Comparison of Various Ensemble Machine Learning Techniques to Recognize the Instantaneous Internet of Things Assault

P Kalyanasundaram [1], S.Markkandeyan [2], N Sengottaiyan [3], J.Vijayakumar [4], H.Senthil Kumar [5]

[1]*Professor, Institute of Electronics and Communication Engineering,Saveetha School of Engineering, SIMATS, Chennai.*
[2]*Professor, Department of Information Technology, KSR Institute for Engineering and Technology, Tiruchengode.*
[3]*Professor, Gopalan College of Engineering and Management, Whitefield, Bangalore.*
[4]*Associate Professor, Department of Electronics and Instrumentation, Bharathiar University, Coimbatore.*
[5]*Assistant Professor, Department of Electronics and Communication Engineering,Gopalan College of Engineering and Management, Whitefield, Bangolore.*

[1]*kalyanasundaramp.sse@saveetha.com*, [2]*drsmkupt@gmail.com*, [3]*gcemprincipal@gmail.com*, [4]*vijayakumar@buc.edu.in*, [5]*halansenthil.ece@gmail.com*

Abstract.

The adoption of Internet of Things (IoT) devices in residences, workplaces, transit, hospitals, and other places has led to an increase in harmful assaults, which are becoming more common. As the space among IOT systems and fog machines is lower than the gap between IoT devices and the cloud, threats may be discovered more quickly. As a result of the massive amounts of data generated by IoT devices, machine learning is commonly employed to identify threats. It's a concern, though, because fog nodes may not have the computing or storage capacity to identify threats in a timely basis. Machine learning model creation and real-time forecasting may be offloaded from the cloud, and both tasks can be performed by fog nodes, according to this article. In the server, an ensemble machine learning method is created based on past data to identify assaults on fog nodes in real time. This method is used on the NSL-KDD database. In terms of numerous performance metrics, such as processing time, specificity, recall, efficiency, and the ROC (receiver operating characteristic) curve, the findings suggest that the proposed technique is successful.

Keywords. IoT; Data Transfer; Machine learning; Ensemble Learning

1. INTRODUCTION

In the past, only computer systems, cell phones, and tablets were associated with the Internet. A wide range of equipment and utilities (e.g., TVs, air-conditioning units, and washing machines) may now be linked over the Internet via the IoT. Healthcare, farming, traffic control, energy conservation, supply of water, unmanned aerial vehicles, and autos are just a few of the many industries using the Internet of Things (IoT). Figure 1 depicts a three-Level IoT Framework: (1) the thing layer (TL), (2) the fog layer (FL), and (3) the cloud layer. A wide range of IoT machines are included in the thing layer, covering home automation, healthcare, smart automobiles, smart drones, and smart buildings, among other applications. With data limits, computation, power, and storage, this layer allows for data collecting. Following the TL is the fog layer, which may include operational resources for managing real-time activities and making swift decisions. Data may be collected and processed in several data centres thanks to the cloud layer's support. It may take quite some time to implement choices in the TL since it is so far removed from the object layer.

The quantity of information created by IoT devices is growing from 18 zeta bytes in 2019 to 73 zeta bytes in 2025, as per a forecast from the International Data Corporation (IDC). Many new dangers arise from the huge inflow of data [1]. Due to the lack of energy, storage, or connectivity, IoT machines and connections tend to be unsafe since they cannot execute fundamental security operations such as encryption. According to IBM X-Force, the number of IoT assaults doubled in 2020. Malware and botnet assaults are putting IoT-enabled networks at danger of losing their anonymity and safety.

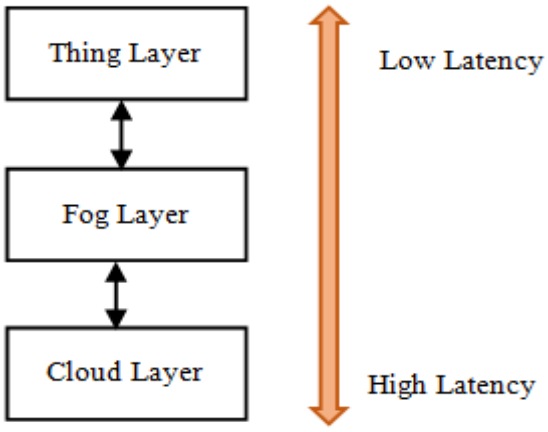

Figure 1: A Three-Level IoT Framework.

Authentication [2], identification, and protection are some of the security measures suggested for the IoT. Incorporating ML algorithms into the IoT might reduce safety and privacy problems [3]. Decisions on where to execute techniques for quick decision making are critical nowadays, whether they are on the cloud, fog or things layer. Any IoT choices may be postponed if

all machine learning judgments are done remotely. Because of the restricted bandwidth, processing power, and energy available at the object and fog layers, it may be challenging to implement ML solutions at these levels.
Deep learning methods are more successful in detecting IoT assaults than typical machine learning algorithms, according to recent study [4]. However, these techniques may only be able to be executed on the cloud layer. The system is designed to make real-time judgments quickly, however in certain cases, such as online surgeries, these methods don't work as well as they should. ML algorithms like support vector machine may only produce relevant results when they are used in conjunction with a feature matching strategy or optimization technique [5][6] in the context of IoT threats. The limited resource requirement cannot be met by this set of techniques. Applications like offline or non-interactive projections among tiny dataset use ML methods such as decision trees, nave Bayes, and K-nearest neighbours. Real-time forecasting is a problem for these algorithms, which are deemed poor.
An ensemble model for detecting IoT assaults is proposed in this research for an IoT system with restricted connectivity and computing power, power and memory. Attacks such as DoS, spoofing, and probes are all taken into consideration. To boost detection rates, no further feature extraction or dimensions reduction algorithms are used. This methodology is best suited for detecting IoT threats in real time and quickly. Step one is picking the best ensemble model that is fast and accurate, and step two is executing the best model so that the decision may be applied in a short period of time. Since picking the optimal ensemble model necessitates a lot of computing power, the first step is done in the cloud, while the second is done in the fog layer, where latency is minimal for real world applications.
The NSL-KDD database is used extensively in this study for data analysis studies. As a real-time representation of IoT assaults on a network, the dataset has been improved from KDD-99. The findings reveal a good precision in a short period of time and with the minimal number of resources required. Here are the sections of the paper's structure: This paper is organized as follows: Section 2 gives an overview of the relevant work, Section 3 outlines our technique, Section 4 outlines simulation scenarios, and Section 5 concludes with the findings.

2. RELATED STUDY

IoT devices and apps are hampered by security flaws that prevent widespread use. It is impossible to utilize standard benchmarks like NSL-KDD to evaluate and validate the efficiency of new Network Intrusion Detection due to the heterogeneous nature of IoT systems (NIDS). In a research [7], the author looked at particular threats in the NSL-KDD database that might affect sensor nodes and networking in IoT contexts to fill the gap. In addition, we examined and reported on the findings of eleven machine learning methods in order to identify the newly emerging assaults. Through quantitative simulation, we demonstrate that approaches based on trees and ensembles outperform all other methods of machine learning. XGBoost is the best supervised algorithm with 95 percent accuracy. Also of note in this study is that the unsupervised Modeling approach surpasses the Nave Bayes classifier by 22.0 percent when it comes to detecting assaults in the NSL-KDD dataset which is a noteworthy research discovery. As a network security measure, intrusion detection has proven useful. Many of the current approaches for detecting network anomalies are based on well-established ml algorithms like KNN, SVM, and so on. It's possible to generate impressive characteristics, but these approaches have a poor rate of precision and depend primarily on human traffic features that have become outdated in the age of big data.
It is recommended that a traffic anomaly identification technique BAT be used to address the issues of poor precision and feature engineering in intrusion identification [8]. BLSTM and attention mechanisms are included in the BAT model. As a result of the attention mechanism, the BLSTM model generates a connectivity flow vector that may be used to classify network traffic. The local aspects of traffic information are also captured using numerous convolutional layers. The BAT model is referred to as BAT-MC because it uses multiple convolutional layers to analyze data samples. Using the softmax classifier, traffic on the network may be classified. No feature engineering skills are required for the suggested end-to-end model, which is capable of independently learning the hierarchy's most important characteristics. It is able to accurately represent network traffic patterns and enhance anomaly detection. The experimental findings conclude that the model performs better than existing comparison approaches on a publicly available benchmark dataset. NIDSs, or Network Intrusion Detection Systems, are critical pieces of cyber-defense equipment.
In order to develop a profile of normal and malicious activity, NIDSs use a variety of techniques. Machine-learning-based NIDS were devised and implemented in article [9] by the author, and their performance was evaluated. In particular, we look at six approaches of supervised learning that fall into three categories: (1) ensemble approaches, (2) NN techniques, and (3) kernel approach. NSL KDD and NSL-Kitsune-2018 datasets are used to test the created NIDSs, which are based on a current real-world IoT traffic that has been exposed to various network assaults. The recognition efficiency, error rates, and inference speed are evaluated using standard performance measures from the ml algorithms literature. In comparison to neural network and kernel approaches, our empirical study shows that ensemble learning have greater precision and fewer margins of error. Neural networks, on the other hand, have the fastest inference speed, which demonstrates their applicability for high-bandwidth networks. Our greatest findings outperform any previous art by 120 percent, as shown in a relation to existing state-of-the-art solutions.
Data sharing and administration of networked "things" are all possible with the Internet of things (IoT). IoT devices are growing and serve a critical role in increasing people's quality of life and their level of living. The actual IoT, on the other hand, is more susceptible to the myriad Internet-based assaults that might lead to the leaking of personal information, data manipulation, and other damage to society and people. The Internet of Things (IoT) relies heavily on network security, and online injection, particularly web shell, is among the most serious threats. Using fundamental machine learning algorithms to identify web shell, author [10] built reliable services for IoT networks. These machine learning models will be enhanced by ensemble approaches and voting [11]. The reliability of web shell incursion has been shown by extensive testing on these models. Simulated findings suggest that RF and ET are appropriate for mild IoT settings, whereas the Voting approach is beneficial in heavy IoT scenarios.

IoT-enabled technology, communications, and apps are used in a smart city in order to increase operational efficiency and improve both the level of services delivered by service providers and the quality of life enjoyed by citizens. There is, however, a greater danger of cyberattacks and threats with the expansion of smart city networks. Detectors coupled to huge public cloud uncover IoT systems in a smart city system to fraudulent malicious activities. Such assaults must be prevented and IoT devices must be protected against failure. The author of article [12] investigated a machine learning algorithm-based attack and anomaly detection method to protect and mitigate IoT cybersecurity risks in a smart city. For example, instead of relying just on a single classifier, we also look at ensemble approaches like bagging, boosting, and stacking. For the described area, we also investigate a combination of selecting features, cross-validation, and multi-class classification that has not been widely studied in the current literature [13] [14]. "Stacking ensemble models beat similar models in terms of performance metrics, suggesting stocking's potential in this area validated with experimental findings using the most current attack database [15] [16] [17].

3. METHODOLOGY

Our goal is to identify assaults in an IoT system using ensemble machine learning methods. The reason for this is because deep learning models demand large amounts of memory. For real-time assault identification, the aim is to find the optimal ensemble approach. Thing, fog, and cloud layers are shown in Figure 2. To do this, youmust go through the three stages listed below (as seen in Figure 2): Putting the right model on the clouds, collecting data at the cloud layer first, and then picking the best prediction from an ensemble of models run on the server. Below is a breakdown of the aforementioned responsibilities.

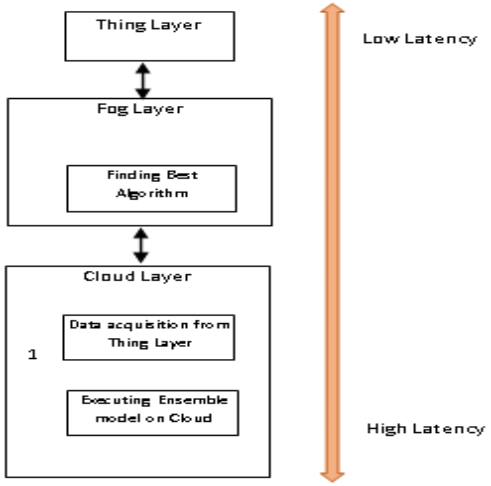

Figure 2. Proposed Approach

3.1. Data Gathering

Gathering information from the things layer and transmitting it to the top layer is part of this process. Input from the thing tier may then be sent to the fog layer in order to do this. The cloud layer may then use the fog layer to convey it. The fog layer may screen information while it is being transferred to the cloud layer, allowing it to select which data should be sent there. The following characteristics may be used to anticipate Internet of Things (IoT) attacks: A user's login information is followed by a list of net datagram fields such as segment characteristics and source and destination addresses (such as IP addresses). Data utilized in our simulation will be given in the next section.

3.2 Choosing Top Model

There are many fundamental machine learning algorithms that may be combined in this stage in order to get the best outcomes. This is a lengthy process; consequently we suggest doing it on the cloud. As a result, we only use the most basic machine learning algorithms since they take less time to run.

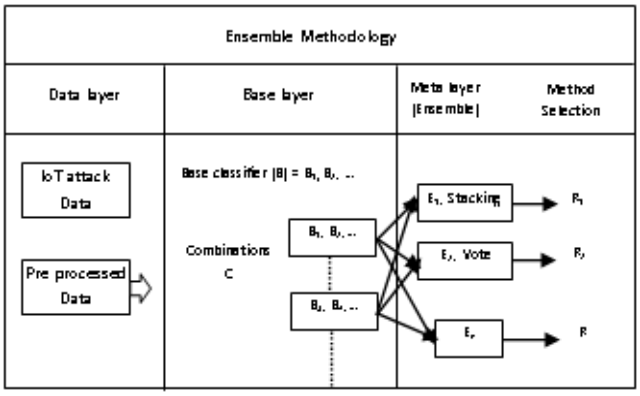

Figure 3. Ensemble Model Selection.

Four layer is composed are shown in Figure 3 to demonstrate this process Pre-processed input from the preceding phase is supplied into the bottom layer in the data layer. Decision trees and KNN are all used in the base layer, as are nave Bayes and decision trees. They are input into the meta layer where ensemble techniques such as stacking and voting are used to combine the outcomes of different pairings. ROC and processing time are taken into consideration while evaluating each ensemble approach. In addition, the model with the best mix of basic classifiers and an ensemble approach is chosen. First, the outcome and results are both set to NULL in Algorithm 1's variables OUTPUT and RESULTS, respectively. The completion time is set to be maximum.

3.3 Executing Top Model on the Fog Layer

Things layer actual statistics is used to run models over the fog layer in this stage based on earlier selections. Base classifiers and an ensemble approach make up the model.A cross-Atlantic experimental analysis of smart Infrastructure IoT networks has been suggested. An IoT gateway transmits data to a cloud-based IoT deployment via the Internet, increasing safety and delay by conducting secure network services. Because fog/edge nodes don't have the capacity to perform heavyweight techniques that take a lot of resources in commercial IoT networks, our technique provides a feasible solution in real time. It is thus reasonable to use only one layer of the fog, the trained model, to reduce the fog node's resource needs. It also makes perfect sense to train the information in the cloud, as outlined in stages 1 and 2 of the process.

4. SIMULATION ENVIRONMENT

4.1 Server Configuration

For the purposes of testing, a 32-bit OS with a Processor speed of 2.80 GHz was used in conjunction with a CoreI511400 processor and 4GB RAM. The cloud node implements the suggested ensemble technique, while the fog node runs the best model. To conduct cloud-based experiments and identify IoT assaults in real time in the fog, researchers turn to the Weka framework.

4.2 Dataset Description

Simulated data is taken from the NSL-KDD database(https://www.unb.ca/cic/datasets/nsl.html). To represent a particular IoT network object, it has a total of 41 characteristics. According to these 41 characteristics, one may categories network invasions into computational data (such as flag or land), content-based data (such as login or root shell data), length or hosts (such as a host's IP address).

The NSL-KDD dataset is shown in Figure 4 by means of two layers: Examples of assaults within each category are shown in the outer layer, which depicts the several forms of IoT attacks in the database. Probe IoT attacks include attacks like Saint, Satan, Nmap, and portsweep, shown in Figure 4. Examine a connected device for holes in its design, which are then manipulated by the hacker for access to secret data.

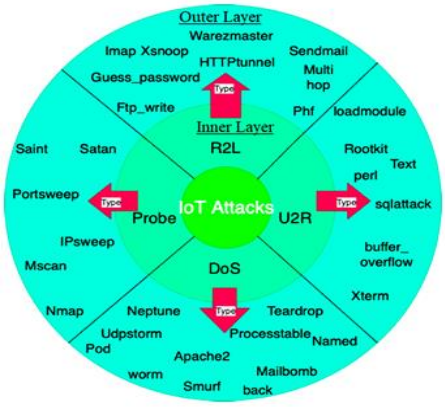

Figure 4. Dataset description.

In the same way, DoS assaults like Neptune, Teardrop, Worm, and Smurf are included in this category. Denial-of-service attacks occur when an attacker uses too many resources, preventing authorized users from accessing a resource, thereby rendering the service unavailable. There are also R2L attacks (remote user to user) and user-to-root attacks (user root to user) that are distinct from one other. Components are highlighted in Figure 4 based on their section. Nominal values predominate in this dataset. TCP, UDP, and FTP are the three main protocols in the collection.

4.3 Data Segregation

While the cloud layer stores previous data on network connections related with IoT threats, the fog layer analyses actual information in order to prevent future assaults. In addition, the cloud layer contains the predicted value and its associated tags, but the fog layer needs this parameter to be forecasted for new additions or tags. There are two sets of data in the NSL-KDD dataset: one for training and the other for testing. Training information is intended as cloud information, while testing information is used as fog data for experimental purposes. To make things more interesting, the cloud layer uses a major portion of the NSL-KDD dataset for train and test, while the fog layer uses the remaining dataset for real-time assessment. At the cloud layer, 80/20 K-cross validation is employed.

4.4 Blending Classifiers

Multiple ML techniques along with couple of ensemble approaches were used to simulate the suggested strategy. Decision trees (DT), random forests (RF), KNNs (KNNs), logistic regression (LRs), and nave Bayes (NBs) are among the classifiers used, along with voting and stacking approaches for ensemble analysis. Table 2 illustrates the specifics of each base classifier pairing used in the base layer. It is tested with ten distinct models. Table 2 lists the variants. It's because we chose five basic predictors, and then constructed two-classifier combinations. As a result, we have ten options.

Model number	**Blending Base Classifier**		
1	decision tree	random forest	K-nearest neighbor
2	random forest	K-nearest neighbor	logistic regression
3	K-nearest neighbor	logistic regression	naïve Bayes

4	logistic regression	naïve Bayes	decision tree
5	naïve Bayes	decision tree	random forest
6	decision tree	K-nearest neighbor	logistic regression
7	random forest	logistic regression	naïve Bayes
8	K-nearest neighbor	naïve Bayes	decision tree
9	logistic regression	decision tree	random forest
10	naïve Bayes	random forest	K-nearest neighbor

Table 2: Blending Base Classifier

5. RESULTS AND DISCUSSIONS

In this section, we assess the outcomes of the suggested technique using 3 aspects: (1) processing time, (2) quality estimate, and (3) variance. More train data is utilized to develop models and run tests in the cloud layer than on a local computer. The fog layer is used to evaluate fresh data. Selecting the most accurate version is done in real-time with data collected in the cloud layer. This is followed by an analysis of data collected from a layer of fog.

5.1 Cloud Layer Analysis

5.1.1 Runtime Analysis

Using the models listed in Table 2, the voting and stacking ensemble procedures are shown in Figure 5 along with their associated execution times, which shows the number of seconds needed to run each individual model, as well as their length in seconds. Stacking takes substantially longer to execute than the voting ensemble approach. Model number 8 with voting approach has the shortest execution time (9.18 s) and uses KNN, NB, and DT as its basis classifications, as per our findings.

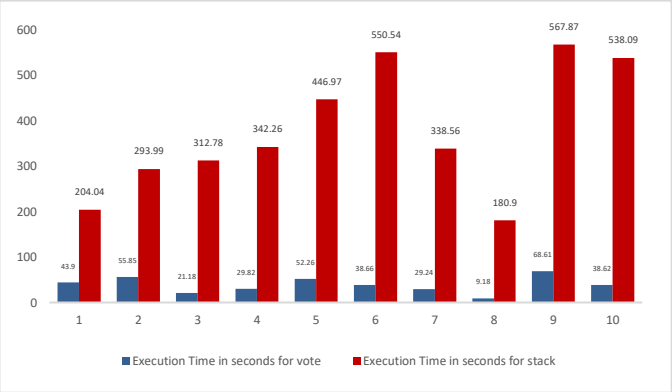

Figure 5: Runtime of various models.

5.1.2 Evaluation Metrics

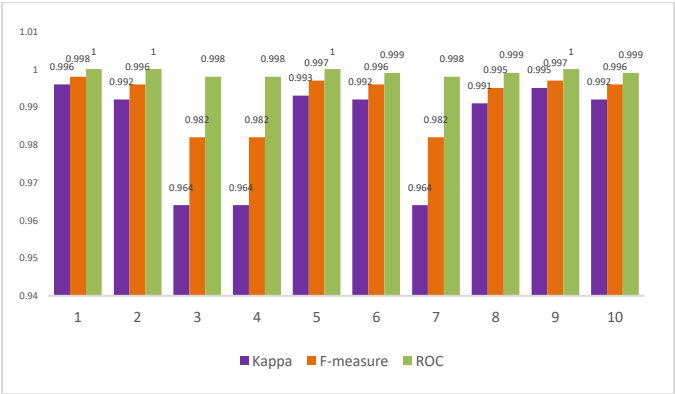

Figure 6: Performance metrics of different models.

Figure 6 depicts performance metrics of different models. As can be seen, all the models had kappa values more than or equal to 0.99, with model 8 having the highest value at 0.991 with an F-measure of 0.995 and an area under the receiver operating characteristic curve (0.99). As can be seen in Figure 7, the voting ensemble approach has errors in terms of MAE, RMSE, RAE, and RRSE. In comparison to any other model, Model 1 with voting has a much lower error rate.Model 8 is chosen for the fog layer despite the fact that it worked well in respect of running time and other performance indicators, as shown in Figure 6.

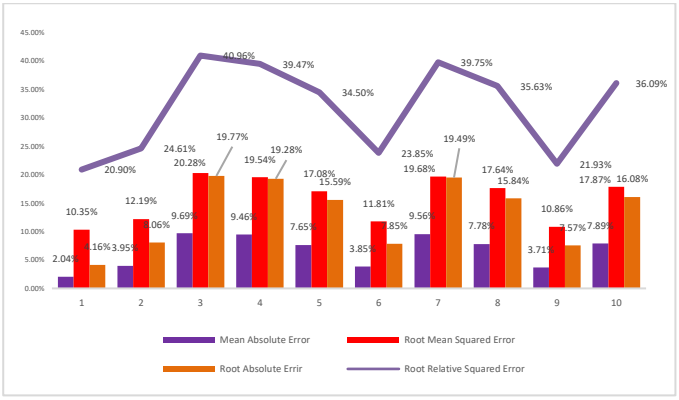

Figure 7. Errors associated with all the models.

Additional measurements of model 8's effectiveness were performed (see Figure 8) to make sure it meets our standards for accuracy and robustness. The Y-axis results are reliable to three decimal places. As a rule, the Model 8's performance in this trial was closer to 99.99%.

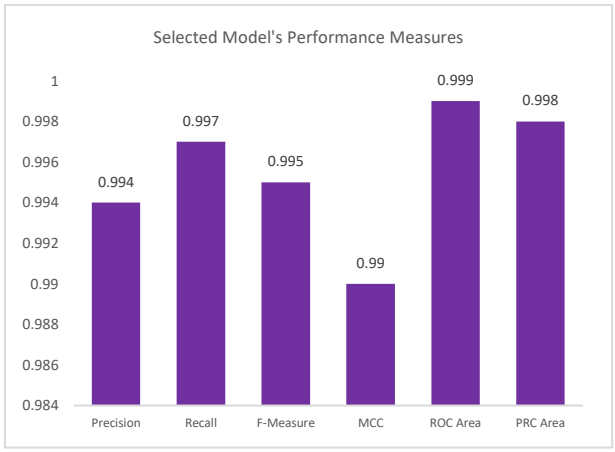

Figure 8: Performance Metrics of selected Models.

We discovered that model 8 based on KNN, DT, and NB outperformed all other models in terms of processing time and Evaluation metrics. Using the voting ensemble approach, it is determined that model 8 takes the least amount of time: 1.18 seconds. In addition, the highest values of kappa, F-measure, ROC, and MCC are 6.39, 98.20, 99.60, and 96.40. It has a mean absolute error of 7.87 percent, RMSE of 17.72 percent, a RAE of 15.91 percent, and a RRSE of 35.68 percent. Model 8 has a RRSE inaccuracy of 27.94 percent, and its minimal effect is 0.6 percent. However, the most time and resource costly approach, model 8, has the largest influence.

5.2 Fog Layer Result Analysis

As a result of the newly added data, we can now analyze the effectiveness of model 8, which combines KNN, Naïve Bayes, and Decision Tree as the model's primary models with voting to create an ensemble model.

5.2.1 Evaluation Metrics

Fog layer evaluation metrics reveals how nicely our model is performing in the fog layer. Figure 9 shows that under the chosen model, every evaluation method is almost equal and at or near the top. When looking at the ROC and MCC curves in conjunction with the F-measure, we see that the averages are 99.9 and 96.40 respectively.

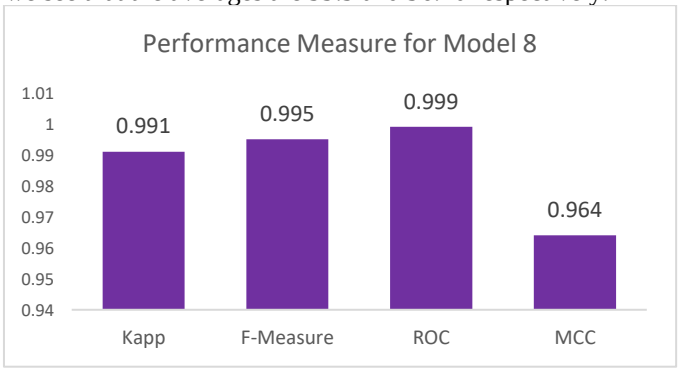

Figure 9: Various Evaluation Metrics

5.2.2 Correlated Errors

Figure 10 shows the various errors on the fog node.

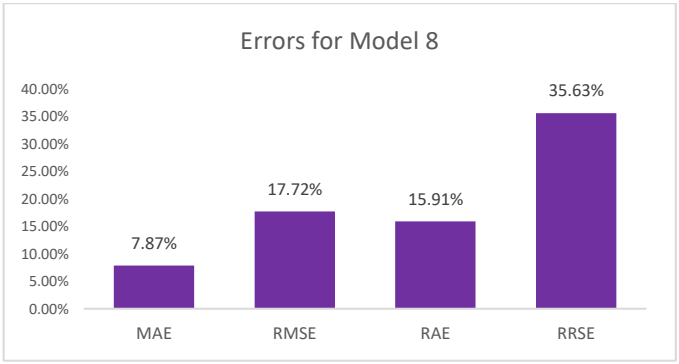

Figure 10: Correlated errors on the fog node.

5.2.3 Runtime and CPU Consumption

Voting as an ensemble approach, we also computed the runtime of our best technique as well as all other technique on the fog node. Figure 11 depicts the total time it took to complete this task. This is a test to see whether the model we choose is efficient in terms of time. Model 8 with voting has the quickest processing time for fog nodes.

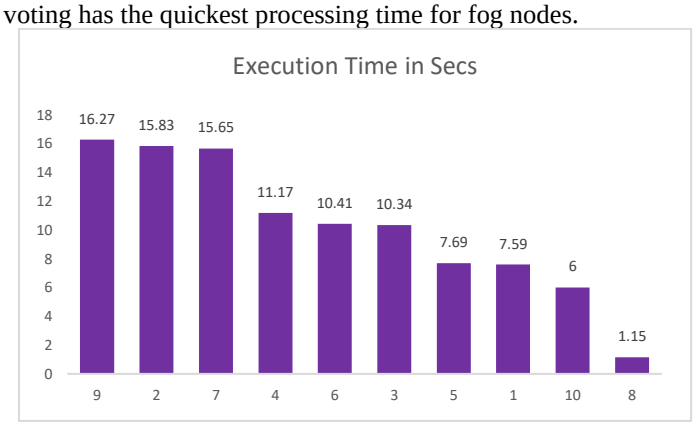

Figure 11: Processing time on the fog node for various models.

Additional calculations were made to determine how much CPU was used in the fog layer. Only a little percentage of the CPU is used by the fog layer. As a result, no extra fog node resources are required for our approach. In addition, our method is fast to implement. This demonstrates the great efficiency of our strategy.

6. CONCLUSION AND FUTURE SCOPE

The purpose of this work is to offer a method that can offload the duty of ensemble ML model choosing to the server, while simultaneously offloading the task of real-world forecasting to fog nodes. By using this method, the server is able to manage high in-depth resource operations, while the fog nodes are able to manage real-time calculations, which simplifies and reduces the amount of effort required for real-time attack detection. On the NSL-KDD dataset, the methodology that has been suggested has been evaluated. The findings that we obtained by using performance metrics, including as kappa, F-measure, ROC, and MCC, revealed that the method that was chosen to represent the cloud layer worked quite well when applied to the fog layer. In addition, the trials showed that the chosen model required a minimum of 1.15 seconds to complete the fog node. According to the findings of the study, stacking takes much more time to implement than the ensemble technique, which includes voting. The NSL-KDD dataset was used in our research. The gathering of data from a genuine testbed simulation is one of our goals for the future. At the moment, both the EU and the US each have access to a number of different testbeds.

7. REFERENCES

[1] Abdulghani, H.A.; Nijdam, N.A.; Collen, A.; Konstantas, D. A Study on Security and Privacy Guidelines, Countermeasures, Threats: IoT Data at Rest Perspective. Symmetry 2019, 11, 774. [CrossRef]

[2] Razdan, S.; Sharma, S. Internet of Medical Things (IoMT): Overview, Emerging Technologies, and Case Studies. IETE Tech. Rev. 2021, 1–14. [CrossRef]

[3] Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network Intrusion Detection for IoT Security Based on Learning Techniques. IEEE Commun. Surv. Tutor. 2019, 21, 2671–2701. [CrossRef]

[4] Jaber, A.N.; Rehman, S.U. FCM–SVM based intrusion detection system for cloud computing environment. Clust. Comput. 2020, 23, 3221–3231.

[5] Hemavathi, D.; Srimathi, H. Effective feature selection technique in an integrated environment using enhanced principal component analysis. J. Ambient. Intell. Humaniz. Comput. 2021, 12, 3679–3688.

[6] Salo, F.; Nassif, A.B.; Essex, A. Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. Comput. Netw. 2019, 148, 164–175.

[7] Liu, J.; Kantarci, B.; Adams, C. Machine learning-driven intrusion detection for contiki-NG-based IoT networks exposed to NSL-KDD dataset. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, Linz, Austria, 13 July 2020; pp. 25–30.

[8] Su, T.; Sun, H.; Zhu, J.; Wang, S.; Li, Y. BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. IEEE Access 2020, 8, 29575–29585.

[9] Abu Al-Haija, Q.; Al-Badawi, A. Attack-Aware IoT Network Traffic Routing Leveraging Ensemble Learning. Sensors 2022, 22, 241. [CrossRef]
[10]Yong, B.; Wei, W.; Li, K.C.; Shen, J.; Zhou, Q.; Wozniak, M.; Połap, D.; Damaševiˇcius, R. Ensemble machine learning approaches for webshell detection in Internet of things environments. In Transactions on Emerging Telecommunications Technologies; Wiley: Hoboken, NJ, USA, 2020; p. e4085. [CrossRef]
[11] Rashid, M.M.; Kamruzzaman, J.; Hassan, M.M.; Imam, T.; Gordon, S. Cyberattacks Detection in IoT-Based Smart City Applications Using Machine Learning Techniques. Int. J. Environ. Res. Public Health 2020, 17, 9347. [CrossRef]
[12] P. Nirmala, et al "An Artificial Intelligence enabled Smart Industrial Automation System based on Internet of Things Assistance," 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), 2022, pp. 1-6, doi: 10.1109/ACCAI53970.2022.9752651
[13] S. Ramesh, et al "Comparison and analysis of Rice Blast disease identification in Greenhouse Controlled Environment and Field Environment using ML Algorithms," 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), 2022, pp. 1-5, doi: 10.1109/ACCAI53970.2022.9752538.
[14] G. Anitha, et al "A Novel Data Communication with Security Enhancement using Threat Management Scheme over Wireless Mobile Networks," 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), 2022, pp. 1-6, doi: 10.1109/ACCAI53970.2022.9752584.
[15] G Ramkumar, Amirthalakshmi T.M, R. Thandaiah Prabu, A. Sabarivani, “An Effectual Plant Leaf Disease Detection using Deep Learning Network with IoT Strategies”, Annals of the Romanian Society for Cell Biology, 2021, Vol.25, Issue.4, Page. 8876 – 8885. 1583-6258.
[16] G. Ramkumar, R. Thandaiah Prabu, NgangbamPhalguni Singh, U. Maheswaran, “Experimental analysis of brain tumor detection system using Machine learning approach”, Materials Today: Proceedings, 2021, ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2021.01.246.
[17] L. Megalan Leo, et al "An IoT Based Automatic Waste Segregation and Monitoring System," 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), 2022, pp. 1262-1267, doi: 10.1109/ICAIS53314.2022.9742926.