iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82932

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Intrusion Detection in IoT and IIoT: Comparing Lightweight Machine Learning Techniques Using TON_IoT, WUSTL-IIOT 2021, and EdgeIIoTset Datasets

Jaswanth Syam Sundar Garugu

Abstract: ***There is increasing need for lightweight, scalable and intelligent intrusion detection systems with the rapid growth of IoT and IIoT. In this paper, author present a comprehensive machine learning and deep learning based intrusion detection system and evaluated it on ToN_IoT, WUSTL-IIoT 2021 and Edge-IIoTset datasets. Data preparation includes outlier removal, chi-squared feature selection, and SMOTE/Undersampling imbalance treatment. Algorithms include Convolutional Neural Networks, Deep Neural Networks, Decision Tree, Random Forest, LightGBM, Bagging, a stacking ensemble based on Random Forest and LightGBM and a voting ensemble made of Boosted Decision Tree, Bagging Random Forest and XGBoost. On ToN_IoT, CNN got 93.8% accuracy, DNN got 94.4%, Random Forest got 96.3%, Decision Tree and LightGBM got 96.6%, and the voting ensemble achieved the best accuracy of 98.4%, 98.9% on WUSTL-IIoT 2021, and 96.7% on Edge-IIoTset. Explainable AI techniques, like SHAP and LIME, provide straightforward feature-level explanations. The framework is built with Flask, offers user signin/signout with sqlite, has real-time data input, display data after preprocessing, display the prediction result, and provides actionable data when detecting intrusions in IoT and IIoT security. This implementation allows for secure interaction, visualization and efficient monitoring in various remote industrial network environments around the world.***

"Index Terms: Internet of Things, Industrial Internet of Things, security, cyber-attacks, intrusion detection, machine learning, lightweight, data balancing, cross-dataset transfer learning".

I. INTRODUCTION

With the growing proliferation of IoT and IIoT, numerous industries have experienced a transformation in how they're smartly automated, communicated, and informed about information to make decisions. The number of IoT devices connected around the world is expected to exceed 55.7×10^9 devices by 2025 [1]. This expansion has several security issues that need to be addressed. It is reported that exposure to cyber threats and the impact on IoT devices has been significant: IoT devices represented 33% of all affected devices in 2020 [2]. Cyber threats are growing increasingly prevalent for critical infrastructure, such as healthcare facilities, transportation systems, energy grids and smart factories, as Industry 4.0, the merging of IT and OT, is being embraced. According to Cybersecurity Ventures, cybercrime will cost $10.5 trillion a year by 2025, and the attacks on IoT and IIoT will significantly drive those costs. Moreover, the cost of one data breach in industrial environments is estimated to be $4.24 million, on average [3].

The typical architecture of IIoT systems has a multi-layered structure, consisting of the Edge, Middleware, Application, IT/OT Environment and Cloud Services layers, each with its own set of functions and threats. The Edge Layer (data collection) could be physically manipulated and the Middleware Layer (API access) could be hacked for illegal access. Likewise, the Application Layer is exposed to malware threats, the IT/OT Layer to insider threats, and the Cloud Layer to data breaches and persistent threats [4, 5]. Such risks have been shown in high-profile attacks – Mirai botnet (2016) launched a large DDoS attack on IoT devices; Stuxnet worm (2010) attacked SCADA systems; Triton virus (2017) targeted industrial safety systems.

One solution to these problems are ML based IDS. ML models can identify novel and emerging attacks by analyzing patterns and anomalies, thus allowing higher adaptability in complex IoT/IIoT environments [6] compared to signature-based systems, which only recognize known attacks. For this purpose, several datasets have been developed, such as UNSW-NB15, CICIDS, DS2OS, N-BaIoT 2018, Bot-IoT [7], X-IIoTID [9] or LATAM-DDOS IOT [10] datasets. TON_IoT, WUSTL-IIoT-2021, and Edge-IIoTset are particularly useful since they model real-world industrial operations. There are still challenges, however, such as having highly skewed distributions where the numbers of normal cases far exceed those of the anomalous ones, which makes anomaly detection hard [8]. Additionally, resource-limited IIoT devices need to have efficient and energy-saving machine learning models.

This article is a contribution to the analysis of the previously conducted studies on these datasets, supervised vs ensemble ML methods, cross-dataset transfer learning, and real-time monitoring of network traffic to predict intrusions.

A. *Research Contributions*

1) This study presents a detailed comparative evaluation of machine learning and deep learning algorithms on the ToN-IoT, WUSTL-IIoT 2021, and Edge-lloTset datasets for intrusion detection in IoT and lloT networks.
2) A lightweight ensemble-driven intrusion detection framework is proposed to enhance security performance in resource-constrained IoT and lloT environments.
3) Explainable Al methods, including SHAP and LIME, are incorporated to provide transparent and interpretable intrusion detection results.
4) A real-time web-based intrusion monitoring and prediction system is developed using the Flask framework for practical deployment and user interaction.
5) The proposed approach improves intrusion detection effectiveness through the combination of ensemble learning models and data balancing strategies.

II. LITERATURE REVIEW

For this, numerous approaches to intrusion detection in IoT and IIoT have been investigated, using different forms of machine learning and deep learning techniques. In [11] Mesadieu, Torre, and Chennamaneni presented a reinforcement learning-based method to build an IDS in the SCADA environment that can adapt to changing attack scenarios. Their study demonstrated the feasibility of reinforcement learning in industrial infrastructures and highlighted the importance of continuous learning instead of relying on large amounts of pre-labeled data.

Eid, Soudan, Nassif, and Injadat [12] investigated the effect of data preprocessing and balancing on the performance of ML models for IIoT intrusion detection. When applied to imbalanced datasets, their comparative analysis showed that the classifiers' performance greatly depends on the application of pre-treatment pipelines and balancing procedures. In another study, Eid et al. [13] proposed to use an optimized CNN and multi-class SMOTE balancing to enhance the categorization of minority attacks. The combination of synthetic oversampling and the DL models further boosted the accuracy of detection, regardless of the highly imbalanced nature of the IIoT data sets.

Farea and Küçük [14] developed the Cooja simulator to design an IoT network intrusion detection system based on ML. They have investigated lightweight IDS models that can be deployed within the limited resources of an IoT node to enable deployment in real world applications. ALSHEHRI et al. [15] advanced this research area with a deep convolutional neural network with self-attention mechanism for the intrusion detection of IoT. They captured long-term dependencies in network traffic data by incorporating attention processes, and consequently, their model successfully identified more complex and stealthy intrusions than the usual CNN models.

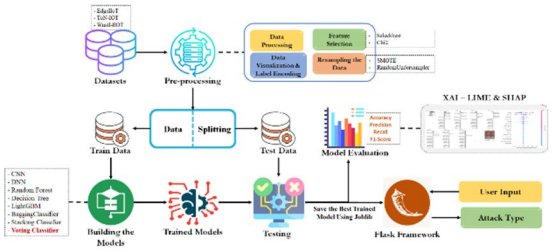
Survey research also has provided useful data. Bansal and Singhrova [16] gave a brief overview of IDS in IoT and IIoT including the existing methodologies, challenges, and advancements. They narrowed their review down to three key points: scalability, explainability, and lightweight IDS. Building on this, Nuaimi, Fourati and Hamed [17] conducted a comprehensive study of intelligent IDS methods for IIoT, which they classified as ML, DL and hybrid. They also uncovered gaps, such as a lack of cross-dataset evaluations, which will inform future research approaches, and a limited consideration of explainability, which will help future research focus on.

IDS development is informed by dataset-based research. Guo et al. [18] evaluated intrusion detection solutions on data provided by the TON_IoT dataset, which features realistic IIoT traffic patterns. Through evaluation, they found that the dataset has the potential to be useful for evaluating the performance of IDSs and that it can be used to compare investigations in a standardized way. In response, Belarbi, Spyridopoulos, Anthi, Mavromatis, Carnelli, and Khan [19] proposed a federated DL system to address the privacy issues in distributed IoT networks. Through their solution, they were able to allow collaborative training without the need to centralize raw data, proving effective for deployment of privacy-preserving and scalable intrusion detection.

Explainability and resilience are important criteria for IDS models. Oseni, Moustafa, Creech, Sohrabi, Strelzoff, Tari and Linkov [20] developed an explainable DL system to detect intrusion in IoT supported transportation systems. They used interpretable AI methods together with DL to make their model transparent in decision making and thus make it easier for the system operators to understand why the forecasts are made. This paper focused on handling the issues of trustworthiness of AI IDS in safety-critical environments, where accountability and resilience are as critical as accuracy.

III. MATERIALS AND METHODS

To address the challenges in the context of IoT and IIoT, the proposed method offers an efficient intrusion detection solution using benchmark datasets like ToN-IoT, WUSTL-IIoT-2021, and Edge-IIoTset. It involves multiple ML and DL algorithms like RF, DT, LightGBM, Bagging Classifier, Stacking Classifier, XGBoost, CNN and DNN, as well as reference models. The framework methodically trains and tests these algorithms to determine their prediction skills in detecting abnormalities and harmful activity. For enhanced usability, a Flask-based front-end is developed to enable real-time interaction with the user and real-time intrusion prediction. It also features an ensemble of [22] Voting Classifier, improved class balancing methods and explainable AI tools such as SHAP and LIME to ensure interpretability and feature-level insights.



"Fig.1 Proposed Architecture"

Figure 1 shows the ML pipeline used for IoT security, which consists of raw data, pre-processing, feature selection (Chi2) and resampling (SMOTE). After data splitting, various models, including a CNN model and a Voting Classifiers model, are trained and tested. Evaluation metrics are used to select the best model using the Joblib; the model is then deployed using Flask for detecting attacks in real-time and interpreting them using XAI.

A. Dataset Collection

1) EdgeIIoT: The EdgeIIoT dataset is used as a standard dataset for detecting intrusions in IoT and IIoT networks. It contains 32,404 of them over 55 different features that store data related to network traffic and protocol-level information such as TCP, UDP, DNS, MQTT, and Modbus connections. Data set includes seven categories of attacks: DDoS (UDP, ICMP, TCP), Port Scanning, Ransomware, Vulnerability Scanner, Password assaults. It has many dangerous traffic patterns and acts as a comprehensive tool for evaluation of ML and DL models for real-time intrusion detection systems.

	frame.time	ip.src_host	ip.dst_host	arp.dst.proto_ipv4	arp.opcode	arp.hw.size	arp.src.proto_ipv4	icmp.checksum	icmp.seq_le
0	1.707159e+09	0	0	0	0	0	0	0	0
1	1.707159e+09	0	0	0	0	0	0	0	0
2	1.707159e+09	0	0	0	0	0	0	0	0
3	1.707159e+09	0	0	0	0	0	0	0	0
4	1.707159e+09	0	0	0	0	0	0	0	0

5 rows × 63 columns

Fig.2 EdgeIIoT Dataset

2) ToN-IoT: ToN-IoT dataset is a large-scale benchmark intrusion detection dataset, which contains 190,474 instances and 20 attributes in the IoT context. It contains different kind of network information like ports, protocols, packet statistics, connection status, DNS parameters and other characteristics of HTTP traffic. There are 10 categories: regular, scanning, DDoS, injection, password, DoS, backdoor, XSS, ransomware, and MITM attacks. ToN-IoT is a valuable source for training and evaluating intrusion detection models that offers a well-balanced representation of a variety of IoT traffic and cyberattacks.

	src_ip	src_port	dst_ip	dst_port	proto	service	duration	src_bytes	dst_bytes	conn_state	...	http_response_body_len	http_status_code
0	192.168.1.37	4444	192.168.1.193	49178	tcp	-	290.371539	101568	2592	OTH	...	0	0
1	192.168.1.193	49180	192.168.1.37	8080	tcp	-	0.000102	0	0	REJ	...	0	0
2	192.168.1.193	49180	192.168.1.37	8080	tcp	-	0.000148	0	0	REJ	...	0	0
3	192.168.1.193	49180	192.168.1.37	8080	tcp	-	0.000113	0	0	REJ	...	0	0
4	192.168.1.193	49180	192.168.1.37	8080	tcp	-	0.000130	0	0	REJ	...	0	0

5 rows × 44 columns

Fig.3 ToN-IoT Dataset

3) WUSTL-IIoT: The WUSTL-IIoT 2021 data set is an extensive benchmark of intrusion detection for IIoT environment which has 1,*194*,464 records and 45 characteristics. It captures network flow attributes like packet counts, byte counts, jitter, loss rates, protocol behavior and application-level data. The data set is split into 5 traffic categories: regular, DoS, reconnaissance, command injection and backdoor attacks. WUSTL-IIoT 2021 covers a wide range of the real industrial IoT network traffic, which will serve as a strong foundation for evaluating ML and DL models in the field of intrusion detection research.

	StartTime	LastTime	SrcAddr	DstAddr	Mean	Sport	Dport	SrcPkts	DstPkts	TotPkts	...	SAppBytes	DAppBytes	TotAppByte	SynAck	RunTime	sTos	SrcJitAct	DstJitAct	Traffic
0	2019-08-19 12:23:28	2019-08-19 12:23:28	192.168.0.20	192.168.0.2	0	59034	502	10	8	18	...	24	20	44	0.001176	0.053037	0	0.000000	0.0	normal
1	2019-08-19 15:13:24	2019-08-19 15:13:24	192.168.0.20	192.168.0.2	0	55841	502	10	8	18	...	24	20	44	0.001308	0.052681	0	0.000000	0.0	normal
2	2019-08-19 13:41:31	2019-08-19 13:41:31	192.168.0.20	192.168.0.2	0	63774	502	10	8	18	...	24	20	44	0.000690	0.051793	0	0.000000	0.0	normal
3	2019-08-19 12:43:19	2019-08-19 12:43:20	209.240.235.92	192.168.0.2	0	61771	80	4	0	4	...	0	0	0	0.000000	0.889555	0	419.338813	0.0	DoS
4	2019-08-19 14:49:44	2019-08-19 14:49:48	192.168.0.20	192.168.0.1	3	0	0	14	0	14	...	476	0	476	0.000000	3.500055	0	525.146562	0.0	normal

5 rows × 49 columns

Fig.4 Wustl-IIOT Dataset

B. Pre-Processing

1) *Data preprocessing:* Data cleansing involves removing outliers, duplicates, and null values, which help ensure that the data is clean, consistent, and reliable. Duplicate rows are also eliminated to prevent duplication and bias, and null values due to poor or incomplete data quality are removed to guarantee data integrity. Outliers are ignored to remove distortion in the model training and thus realistic traffic patterns are obtained. Further, unnecessary columns like IP address and time stamps are eliminated to reduce noise, calculation cost, and dimensionality, which enables model to focus on essential features for precise and rapid intrusion detection.
2) *Data Visualization:* Data visualization helps to understand the feature correlation and traffic patterns in IoT and IIoT data. Correlation matrices exhibit the relationship among the variables, highlighting multicollinearity and duplicated features. Sample results are graphically displayed in a distribution of attack types and normal traffic, to identify imbalances or anomalies. This knowledge is used for feature engineering, selection, and preprocessing methods and to ensure that models are trained on the patterns of data that are relevant and representative to give a good result for intrusion detection.
3) *Label Encoding:* Label Encoding is a method for encoding categorical data that can be input into ML models. Many features are not represented as models, and are often strings, like protocol type, service, and attack category. Each category is encoded with a unique integer, which makes it easy to store and process the data while retaining its meaning. This stage allows the algorithms to successfully apply the categorical information, which leads to the accurate learning and prediction in the intrusion detection system deployed in an IoT and IIoT network.
4) *Feature Selection:* Feature selection is used to select features to reduce dimensionality by retaining the most informative features and discarding redundant or irrelevant features. The strength of the correlations between features and target labels are determined using SelectKBest and chi-squared statistics. This stage helps to enhance the interpretability of the model, shorten training times and boost performance by focusing on key variables. It makes sure intrusion detection models are accurate, efficient, and able to detect complicated attack patterns.
5) *Data Resampling:* Class imbalance is one of the most serious issues in intrusion detection data sets, with some attack types being much under-represented and is solved using data resampling. Oversampling with SMOTE creates synthetic data points for the minority classes while RandomUnderSampler reduces the number of data points in the major class, leading to data balancing. The technique eliminates the bias of the model towards the majority classes and ensures that all the attack categories are given equal treatment. Balanced data results in more reliable learning, higher sensitivity to rare risks and improved scores in precision, recall, and F1.

C. Training and Testing

The dataset can be split into training and testing to ensure proper assessment of the model's performance. Usually, a significant portion of this data is dedicated to training, enabling algorithms to identify patterns and relationships in IoT and IIoT data. During training, the testing set is not provided, so it's possible to evaluate the prediction skills without any prejudice. This method can be used to assess important metrics like accuracy, recall, precision, and F1 score, ensuring that intrusion detection models perform effectively in real-world scenarios.

D. Algorithms

1) Convolutional Neural Network (CNN): CNN automatically learns spatial feature hierarchies from network traffic, identifies complicated patterns and abnormalities. It enables intrusion detection with high accuracy, can effectively handle big data, is able to detect subtle attack characteristics and enhances the real-time responsiveness and robustness of IoT and IIoT networks.

$$S(i,j) = \sum_{m}\sum_{n} I(i+m, j+n) \cdot K(m,n) \quad (1)$$

2) Deep Neural Network (DNN): DNNs can capture the non-linear relationships between features from IoT and IIoT through multiple hidden layers. It learns subtle feature interactions, detects normal and harmful activity, handles different datasets, and maintains consistent anomaly detection while improving the accuracy of anomaly detection and speed of detection.
3) Random Forest (RF): RF is an ensemble model that integrates many decision trees, which are trained on different subsets of features, to achieve stability, avoid overfitting, manage high-dimensional data and address class imbalance, while also allowing for interpretable feature importance for accurate and stable ID.

$$Gini = 1 - \sum_{i=1}^{C} (Pi)^2 \quad (2)$$

4) Decision Tree (DT): DT hierarchically partitions the data to create understandable classification rules, including the ability to detect regular and malicious traffic quickly and accurately. It is suitable for categorical and numerical features, and can be the basis of ensemble models.

$$I\,(i) = 1 - \sum_{i=1}^{k} p_i^2 \quad (3)$$

5) LightGBM: LightGBM builds successive gradient-boosted trees to correct previous mistake and effectively processes high-dimensional and imbalanced network data. It enhances detection accuracy, features importance, and allows for quick, scalable and reliable intrusion monitoring.
6) Bagging Classifier: Bagging trains several base classifiers on a random subset of data; reduces variance and overfitting. It improves generalization capabilities, addresses imbalanced classes, enhances detection capabilities for a broader spectrum of attacks, and guarantees scalable, interpretable and reliable classification of network data.
7) Stacking Classifier: Stacking is a combination of several base models along with a meta-learner that exploit each individual classifiers' strengths. It reduces the effects of bias and variance, enhances generalization, identifies complex intrusions and guarantees robust high accuracy classification for a wide range of IoT and IIoT data sets.

$$\hat{y} = g(Y_{base}) = g\big(f_1(x), f_2(x), \dots, f_m(x)\big) \quad (4)$$

8) Voting Classifier: Voting is a combination of many classifiers by majority or weighted voting that enhances the robustness and accuracy. It takes advantage of complementary model strengths, class imbalance and robust, interpretable, scalable real-time detection of normal and harmful network activities.

$$\hat{y} = argmax_c \left(\sum_{i=1}^{n} II\,(\hat{y}_i = c) \right) \quad (5)$$

E. Integration of XAI and Flask Framework

It features a user-friendly UI, implemented using the Flask framework, and includes XAI features. The frontend has a registration and login function, which is powered by a SQLite database so users can operate on a secure basis. Network feature values are inputted by users and preprocessed before being supplied to trained ML models. The algorithms then process the data and make predictions for intrusion detection, which is clearly displayed on the frontend, making it real-time usable and accessible.

The system also includes the Voting Classifier ensemble for an improvement of the accuracy and robustness of the prediction. In addition, XAI techniques like LIME and SHAP are employed to interpret feature dependencies and significance, offering interpretable insights into the models' decision-making processes. The synergy of the powerful ensemble learning with feature explainability ensures that the system is reliable in detecting cyber threats, and understandable, thereby fostering user confidence and knowledge in IoT and IIoT network security scenarios.

IV. EXPERIMENTAL RESULTS

1) Accuracy: The accuracy of a test is based on its ability to correctly classify patient and healthy cases. Calculate the proportion of true positives and true negatives for all evaluated cases as the accuracy measure for measuring the accuracy of a test. Where TP represents True Positive, TN represents True Negative, FP represents False Positive, and FN represents False Negative. Mathematically speaking, this could be represented as

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} (6)$$

2) Precision: Precision is the number of true cases or samples correctly classified divided by the number of positive cases or samples. Thus, the precision can be calculated using the following formula:

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} (7)$$

3) Recall: Recall is a metric in ML used for measuring how well a model can identify all of the instances of a particular class. The ratio of correctly predicted positive observations to the total number of actual positive observations, and gives information on how complete a model is when it comes to collecting positive examples of a class.

$$Recall = \frac{TP}{TP + FN} (8)$$

4) F1-Score: F1 F1 Score is a ML evaluation metric that evaluates the accuracy of a model. It is a combination of Precision and Recall of the model. The accuracy statistic is the number of times the model predicted correctly on the entire set of data.

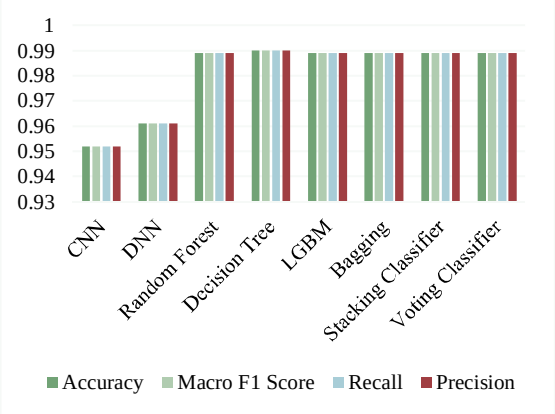
$$F1\ Score = 2 * \frac{Recall\ X\ Precision}{Recall + Precision} * 100 (9)$$

Table.1 Performance Evaluation – Wustl-IIoT 2021 Dataset

ML Model	Accuracy	Macro F1 Score	Recall	Precision
CNN	0.952	0.952	0.952	0.952
DNN	0.961	0.961	0.961	0.961
Random Forest	0.989	0.989	0.989	0.989
Decision Tree	0.990	0.990	0.990	0.990
LGBM	0.989	0.989	0.989	0.989
Bagging	0.989	0.989	0.989	0.989
Stacking Classifier	0.989	0.989	0.989	0.989
Voting Classifier	0.989	0.989	0.989	0.989

The overall performance in IoT and IIoT intrusion detection is the best for DT in Table.1 suggesting that the prediction accuracy is highest for Decision Tree.

Fig.5 Comparison Graph – Wustl-IIoT 2021 Dataset



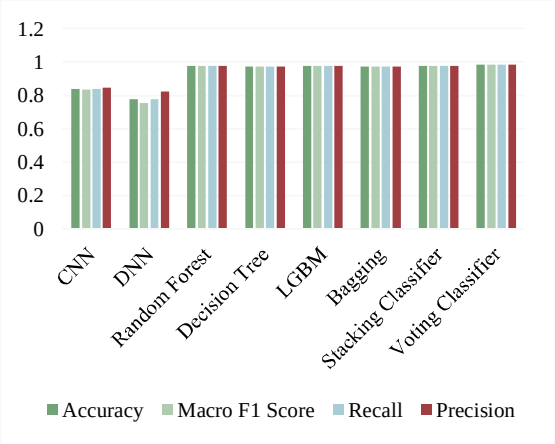
In Figure 5, the highest metrics achieved is 0.990 for DT. The accuracy, precision, recall, and macro F1 score are displayed in green, light green, blue, and red respectively.

Table.2 Performance Evaluation – ToN-IoT dataset

ML Model	Accuracy	Macro F1 Score	Recall	Precision
CNN	0.837	0.836	0.837	0.845
DNN	0.779	0.755	0.779	0.825
Random Forest	0.976	0.976	0.976	0.976
Decision Tree	0.972	0.972	0.972	0.973
LGBM	0.977	0.977	0.977	0.977
Bagging	0.973	0.973	0.973	0.973
Stacking Classifier	0.978	0.978	0.978	0.978
Voting Classifier	0.984	0.984	0.984	0.985

As shown in Table.2, Voting Classifier exhibits excellent performance, proving its strength and accuracy in the identification of network intrusions in IoT and IIoT environments.

Fig.6 Comparison Graph – ToN-IoT Dataset



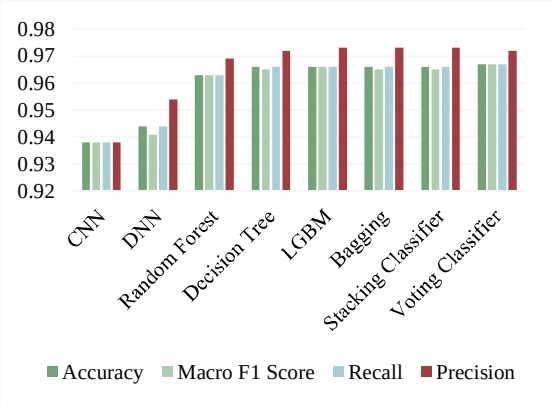
The performance of the Voting Classifier is best as seen in figure 6. The difference between model performance is represented using Accuracy (green), Precision (red), Recall (blue) and Macro F1 Score (light green).

Table.3 Performance Evaluation – Edge-IIoT dataset

ML Model	Accuracy	Macro F1 Score	Recall	Precision
CNN	0.938	0.938	0.938	0.938
DNN	0.944	0.941	0.944	0.954
Random Forest	0.963	0.963	0.963	0.969
Decision Tree	0.966	0.965	0.966	0.972
LGBM	0.966	0.966	0.966	0.973
Bagging	0.966	0.965	0.966	0.973
Stacking Classifier	0.966	0.965	0.966	0.973
Voting Classifier	0.967	0.967	0.967	0.972

In Table.3, the Voting Classifier had the best overall performance, outperforming the other models tested.

Fig.7 Comparison Graph – Edge-IIoT Dataset



As can be seen in figure 7, the Voting Classifier achieved the highest performance in all the metrics – Accuracy (green), Macro F1 Score (light green), Recall (blue), and Precision (red).

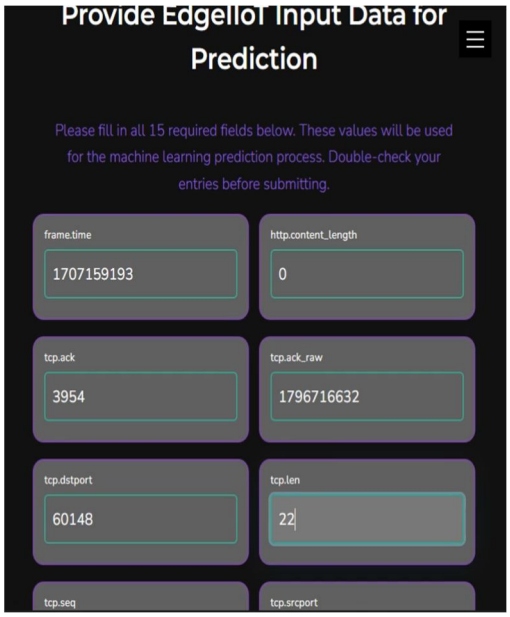


Fig.8 –Enter Input data for EdgeIIoT

In Figure 8, users enter EdgeIIoT dataset parameters to help the algorithm detect and classify certain network threat types.

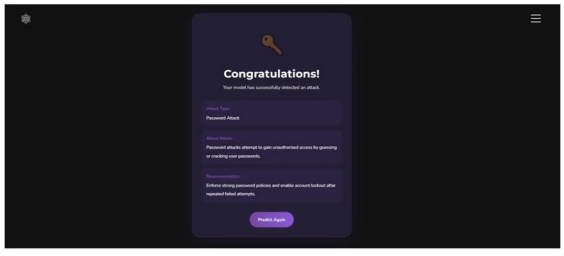


Fig.9 Predicted Results for EdgeIIoT

The system is able to process the data from the EdgeIIoT and accurately classify the attack as "Password Attack" as shown in figure 9.

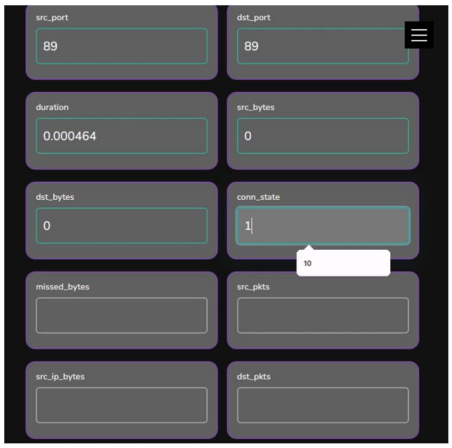


Fig.10 Enter Input Data for ToN-IOT

The users input the values of the ToN-IoT datasets into the system as shown in Fig.10, which it uses to compute the network activity and identify specific types of attacks.

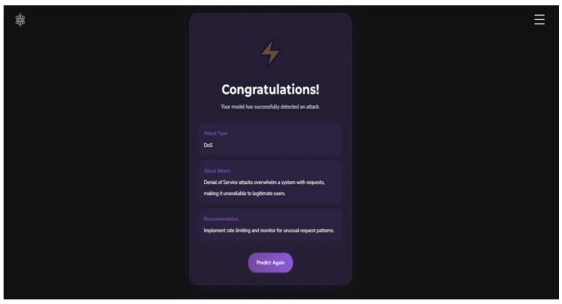


Fig.11 Predicted Results for ToN-IOT

The overall system is shown in figure 11 which is able to identify the network intrusion as a DOS attack.

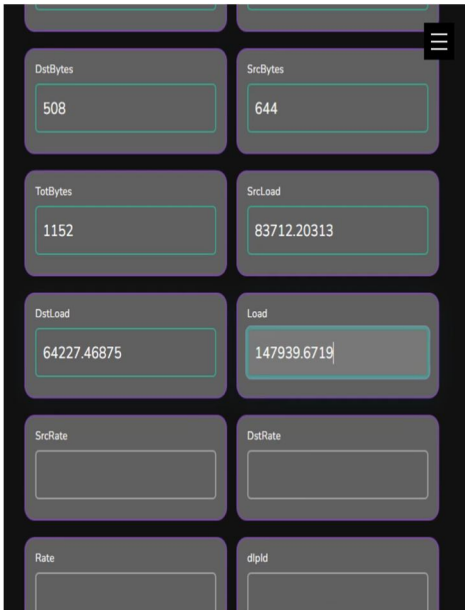


Fig.12 Enter Input Data for Wustl-IIOT

The users provide the data parameters for WUSTL-IIoT in Fig.12, enabling the system to analyze the network traffic and identify some type of attacks.

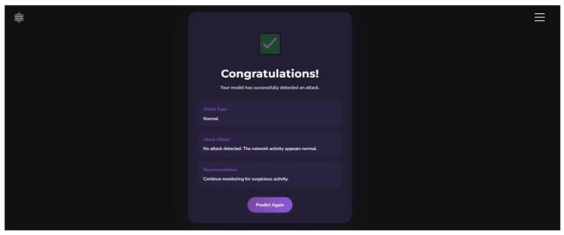


Fig.13 Predicted Results for Wustl-IIOT

The system is able to analyze the WUSTL-IIoT input and correctly identify the network behavior is normal, meaning that no assaults were detected in Figure 13.

V. CONCLUSION

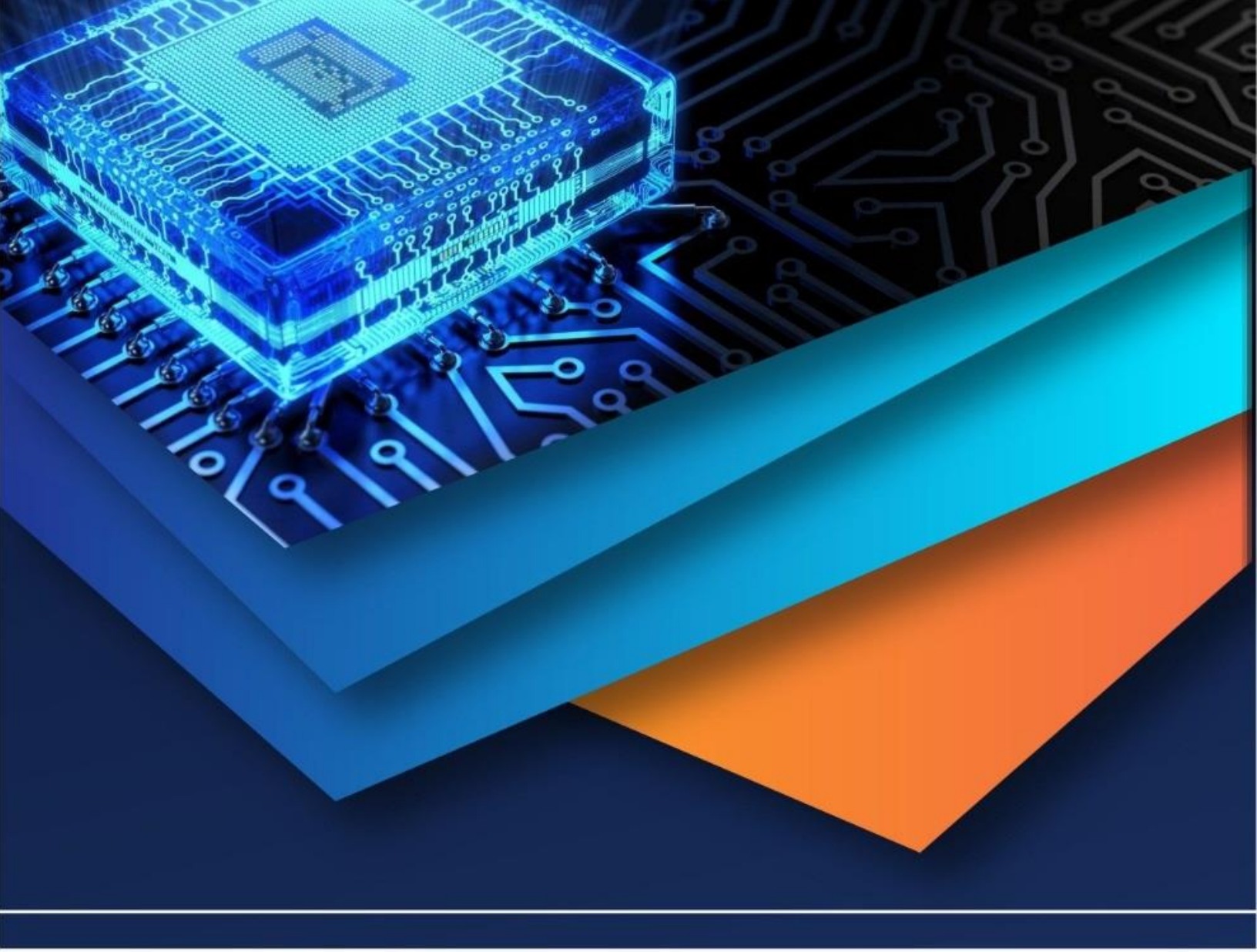
The paper demonstrates that the lightweight ML models can be integrated with powerful ensemble methods to accurately detect intrusions in IoT and IIoT networks. The system was able to effectively manage high-dimensional and imbalanced network traffic data across varied datasets thanks to comprehensive preprocessing, chi-squared feature selection, and class balancing with SMOTE and undersampling. Applying each of the classifiers together with ensemble methods significantly boosted detection reliability and generalization. The Voting Classifier has consistently performed better than all the other models tested with an accuracy of 98.4%, 98.9%, and 96.7% on the ToN_IoT, WUSTL-IIoT 2021, and Edge-IIoTset datasets, respectively, thus showing its robustness to detect heterogeneous attack patterns. The Explainable AI techniques, such as SHAP and LIME, were employed to enhance model transparency, offering clear explanations on feature contributions and boosting trust in automated decisions. For real-world application the intrusion detection framework was designed using Flask framework that offers a lightweight web based user interface, secure user sign-up, sign-in, real-time data entry, preprocessing, and prediction display through SQLite. In general, the recommended IDS offers a precise, comprehensible, and deployable answer to proactive cyber safety checking in a contemporary IoT and IIoT infrastructure.

Future efforts will involve developing the system with an ML-based intrusion detection solution and application on the microcontroller platforms that are widely used in IoT and IIoT applications. With this approach a comprehensive evaluation of the performance in realistic hardware and network scenarios will be possible. Study of energy use, latency and adaptability to dynamic traffic patterns in resource constrained environments will be emphasized. Additionally, other optimization techniques could be explored to reduce computation cost without compromising resilience and accuracy. Lightweight security protocols and edge AI systems will be integrated allowing for smooth detection and response. The upgrades are meant to make the intrusion detection system more useful, scalable and resilient, making it a better fit for industrial, large-scale IoT environments.

REFERENCES

[1] Alam, K., Monir, M. F., Hassan, Z., & Habib, M. T. (2024, October). Optimizing IoT Network Intrusion Detection: A Deep Learning Approach. In 2024 7th Conference on Cloud and Internet of Things (CIoT) (pp. 1-5). IEEE.
[2] Lazzarini, R., Tianfield, H., & Charissis, V. (2023). A stacking ensemble of deep learning models for IoT intrusion detection. Knowledge-Based Systems, 279, 110941.
[3] Kim, Y. G., Ahmed, K. J., Lee, M. J., & Tsukamoto, K. (2022, August). A Comprehensive Analysis of Machine Learning-Based Intrusion Detection System for IoT-23 Dataset. In International Conference on Intelligent Networking and Collaborative Systems (pp. 475-486). Cham: Springer International Publishing.
[4] Morshedi, R., Matinkhah, S. M., & Sadeghi, M. T. (2024). Intrusion detection for IoT network security with deep learning. Journal of AI and Data Mining, 12(1), 37-55.
[5] Eren, K. K., Küçük, K., Özyurt, F., & Alhazmi, O. H. (2025). Simple Yet Powerful: Machine Learning-Based IoT Intrusion System With Smart Preprocessing and Feature Generation Rivals Deep Learning. IEEE Access.
[6] J. A. Beauty Angelin and C. Priyadharsini, ‘‘Deep learning based network based intrusion detection system in industrial Internet of Things,’’ in Proc. 2nd Int. Conf. Intell. Data Commun. Technol. Internet Things (IDCIoT), Jan. 2024, pp. 426–432.
[7] M. Ramaiah and M. Y. Rahamathulla,‘‘Securing the industrial IoT: A novel network intrusion detection models,’’ in Proc. 3rd Int. Conf. Artif. Intell. For Internet Things (AIIoT), May 2024, pp. 1–6.
[8] S. I. Popoola, A. L. Imoize, M. Hammoudeh, B. Adebisi, O. Jogunola, and A. M. Aibinu, ‘‘Federated deep learning for intrusion detection in consumer-centric Internet of Things,’’ IEEE Trans. Consum. Electron., vol. 70, no. 1, pp. 1610–1622, Feb. 2024.
[9] Z. Ahmed and S. S. Askar, ‘‘EdgeGuard: Machine learning for proactive intrusion detection on edge networks,’’ Artif. Intell. Cybersecurity, vol. 1, pp. 37–43, Jun. 2024.
[10] R. S. Tiwari, D. Lakshmi, T. K. Das, A. K. Tripathy, and K.-C. Li, ‘‘A lightweight optimized intrusion detection system using machine learning for edge-based IIoT security,’’ Telecommun. Syst., vol. 87, no. 3, pp. 605–624, Nov. 2024.

[11] F. Mesadieu, D. Torre, and A. Chennamaneni, ‘‘Leveraging deep reinforce ment learning technique for intrusion detection in SCADA infrastructure,’’ IEEE Access, vol. 12, pp. 63381–63399, 2024.
[12] A. M. Eid, B. Soudan, A. B. Nassif, and M. Injadat, ‘‘Comparative study of ML models for IIoT intrusion detection: Impact of data preprocessing and balancing,’’ Neural Comput. Appl., vol. 36, no. 13, pp. 6955–6972, May 2024.
[13] A. M. Eid, B. Soudan, A. B. Nassif, and M. Injadat, ‘‘Enhancing intrusion detection in IIoT: Optimized CNN model with multi-class SMOTE balancing,’’ Neural Comput. Appl., vol. 36, no. 24, pp. 14643–14659, Aug. 2024.
[14] A. H. Farea and K. Küçük, ‘‘Machine learning-based intrusion detection technique for IoT: Simulation with cooja,’’ Int. J. Comput. Netw.Inf. Secur., vol. 16, no. 1, pp. 1–23, Feb. 2024.
[15] M. S. Alshehri, O. Saidani, F. S. Alrayes, S. F. Abbasi, and J. Ahmad, ‘‘A self-attention-based deep convolutional neural networks for IIoT networks intrusion detection,’’ IEEE Access, vol. 12, pp. 45762–45772, 2024.
[16] K. Bansal and A. Singhrova, ‘‘Review on intrusion detection system for IoT/IIoT-brief study,’’ Multimedia Tools Appl., vol. 83, no. 8, pp. 23083–23108, Aug. 2023.
[17] M. Nuaimi, L. C. Fourati, and B. B. Hamed, ‘‘Intelligent approaches toward intrusion detection systems for industrial Internet of Things: A systematic comprehensive review,’’ J. Netw. Comput. Appl., vol. 215, Jun. 2023, Art. no. 103637.
[18] G.Guo,X.Pan,H.Liu,F.Li,L.Pei,andK.Hu,‘‘AnIoTintrusiondetection system based on TON IoT network dataset,’’ in Proc. IEEE 13th Annu. Comput. Commun. Workshop Conf. (CCWC), Mar. 2023, pp. 0333–0338.
[19] O. Belarbi, T. Spyridopoulos, E. Anthi, I. Mavromatis, P. Carnelli, and A. Khan, ‘‘Federated deep learning for intrusion detection in IoT networks,’’ in Proc. IEEE Global Commun. Conf. (GLOBECOM), vol. 3125, Dec. 2023, pp. 85–99.
[20] A. Oseni, N. Moustafa, G. Creech, N. Sohrabi, A. Strelzoff, Z. Tari, and I. Linkov, ‘‘An explainable deep learning framework for resilient intrusion detection in IoT-enabled transportation networks,’’ IEEE Trans. Intell. Transp. Syst., vol. 24, no. 1, pp. 1000–1014, Jan. 2023.
[21] D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, ‘‘Intrusion detection system using feature extraction with machine learning algorithms in IoT,’’ J. Sensor Actuator Netw., vol. 12, no. 2, p. 29, Mar. 2023.
[22] M. M. Shtayat, M. K. Hasan, R. Sulaiman, S. Islam, and A. U. R. Khan, ‘‘An explainable ensemble deep learning approach for intrusion detection in industrial Internet of Things,’’ IEEE Access, vol. 11, pp. 115047–115061, 2023.
[23] M. Wang, N. Yang, and N. Weng, ‘‘Securing a smart home with a transformer-based IoT intrusion detection system,’’ Electronics, vol. 12, no. 9, p. 2100, May 2023.
[24] S. Li, G. Chai, Y. Wang, G. Zhou, Z. Li, D. Yu, and R. Gao, ‘‘CRSF: An intrusion detection framework for industrial Internet of Things based on pretrained CNN2D-RNN and SVM,’’ IEEE Access, vol. 11, pp. 92041–92054, 2023.
[25] M. M. Rashid, S. U. Khan, F. Eusufzai, M. A. Redwan, S. R. Sabuj, and M. Elsharief, ‘‘A federated learning-based approach for improving intrusion detection in industrial Internet of Things networks,’’ Network, vol. 3, no. 1, pp. 158–179, Jan. 2023.
[26] A. M. Eid, A. B. Nassif, B. Soudan, and M. N. Injadat, ‘‘IIoT network intrusion detection using machine learning,’’ in Proc. 6th Int. Conf. Intell. Robot. Control Eng. (IRCE), Aug. 2023, pp. 196–201.
[27] T. Gaber, J. B. Awotunde, S. O. Folorunso, S. A. Ajagbe, and E. Eldesouky, ‘‘Industrial Internet of Things intrusion detection method using machinelearningandoptimizationtechniques,’’WirelessCommun.Mobile Comput., vol. 2023, pp. 1–15, Apr. 2023.
[28] B. Xu, L. Sun, X. Mao, R. Ding, and C. Liu, ‘‘IoT intrusion detection system based on machine learning,’’ Electronics, vol. 12, no. 20, p. 4289, Oct. 2023.
[29] Z. Benamor, Z. A. Seghir, M. Djezzar, and M. Hemam, ‘‘A comparative study of machine learning algorithms for intrusion detection in IoT networks,’’ Revue d’Intell. Artificielle, vol. 37, no. 3, pp. 567–576, Jun. 2023.
[30] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, ‘‘X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial Internet of Things,’’ IEEE Internet Things J., vol. 9, no. 5, pp. 3962–3977, Mar. 2022.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY