in iot network),” in 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184). IEEE, 2020, pp. 787–794.
8. D. K. K. Reddy and H. Behera, “Catboosting approach for anomaly detection in iot-based smart home environment,” in Computational Intelligence in Data Mining. Springer, 2022, pp. 753–764.
9. O. Almomani, M. A. Almaiah, A. Alsaaidah, S. Smadi, A. H. Mohammad, and A. Althunibat, “Machine learning classifiers for network intrusion detection system: comparative study,” in 2021 International Conference on Information Technology (ICIT). IEEE, 2021, pp. 440–445.
10. H. Azizan, S. A. Mostafa, A. Mustapha, C. F. M. Foozy, M. H. A. Wahab, M. A. Mohammed, and B. A. Khalaf, “A machine learning approach for improving the performance of network intrusion detection systems,” Annals of Emerging Technologies in Computing, vol. 5, no. 5, pp. 201–208, 2021.
11. S. Latif, F. F. Dola, M. Afsar, I. J. Esha, and D. Nandi, “Investigation of machine learning algorithms for network intrusion detection.” International Journal of Information Engineering & Electronic Business, vol. 14, no. 2, 2022.

12. [12] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, ”Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset.” Future Gener. Comput. Syst.2019.
13. M. A. Jabbar and R. Aluvalu, ”RFAODE: A novel ensemble intrusion detection system”, Procedia computer science, vol. 115, pp. 226-234, 2017.
14. Paterson, Colin & Calinescu, Radu & Ashmore, Rob. (2021). Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. ACM Computing Surveys. 54. 10.1145/3453444.
15. Verma, P., Anwar, S., Khan, S., & Mane, S.B. (2018). Network Intrusion Detection Using Clustering and Gradient Boosting. 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 1-7.
16. Bansal and S. Kaur, “Extreme gradient boosting based tuning for classification in intrusion detection systems,” in International Conference on Advances in Computing and Data Sciences. Springer, 2018.
17. Obeidat, N. Hamadneh, M. Alkasassbeh, M. Almseidin, and M. AlZubi, “Intensive pre-processing of kdd cup 99 for network intrusion classification using machine learning techniques,” 2019.
18. X. Larriva-Novo, C. Sanchez-Zas, V. A. Villagra, M. Vega-Barbas, and D. Rivera, “An approach for the application of a dynamic multi-class classifier for network intrusion detection systems,” Electronics, vol. 9, no. 11, p. 1759, 2020.
19. H. Rajadurai and U. D. Gandhi, “A stacked ensemble learning model for intrusion detection in wireless network,” NEURAL COMPUTING & APPLICATIONS, 2020.
20. X. Shi, Y. Cai, and Y. Yang, “Extreme trees network intrusion detection framework based on ensemble learning,” in 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA). IEEE, 2020, pp. 91–95.
21. D. D. Kulkarni, S. Rathore, and R. K. Jaiswal, “Intrusion detection system for iot networks using neural networks with extended kalman filter,” in 2021 International Conference on Computer Communications and Networks (ICCCN). IEEE, 2021, pp. 1–7.
22. [22] H. Tyagi and R. Kumar, “Attack and anomaly detection in iot networks using supervised machine learning approaches.” Rev. d’Intelligence Artif., vol. 35, no. 1, pp. 11–21, 2021.
23. S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, “Federated deep learning for zero-day botnet attack detection in iot edge devices,” IEEE Internet of Things Journal, 2021.
24. Churcher, R. Ullah, J. Ahmad, F. Masood, M. Gogate, F. Alqahtani, B. Nour, W. J. Buchanan et al., “An experimental analysis of attack classification using machine learning in iot networks,” Sensors, vol. 21, no. 2, p. 446, 2021.
25. S. Bagui, and K. Li, ”Resampling imbalanced data for network intrusion detection datasets.” J. Big Data 2021.
26. "A SURVEY ON ENERGY EFFICIENT USAGE OF INTRUSION DETECTION SYSTEM IN MOBILE AD HOC NETWORKS", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.6, Issue 4, page no.648-651.
27. Q. Wang and X. Wei, “The detection of network intrusion based on improved adaboost algorithm,” in Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, 2020, pp. 84–88.
28. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu LightGBM: a highly efficient gradient boosting decision tree Advances in Neural Information Processing Systems (2017), pp. 3146-3154.
29. L. Breiman, ”Random Forests”, Machine Learning 45(1):5-32, 2001. [30] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, M. Asadpour Boosting methods for multi-class imbalanced data

classification: an experimental review J Big Data, 7 (2020), p. 70.

30. Aishwarya, Ch, et al. "Intrusion detection system using KDD cup 99 dataset." International Journal of Innovative Technology and Exploring Engineering (IJITEE) 9.4 (2020): 3169-3171.
31. T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
32. R. Qaddoura, A. Al-Zoubi, I. Almomani, and H. Faris , ”A Multi-StageClassification Approach for IoT Intrusion Detection Based on Clustering with Oversampling.” Appl. Sci. 2021.