Automated Network Intrusion Detection for Internet of Things: Security Enhancements

Aliya Farheen, Dr. YVSS Pragathi

aliyafarheen264@gmail.com, drypragathi@stanley.edu.in

Stanley College of Engineering and Technology for Women

Chapel Road, Abids, Hyderabad, Telangana, India

Abstract: As related devices rapid share personal, sensitive, and vital facts, security attacks are become more complex and frequent. Internet of things (IoT) systems rely critically on green security solutions. Several machine learning strategies—including Random forest, decision Tree, AdaBoost, BernoulliNB, KNN, and Logistic Regression—had been assessed for intrusion detection using the UNSW_NB15 dataset. To improve performance a voting Classifier incorporating Bagging Random forest (BagRF) and Boosted decision Tree (BoostedDT) turned into used. The usage of the filter out approach, the VotingClassifier showed 79.7% of accuracy; the use of the Wrapper approach, 95.4% of accuracy; the usage of the embedded approach, 95.7% of accuracy; the use of the Pearson Correlation Coefficient (p.c) approach, 93.9% of accuracy. Those findings show how well ensemble processes could enhance IoT device network intrusion detection. The suggested technique provides robust and effective detection features, therefore improving IoT security against ever sophisticated cyberthreats.

"Index Terms - *Network Intrusion Detection, Internet of Things (IoT), Machine Learning, Ensemble Methods, Feature Selection, Voting Classifier".*

1. INTRODUCTION

The spread of net use has greatly raised the global switch of touchy and critical personal data over the world huge web. This has caused attackers to take advantage of safety flaws in order to get illegal access to critical facts, therefore endangering data availability, confidentiality, and integrity. Cybersecurity has therefore evolved into a vital field meant to reduce those risks. In this field, the "network Intrusion Detection (NID)" system is absolutely vital. Monitoring computer structures and network traffic, an NID gadget investigates interest to find intrusions. Aside from spotting intrusions, it supervises network activity, detects suspicious or adversarial behaviour, and factors out coverage infractions, therefore facilitating the efficient tracking of modern threats by network managers. [1][3].

"Deep learning (DL) and machine learning (ML)" approaches have come to be effective gadgets for processing big quantities of data and enhancing computational results. While DL techniques automatically extract critical capabilities, permitting enhanced type accuracy, these techniques allow for exceptional function selection from datasets using ML approaches. Current developments in ML and DL have shown amazing energy in identifying intrusions inside "internet of things (IoT)" systems. These structures use algorithms able to recognise complicated styles in network traffic, therefore enhancing the attack detection. Their capacity to change with the times has made them important in contemporary cybersecurity systems since they offer automated and scalable solutions to guard IoT systems [2][4][5].

The growing number of linked devices exchanging touchy statistics in IoT networks has made cybersecurity troubles in them extremely important. Many times deployed in areas with low computational resources, IoT devices are vulnerable to cyberattacks. Strong "Intrusion Detection systems (IDS)" placed at edge nodes assist to lessen those weaknesses. Due to their effectiveness in handling massive IoT networks, IDS solutions using ML and DL approaches are becoming more and more looked for. Those systems are correct in guaranteeing the security of IoT environments seeing that they no longer only identify hazards but additionally trade with the assault pattern. Latest research underline the want of including IDS into IoT systems since they assist to preserve data integrity and availability. [6][7}.

Recent years have visible improvements in AI-enabled IDS that provide exact and fast anomaly detection, hence strengthening network security. Studies on hybrid methods—this is, integrating several algorithms to enhance intrusion detection ability—keep on. The want on smart, automated systems for cybersecurity becomes ever extra evident as IoT use increases, therefore stressing the want of building strong NID systems [5].

2. RELATED WORK

to handle the growing cybersecurity issues in those contexts, studies on the developments in "intrusion detection systems (IDS) for internet of things (IoT)" networks has been intensively conducted. Mahmood et al. [8] offered a structure to maximise community security via combining multi-issue authentication with machine learning. Combining sturdy authentication structures with powerful ML algorithms in their method improves intrusion detection capabilities, therefore increasing detection accuracy and network resilience. analyzing IDS methods, deployment tactics, validation methodologies, and IoT environment problems, Khraisat and Alazab [9] produced a vital overview. They underlined the limits of current structures and the need of choosing suitable public datasets to assess IDS performance.

Presenting a thorough survey contrasting IoT and non-IoT-related intrusion detection systems were Abdulkareem et al. [10]. Their work tested the unique difficulties IoT networks experience—such as restrained resources and heterogeneity—as well as suggested hybrid techniques to properly handle these problems. In keeping with this, Maseer et al. [11] presented DeepIoT. IDS, a hybrid deep learning version supposed to enhance IoT network safety. To boom the detection of intricate infiltration styles, their approach integrates "long short-term memory (LSTM) networks with convolutional neural networks (CNN)", therefore attaining great accuracy and robustness in IoT systems.

Emphasising the incorporation of advanced feature selection techniques to increase detection performance, Qaddos et al. [12] proposed a new intrusion detection framework for optimising IoT security. Even in IoT gadgets with restrained assets, their approach became rather powerful at spotting risks. With voice command automation, Netrant et al. [13] designed and evaluated an IoT-driven smart home security gadget. Their solution gives a creative way to guard IoT devices in houses by including IDS to monitor and secure clever home networks.

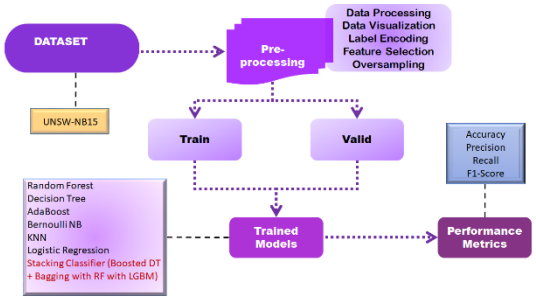
Albalwy [14] and Almohaimeed concentrated on feature selection methods to improve IoT network protection via intrusion detection. They underlined the want of choosing pertinent features to lower computing complexity and hold excessive detection accuracy via simplicity. Leveraging "software-

defined networking (SDN)", Violettas et al. [15] evolved a softwarized IDS for RPL-primarily based IoT networks to improve the scalability and flexibility of intrusion detection in IoT systems. Their era dynamically updates detection rules to change with growing hazards.

Those studies taken together highlight the want of such as superior machine learning, feature selection, and hybrid processes to build robust and efficient IDS for IoT networks, so solving the special difficulties presented by heterogeneous IoT settings and aid-constrained systems.

3. MATERIALS AND METHODS

This paper suggests a sophisticated automatic "network Intrusion Detection (NID) system intended for internet of things (IoT)" surroundings. To improve data excellent and relevance, the system will observe several feature selection tactics including filter out, Wrapper, and Embedded techniques (Lasso with Random forest and Recursive characteristic elimination), [9][14]. Balanced dataset construction using the "synthetic Minority Over-sampling technique (SOMTE)" will help to resolve class imbalance. To raise intrusion detection accuracy [8] [12], the system will apply "Random forest, decision Tree, AdaBoost, and k-Nearest Neighbours" among other machine learning techniques. Extra strong detection will also come from a stacking classifier incorporating Boosted decision tree, bagging with Random forest, and LightGBM [5] [6]. IoT network security is supposed to be strengthened by this integrated strategy.



"Fig.1 Proposed Architecture"

the picture (Fig. 1) targeting security, this network intrusion detection gadget for IoT devices beginning with the US-NB15 dataset, it goes through data preprocessing including visualization, label encoding, feature selection, and oversampling to handle class balance. Training and validation units are then segregated out from the preprocessed cloth. On the facts, Random forest, decision Tree, AdaBoost, Bernoulli NB, KNN, Logistic Regression, and a stacking classifier amongst other "machine learning models are trained and tested. Using accuracy, precision, recall, and F1-score, the system ranks" version performance to identify the best a success intrusion detection method.

i) Dataset Collection:

With 5 instances—each reflecting a network activity—the UNSW_NB15 training set is the one used on this experiment. There are 40-5 features in the dataset, each reflecting various facets of network behaviour and properties. these characteristics contain data about the traffic, protocols, and network moves. The thorough collection of statistics points helps the model to distinguish between benign and malicious behaviour, therefore giving a sturdy basis for the creation and testing of the intrusion detection system.

"Fig.2 Dataset Collection Table"

ii) Pre-Processing:

Getting the dataset ready for model development depends severely on data pretreatment. Removing duplicates, cleaning, normalising labels, encoding labels, choosing pertinent functions, utilising oversampling techniques to balance data, and separating the dataset into training and validation subsets constitute part of it.

a) Data Processing: data processing entails resolving missing values, eliminating duplicate entries, and normalising numerical quantities to guarantee homogeneity. Those procedures help the data to be greater consistent and of first-class, so ready for extra study. Standardising the data allows the model to operate more effectively and efficiently, hence improving the intrusion detection outcomes.

b) Data Visualization: understanding the distribution of the dataset and the interactions among variables requires first knowledge of statistics visualisation. To find trends, styles, and anomalies in the data, one creates graphical representations such histograms, field graphs, and scatter plots. Visualisation helps to choose significant features and provide understanding of network behaviour, therefore improving the model for higher accuracy.

c) Label Encoding: Numerical values are derived from class string labels via label encoding. Assigning a distinct integer to every class helps the machine learning model to extra effectively handle the data. When working with non-numeric data, this degree is absolutely essential since most algorithms demand numerical input. Label encoding guarantees the seamless switch of textual elements into a layout fit for training the intrusion detection system.

d) Feature Selection: feature selection seeks to minimise in the dataset the quantity of superfluous or redundant characteristics. The filter method evaluates each feature's particular applicability. Using predictive models, the Wrapper method ranks feature sets. Using Lasso and Random forest with "Recursive feature elimination (RFE)", the embedded method aggregates feature selection with model training. by using choosing the most critical attributes and hence enhancing computational efficiency, these methods enhance model performance.

e) Oversampling of Data: class imbalance in the dataset is addressed via overampling. The "synthetic Minority Over-sampling technique (SMOTE)" creates synthetic samples for the under-represented class, therefore guaranteeing a more balanced dataset. increasing the range of minority class instances enables the version detect intrusions, therefore ensuring it is not biassed towards the majority class and producing extra accurate forecasts.

iii) Training & Testing:

Training and validation units separate the facts so that the performance of the model can be assessed. The model is taught from the training set; the validation set evaluates if it could extend to unprocessed facts. After machine learning algorithms train the model, testing follows to ascertain its accuracy, precision, recall, and other measures. This method clarifies the model for best intrusion detection.

iv) Algorithms:

Random Forest is a method of ensemble learning whereby several decision trees are created and their outputs are merged to increase forecast accuracy and lower overfitting. by use of feature pattern analysis, it efficiently kinds network data and detects intrusions, thereby guaranteeing strong attack detection in IoT settings. [1][5]

Decision Tree is a supervised learning method splitting data depending on feature values using a tree-like framework. In IoT systems, it helps clear, understandable classifications of network data, therefore enabling detection of anomalies and intrusions depending on unique criteria [9][10].

AdaBoost, Adaptive Boosting, brief for weak classifier combination, is an ensemble method producing a robust model. by iteratively improving classifiers, raising sensitivity to diffused assault styles, and increasing general detection accuracy in IoT networks, it increases intrusion detection [6][8].

Bernoulli Naive Bayes suitable for binary data since it is a probabilistic classifier grounded on Bayes' theory. by use of feature life or absence, it distinguishes network events as benign or malicious, therefore offering effective and real-time security threat detection in IoT contexts [11][12].

K-Nearest Neighbors (KNN) is a non-parametric method assigning labels depending on the majority class of the nearest neighbours. it is useful for IoT security threat detection since it finds abnormalities in network traffic by matching fresh data points to historical trends [9][13].

Logistic Regression is a statistical approach based on predictor variables for binary type, hence guiding results. It is a short selection for intrusion detection in IoT systems since it approximates the possibility of a network event being an intrusion, therefore guiding protection warnings [12][14].

Combining several classifiers, the stacking classifier increases prediction performance. Combining the strengths of models such as Boosted decision trees, Bagging with Random forest, and LightGBM improves intrusion detection accuracy and dependability, thereby offering entire understanding of IoT network security [7][8].

4. RESULTS & DISCUSSION

Accuracy: The ability to properly separate patients from healthy cases defines their accuracy. Calculation of the percentage of actual positive and actual bad bad in all cases analyzed helps to project the accuracy of the test. This is mathematics.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} (1)$$

Precision: Accuracy measurements of positively classified events or samples, which are the percentage of positively classified events. The formula for determining the accuracy is:

$$Precision = \frac{\text{True Positive}}{\text{True Positive } + \text{False Positive}} (2)$$

Recall: In machine learning, callbacks are statistical measures of the model's ability to localize all relevant criteria for a particular class. It provides facts regarding the completeness of the version of the accurately predicted positive observation of actual positive positives.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} \ + \ \mathrm{FN}} (3)$$

F1-Score: In machine learning, F1 score is a metric of version correctness. It blends a version's recall and precision ratings. across the complete dataset, the accuracy measure counts the quantity of times a model produced a right prediction.

$$F1\ Score = 2 * \frac{Recall\ \mathrm{X}\ Precision}{Recall + Precision} * 100(1)$$

Table (1) assess for each algorithm the performance measures: "accuracy, precision, recall, and F1-score. The voting Classifier (BoosetdDT + BaggingRF)" routinely beats all different algorithms across all measures. Furthermore providing a comparative study of the metrics for the various algorithms are the tables.

Table (2) assess for every algorithm the performance measures: accuracy, precision, recall, and F1-rating. Boosted DT+ BagRF routinely beats all other algorithms across all measures. Furthermore providing a comparative study of the metrics for the various algorithms are the tables.

Table (3) assess for every algorithm the performance measures: "accuracy, precision, recall, and F1-score. The voting Classifier (BoosetdDT + BaggingRF)" routinely beats all other algorithms across all measures. Furthermore providing a comparison of the metrics for the various methods are the tables.

Table (4) assess for each algorithm the performance indicators—"accuracy, precision, recall, and F1-score. The voting Classifier (BoosetdDT + BaggingRF)" routinely beats all other algorithms across all measures. Furthermore providing a comparative examination of the metrics for the various algorithms are the tables.

"Table.1 Performance Evaluation Metrics - Filter Method"

ML Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.752	0.758	0.752	0.751
Decision Tree	0.601	0.715	0.601	0.619
AdaBoost	0.529	0.656	0.529	0.567
BernoulliNB	0.170	0.994	0.170	0.283
KNN	0.652	0.754	0.652	0.660
Logistic Regression	0.594	0.637	0.594	0.598
VotingClassifier (BoostedDT+ BagRF)	**0.797**	**0.804**	**0.797**	**0.797**

"Table.2 Performance Evaluation Metrics - Wrapper Method"

ML Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.871	0.871	0.871	0.869
Decision Tree	0.604	0.773	0.604	0.648
AdaBoost	0.668	0.699	0.668	0.666
BernoulliNB	0.296	0.862	0.296	0.436
KNN	0.799	0.800	0.799	0.797
Logistic Regression	0.174	0.702	0.174	0.269

VotingClassifier (BoostedDT+ BagRF)	**0.954**	**0.955**	**0.954**	**0.953**

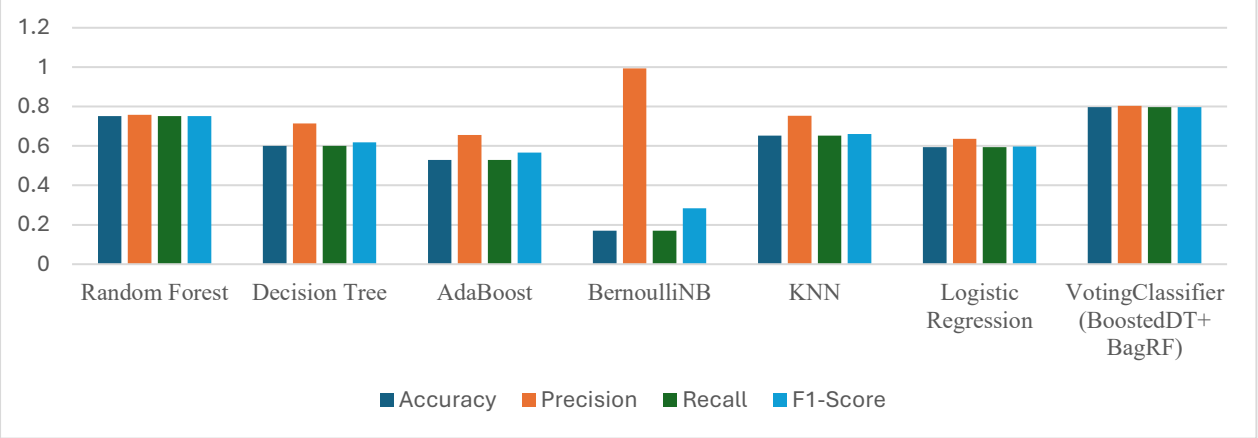
"Table.3 Performance Evaluation Metrics - Embedded Method"

ML Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.836	0.835	0.836	0.835
Decision Tree	0.578	0.703	0.578	0.603
AdaBoost	0.647	0.665	0.647	0.652
BernoulliNB	0.298	0.888	0.298	0.444
KNN	0.724	0.725	0.724	0.721
Logistic Regression	0.500	0.592	0.500	0.516
VotingClassifier (BoostedDT+ BagRF)	**0.957**	**0.957**	**0.957**	**0.956**

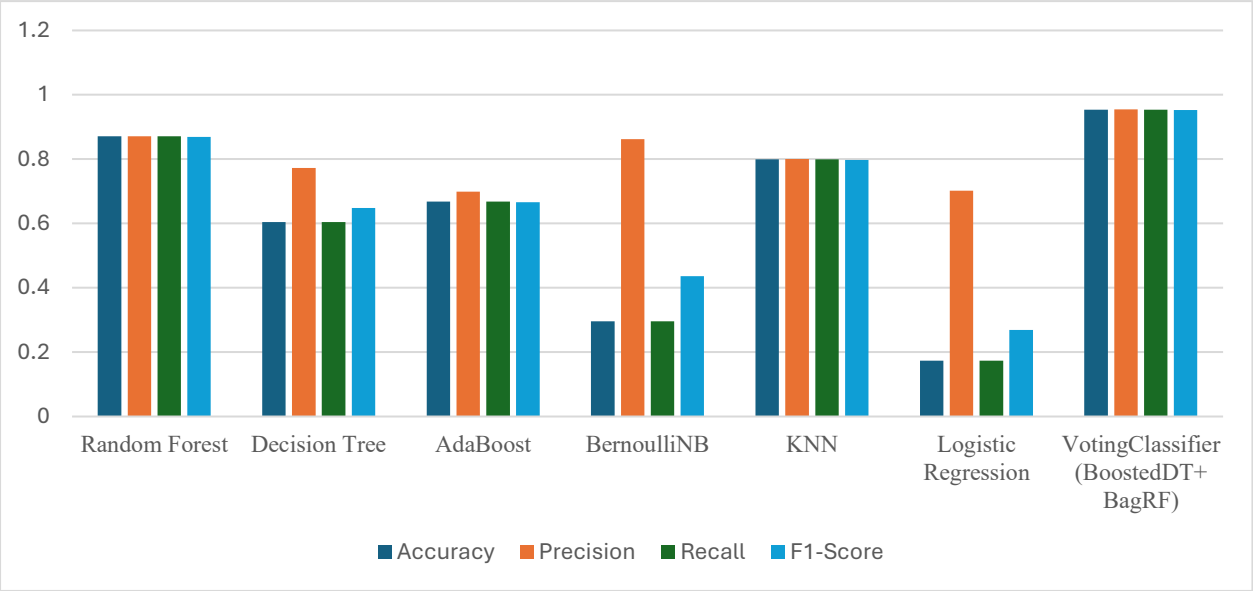
"Table.4 Performance Evaluation Metrics – PCC"

ML Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.871	0.874	0.871	0.870
Decision Tree	0.601	0.740	0.601	0.621
AdaBoost	0.680	0.705	0.680	0.686
BernoulliNB	0.282	0.818	0.282	0.417
KNN	0.843	0.844	0.843	0.841
Logistic Regression	0.633	0.697	0.633	0.653
VotingClassifier (BoostedDT+ BagRF)	**0.939**	**0.941**	**0.939**	**0.939**

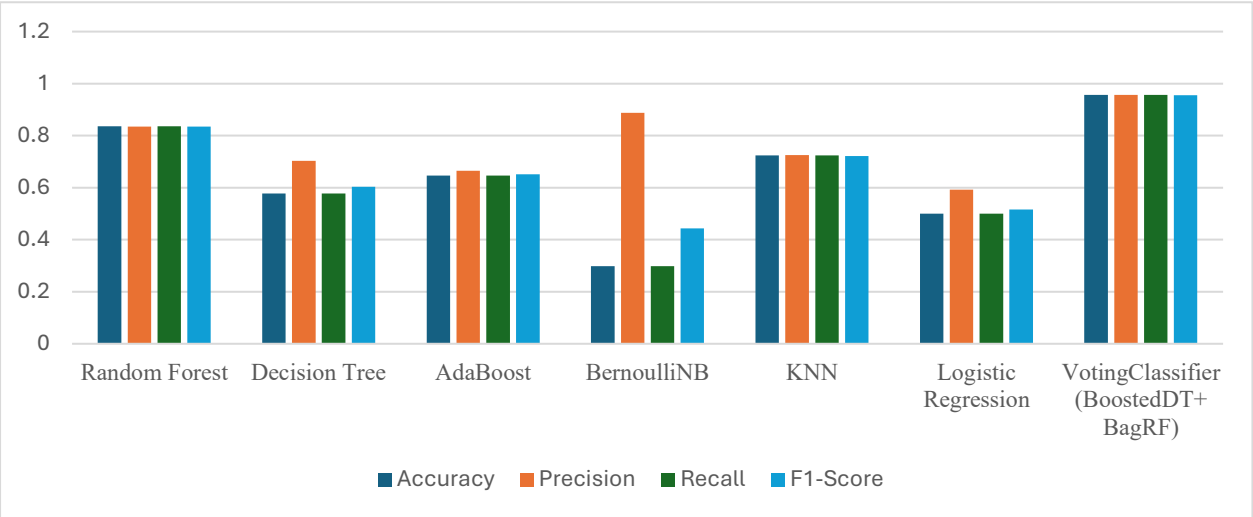
"Graph.1 Comparison Graphs - Filter Method"



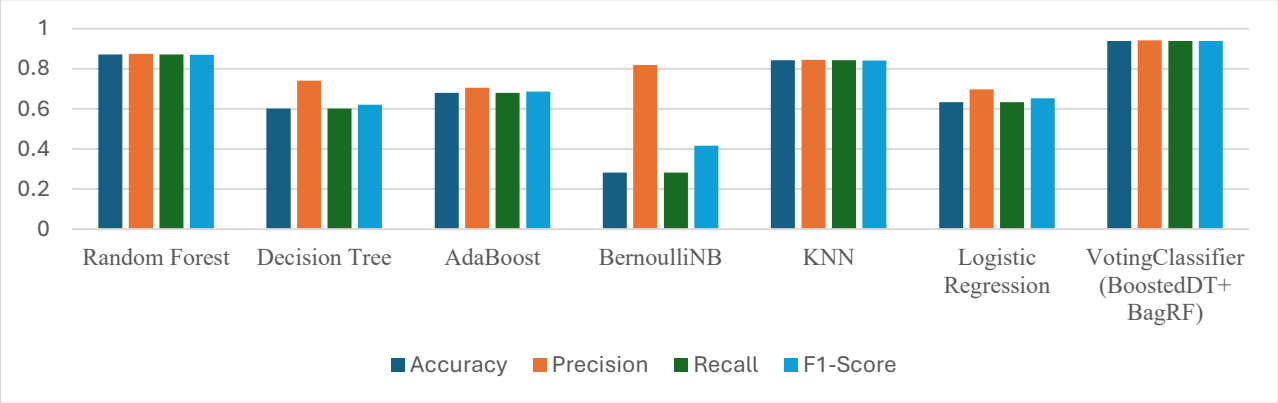
"Graph.2 Comparison Graphs - Wrapper Method"



"Graph.3 Comparison Graphs - Embedded Method"



"Graph.4 Comparison Graphs – PCC"

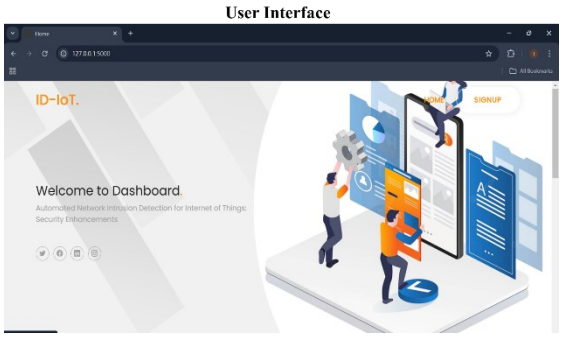


Blue denotes accuracy; orange, precision; green, recall; and sky - blue, Graph (1) F1-score. The voting Classifier (BoosetdDT + BaggingRF) achieves the highest values and shows better performance over all measures than the other models. These conclusions are vividly shown by the graphs above.

Blue denotes accuracy; orange, precision; green, recall; and sky - blue, Graph (2) F1-score. The voting Classifier (BoosetdDT + BaggingRF) achieves the highest values and shows better performance over all measures than the other models. Those conclusions are vividly shown by the graphs above.

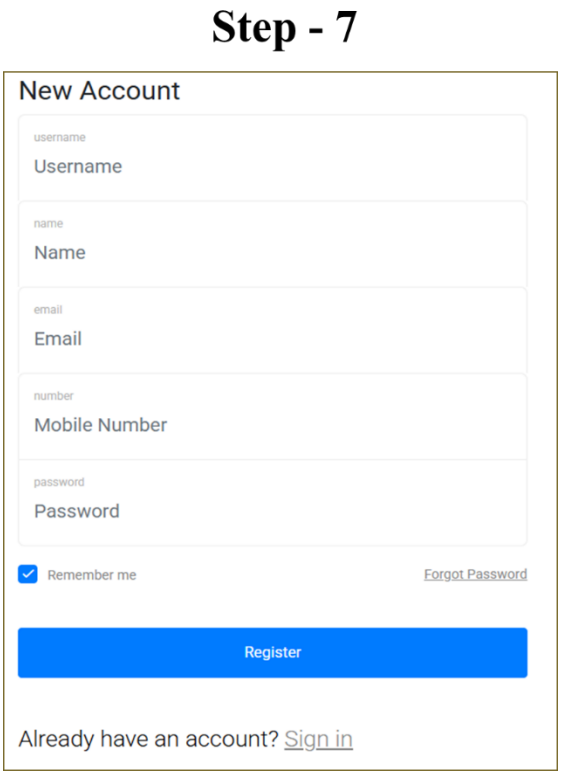
Blue denotes accuracy; orange, precision; green, recall; and sky - blue, Graph (3) F1-score. The voting Classifier (BoosetdDT + BaggingRF) achieves the highest values and shows better performance over all measures than the other models. those conclusions are vividly shown by the graphs above.

Blue denotes accuracy; orange, precision; green, recall; and sky - blue, Graph (4) F1-score. The voting Classifier (BoosetdDT + BaggingRF) achieves the highest values and shows better performance over all measures than the other models. those conclusions are graphically shown above.



"Fig. 3 Dash Board"

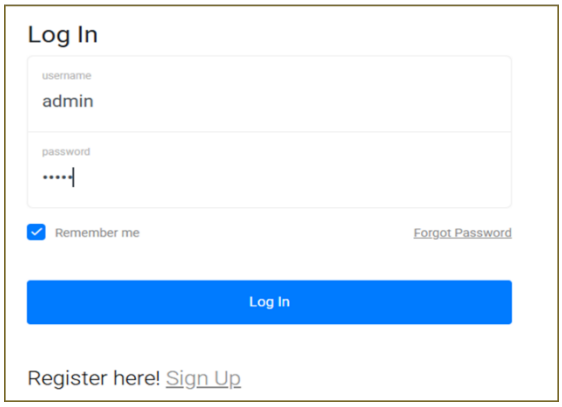
Designed presumably with "internet of things (IoT)" security in mind, figure 3 displays the user interface of a dashboard for an intrusion detection system. It has an inviting message and a picture of people working on a network.



"Fig. 4 Register page"

Figure 4 displays a user registration shape. One needs a username, call, e mail, mobile number, and password. For current users, it also has a link to "sign in" and a "register" button.

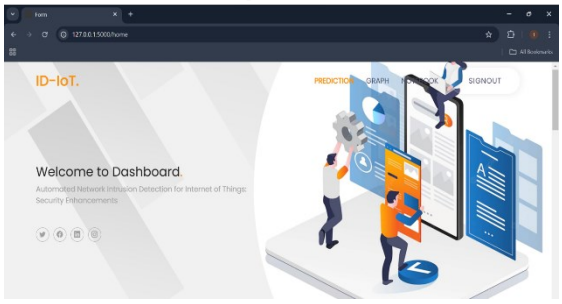
Step - 8



"Fig. 5 Login page"

The login page seen in Fig. 5 bears the words "Log In." "Admin" fills the username field already. It boasts a "Log In" button and a password area as well. Additionally available are links for a "Forgot Password" and a "remember me". Users can register for a fresh account as well.

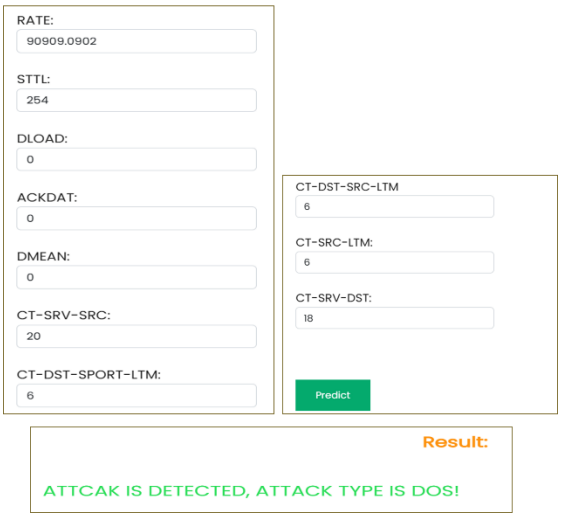
Step - 9



"Fig. 6 Main page"

Figure 6 presents the dashboard of a web application. The name is "Welcome to Dashboard" and the slogan revolves on automatic internet of factors network intrusion detection.

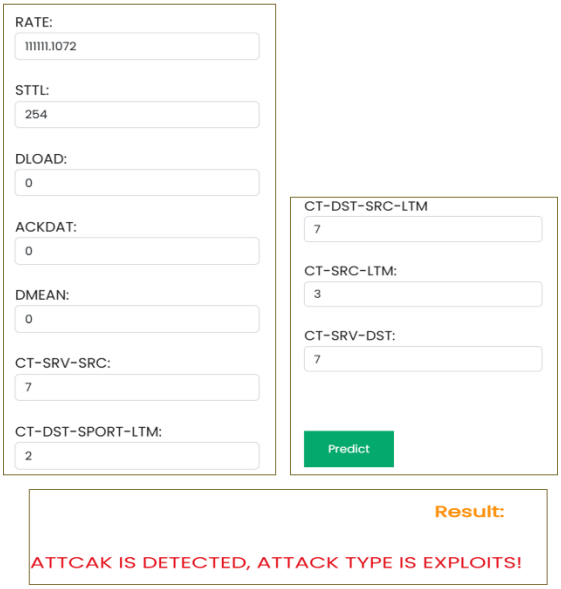
Step – 9
Test case 1



"Fig. 7 Test case – 1"

Figure 7 presents a shape for identifying network assaults. It gathers facts including rate, STTL, DLOAD, ACKDAT, and several connection-related aspects. Following data entry, the shape forecasts an outcome as a "DOS" attack.

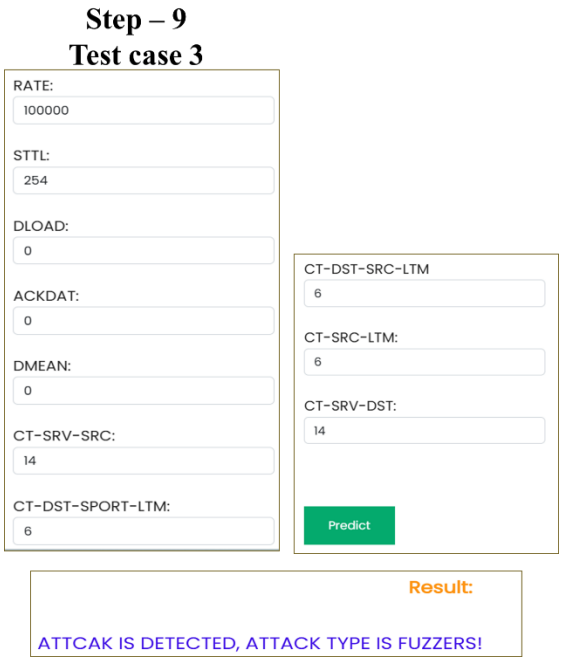
Step – 9
Test case 2



"Fig. 8 Test case – 2"

Figure 8 presents a shape for identifying network attacks. It gathers facts including rate, STTL,

DLOAD, ACKDAT, and several connection-related aspects. Following data entry, the form forecasts a "EXPLOITS" attack as the conclusion.



"Fig.9 Test case – 3"

Figure 9 presents a form for identifying network assaults. It gathers information including rate, STTL, DLOAD, ACKDAT, and several connection-related aspects. Following data entry, the form projects the end result as a "FUZZERS" attack.

"Fig. 10 Test case – 4"

Figure 10 presents a shape for identifying network assaults. It gathers information including price, STTL, DLOAD, ACKDAT, and several connection-related aspects. Following facts access, the form forecasts the result as a "familiar" assault.

"Fig. 11 Test case – 5"

The shape for spotting network threats is shown in Fig. 11 It gathers information including rate, STTL, DLOAD, ACKDAT, and several connection-related aspects. The form forecasts, upon facts entering, "NO attack IS DETECTED, it is normal!"

5. CONCLUSION

Finally, the automated network Intrusion Detection (NID) system for internet of things (IoT) systems effectively solves the important issues related to security concerns and data integrity demand. Using machine learning methods and the UNSW_NB15 training set, the system showed strong ability to identify network breaches. Combining Boosted decision Tree (BoostedDT) with Bagging Random forest (BagRF), the VotingClassifier stood out among the assessed methods as most a hit. With 95.7% using the embedded approach, 95.4% using the Wrapper method, and 93.9% using the Pearson

Correlation Coefficient (%) approach, it received notable detection accuracy throughout many feature selection techniques. These findings spotlight how well ensemble models find problematic incursion patterns in internet of things structures. By means of effective and specific intrusion detection, thereby making sure secure communication and data exchange amongst changing cyber threats, the counseled approach improves the dependability and resilience of IoT environments.

Extending the automated "network Intrusion Detection (NID)" system to manage real-time intrusion detection in large-scale IoT systems will constitute a part of the future scope of this effort. Deep learning fashions and progressed ensemble techniques ought to assist to elevate detection accuracy and flexibility to match changing attack patterns even further. Moreover, including side computing into the system would improve response time, therefore optimising resource-limited IoT devices and guaranteeing strong network security.

REFERENCES

[1] Khan, A. R., Kashif, M., Jhaveri, R. H., Raut, R., Saba, T., & Bahaj, S. A. (2022). Deep learning for intrusion detection and security of Internet of things (IoT): current analysis, challenges, and possible solutions. Security and Communication Networks, 2022(1), 4016073.

[2] Saheed, Y. K., Abiodun, A. I., Misra, S., Holone, M. K., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. Alexandria Engineering Journal, 61(12), 9395-9409.

[3] Mishra, N., & Pandya, S. (2021). Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review. IEEE Access, 9, 59353-59377.

[4] Karthikeyan, M., Manimegalai, D., & RajaGopal, K. (2024). Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection. Scientific Reports, 14(1), 231.

[5] Alkadi, S., Al-Ahmadi, S., & Ben Ismail, M. M. (2023). Toward improved machine learning-based intrusion detection for Internet of Things traffic. Computers, 12(8), 148.

[6] Shahin, M., Maghanaki, M., Hosseinzadeh, A., & Chen, F. F. (2024). Advancing network security in industrial IoT: a deep dive into AI-enabled intrusion detection systems. Advanced Engineering Informatics, 62, 102685.

[7] Ge, M., Syed, N. F., Fu, X., Baig, Z., & Robles-Kelly, A. (2021). Towards a deep learning-driven intrusion detection approach for Internet of Things. Computer Networks, 186, 107784.

[8] Mahmood, R. K., Mahameed, A. I., Lateef, N. Q., Jasim, H. M., Radhi, A. D., Ahmed, S. R., & Tupe-Waghmare, P. (2024). Optimizing network security with machine learning and multi-factor authentication for enhanced intrusion detection. Journal of Robotics and Control (JRC), 5(5), 1502-1524.

[9] Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4, 1-27.

[10] Abdulkareem, S. A., Foh, C. H., Shojafar, M., Carrez, F., & Moessner, K. (2024). Network Intrusion Detection: An IoT and Non IoT-Related Survey. IEEE Access.

[11] Maseer, Z. K., Yusof, R., Mostafa, S. A., Bahaman, N., Musa, O., & Al-Rimy, B. A. S. (2021). DeepIoT. IDS: Hybrid deep learning for enhancing IoT network intrusion detection. Computers, Materials and Continua, 69(3), 3946-3967.

[12] Qaddos, A., Yaseen, M. U., Al-Shamayleh, A. S., Imran, M., Akhunzada, A., & Alharthi, S. Z. (2024). A novel intrusion detection framework for optimizing IoT security. Scientific Reports, 14(1), 21789.

[13 Netinant, P., Utsanok, T., Rukhiran, M., & Klongdee, S. (2024). Development and Assessment of Internet of Things-Driven Smart Home Security and Automation with Voice Commands. IoT, 5(1), 79-99.

[14] Almohaimeed, M., & Albalwy, F. (2024). Enhancing IoT Network Security Using Feature Selection for Intrusion Detection Systems. Applied Sciences (2076-3417), 14(24).

[15] Violettas, G., Simoglou, G., Petridou, S., & Mamatas, L. (2021). A softwarized intrusion detection system for the RPL-based Internet of Things networks. Future Generation Computer Systems, 125, 698-714.