iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67594

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 13 Issue III Mar 2025- Available at www.ijraset.com

An Efficient IOT-Malware Classification Model Using Ensemble Machine Learning

Mrs. K. Harini[1], Ch. Rohini[2], P. Hasitha[3], K. Rakesh[4], N. Yuva Bhanu Prakash[5]
[1]*Assistant Professor, Raghu Institute of Technology, Visakhapatnam, Andhra Pradesh*
[2, 3, 4, 5]*Student, Raghu Institute of Technology, Visakhapatnam, Andhra Pradesh*

Abstract: ***In the realm of Internet of Things (IoT) security, malware classification is crucial for identifying, isolating, and mitigating the impacts of malicious software that exploits system vulnerabilities. Effective IoT malware classification employs various machine learning algorithms to enhance detection accuracy and system resilience. The most commonly used Machine Learning algorithms for Malware Classification are "Support Vector Machine (SVM), Random Forest (RF), Naive Bayes (NB), and Logistic Regression (LR)". Each of these algorithms has distinct advantages in identifying malware, but this study explores the creation of an Ensemble / Hybrid model by integrating the two most effective machine learning algorithms, which can lead to superior performance. The proposed model outperforms the simple models mentioned above and other related works discussed in the literature. Key performance metrics evaluated for this work includes Accuracy, Precision, Recall, F-Score. This advancement is vital for improving the security posture of IOT systems by providing a more reliable mechanism for early detection and effective response to emerging malware threats.***

Index Terms: ***Malware classification, Machine Learning, Ensemble Model, Support Vector Machine (SVM), Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR), Hybrid Model.***

I. INTRODUCTION

The Internet of Things (IoT) has revolutionized technology by linking devices, enhancing automation, and enhancing efficiency. However, due to this connectivity, there are security risks, particularly from malware, which can infect systems and disrupt operations. New solutions are required because conventional signature-based malware detection methods are useless against evolving threats. Through pattern identification in data, machine learning (ML) is a workable technique for IoT malware classification and detection of upcoming threats. Support Vector Machine (SVM), Random Forest (RF), Naive Bayes (NB), and Logistic Regression (LR) are key algorithms. High-dimensional space is where SVM excels, RF combines decision trees for precision, NB is simple yet efficient, and LR provides probabilistic binary classification. The computational requirements of SVM, the overfitting risk of RF, the independence assumption of NB, and the difficulty of LR with non-linear relationships are just some of the limitations of each method. Multiple models are integrated in ensemble methods to avoid these problems and enhance efficiency. The research proposes an Ensemble/Hybrid model that integrates two effective methods to enhance robustness and accuracy. For ensuring effective malware detection, performance will be measured with metrics like Accuracy, Precision, Recall, F-score. The Internet of Things increases their attack surface, making them more vulnerable to malware. Ensemble models provide adaptable, dynamic methods for addressing vulnerabilities. Iterative updates, user testing, and model combination are all part of realistic implementation. Through the construction of a robust ensemble model for early malware detection, this research enhances IoT security and safeguards systems against evolving threats.

II. RELATED WORKS

According to several recent papers, IOT - Malware classification in the context of machine learning and practices is not yet complete, and it remains an ongoing study topic, as evidenced by the following:

A. Kumar and T. J. Lim's[1] research paper "EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques" aims to improve IoT network security by detecting malware early. They used supervised algorithms like Random Forest, Gradient Boosting, and Support Vector Machines to detect fraudulent activities in real-world network traffic datasets. The study highlights the potential of lightweight machine learning models in resource-constrained IoT systems.

Y. Meidan et al.'s[2] paper, "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," aims to identify IoT botnet attacks using deep autoencoders for unsupervised anomaly detection. The researchers used a dataset from the UCI Machine Learning Repository, including traffic from nine IoT devices attacked by Bashlite and Mirai botnets. The model demonstrated high accuracy and a 100% true positive rate.

Gandotra, Bansal, and Sofat's[3] study explores malware analysis and categorization methods, focusing on polymorphic and metamorphic malware. The paper categorizes malware analysis into static and dynamic methods, emphasizing machine learning's ability to recognize patterns in behaviour. Techniques include Naive Bayes, Support Vector Machines, boosted decision trees, data mining, structural analysis, and image-based classification. The study emphasizes the importance of combining strategies to handle changing threats.

The study by Jamal, Hayat, and Nasir[4] uses Artificial Neural Networks (ANN) for malware detection and classification in Internet of Things networks. Using the Ton_IoT dataset, they achieve classification accuracy of 97.08% and detection accuracy of 94.17%. The method reduces feature engineering and offers a scalable solution for handling malware in complex network environments.

The study by Giang L. Nguyen et al.[5] highlights the need for early detection techniques to reduce the harm caused by IoT botnets. They propose a collaborative machine learning methodology using a dataset of 3,888 benign samples and 5,023 botnet samples. The method achieves 99.37% detection accuracy, enhancing real-time security in IoT settings.

García, Uhlíř, and Rehak's[6] 2014 study uses long-term network data collection to analyse botnet Command and Control (C&C) channel behaviours. They developed a unified behavioural model, a unique dataset, and comprehensive behavioural features, which can help identify and reduce botnet risks, and contribute to the development of heuristic and machine learning techniques.

The paper of Samantray and Tripathy[7] 2021 "An Opcode-Based Malware Detection Model Using Supervised Learning Algorithms" proposes a model for detecting malware based on opcode analysis. The authors employ supervised learning algorithms to classify malware, demonstrating the model's potential for improving detection accuracy. The research shows that opcode features are effective in distinguishing between benign and malicious software, providing a promising approach for malware detection.

Samantray and Tripathy's[8] "A Knowledge-Domain Analyser for Malware Classification" presents a knowledge-based model for classifying malware using a domain-specific approach. The model incorporates various classification techniques to enhance accuracy in detecting malware. It focuses on understanding the specific characteristics of malware and how to categorize them effectively.

Torabi, Bou-Harb, and Assi[9] have developed a novel methodology for identifying and grouping IoT malware into families using a multi-dimensional deep learning framework. The approach uses a multi-dimensional feature space, combining recurrent and convolutional neural networks. The framework outperformed current techniques in classification accuracy and robustness, demonstrating potential for proactive malware defence in IoT environments.

Sharma et al.'s[10] "A Systematic Review of IoT Malware Detection using Machine Learning" provides a comprehensive analysis of machine learning methods for identifying malware in IoT environments. They evaluate various techniques, including supervised, unsupervised, and hybrid methods, and highlight the importance of integrating ensemble methods and improving feature extraction techniques for reliable malware detection systems.

A. A. Ali et al.'s[11] paper "Securing the Internet of Things (IoT) Application: Best Practices and Challenges in IoT Software" analyses security issues in IoT applications, including poor update rules, weak authentication procedures, and unsecured communication interfaces. They suggest strategies like frequent software upgrades, strong encryption, and multi-factor authentication, as well as privacy-aware setup and security-by-design approaches.

The paper "Machine Learning for Healthcare-IoT Security: A Review and Risk Mitigation" by M. A. Khatun et al[12]. examines security issues and risk-reduction techniques for the Healthcare Internet of Things (H-IoT). It highlights weaknesses in systems due to new risks like 5G-IoT and generative AI, and suggests using machine learning and deep learning to develop strong authentication procedures and detect anomalies.

Nakahara et al[13]. (2022) propose a hybrid method for identifying malware in IoT devices using lightweight flow data, machine learning techniques, and a whitelist-based detection strategy. This approach addresses resource-constrained devices' computational load and improves IoT security. The method effectively processes known safe actions and manages anomalies, demonstrating potential for practical implementations with minimal computing overhead.

The paper "Enhancing Security in IoT and IIoT Networks: An Intrusion Detection Scheme Leveraging Deep Transfer Learning" by Basharat Ahmad et al[14]. presents an innovative method for protecting IoT and Industrial IoT networks using a deep transfer learning-based intrusion detection system. The method addresses challenges in diverse IoT contexts and lack of labelled training data, improving detection accuracy and reducing computational overhead. The strategy is flexible and scalable, making it a strong option for dynamic IoT networks.

Z. Li et al[15]. present a smart plug system for real-time IoT malware detection, using a hybrid strategy combining machine learning algorithms and a whitelist mechanism. The system, which uses timestamps, IP addresses, and packet counts, is efficient and offers a proactive approach to IoT network security.

The A. D. B, M. K. S and P. Joshi's[16] study "Advancing IoT Security: A Stacked Hybrid AI Approach for Anomaly Detection" uses a hybrid AI-based model to enhance IoT security. The model uses a stacked ensemble architecture, combining multiple machine learning models for efficient detection. The model outperforms conventional techniques with high detection rates and low false positives, making it a scalable and effective monitoring method.

Nanthiya et al.'s[17] study uses the IoT-23 Botnet dataset to detect Distributed Denial of Service attacks in IoT networks using Support Vector Machines (SVM). The machine learning technique distinguishes between regular and malicious traffic, achieving high accuracy and minimal false positive rates, making it a practical solution for protecting IoT environments.

Ammar Odeh and Anas Abu Taleb's[18] paper "Ensemble-Based Deep Learning Models for Enhancing IoT Intrusion Detection" (2023) introduces a novel model that uses ensemble deep learning techniques for intrusion detection. The model integrates CNN, GRU, and LSTM networks, enhancing detection accuracy and reducing false positives and false negatives. The study shows promising performance of over 90% recognition.

Ahanger, Khan, and Masoodi's[19] 2023 paper presents an ensemble machine learning method for IoT network intrusion detection, integrating k-Nearest Neighbours, Random Forests, and Decision Trees, demonstrating improved detection performance and robustness compared to single models.

III. Methods and Algorithms

A. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm for classification and regression tasks that employs a nonlinear mapping technique to map input patterns into a higher dimensional feature space. Seeks to identify the optimal separating hyperplane in a feature space to maximize the margin between classes, and it is especially useful for complicated non-linear data when kernel functions are employed.

RF is an extremely strong ensemble machine learning algorithm that uses multiple decision trees. It has outstanding accuracy and is highly resistant to overfitting by aggregating outputs through voting (classification) or averaging (regression). RF performs bootstrap sampling to develop trees, builds them up to their full size without pruning, and averages or votes by majority to combine predictions.

B. Naive Bayes (NB)

This algorithm is fast and efficient for large datasets because it relies on the Bayes theorem, which assumes independence of features and calculates the probability of a class given observed feature values and thus is fast and efficient with large datasets. NB is ideal for low-resource IoT devices as it is fast to train and predict.

C. Logistic Regression (LR)

LR is a statistical method for predicting the probability of a binary outcome (e.g., yes/no) suitable for understanding feature contributions. It assumes that the log-odds of the independent and dependent variables are linearly related. Due to its efficacy, simplicity, and probabilistic outputs, LR is an ideal option for real-time IoT virus detection.

Architecture Diagram

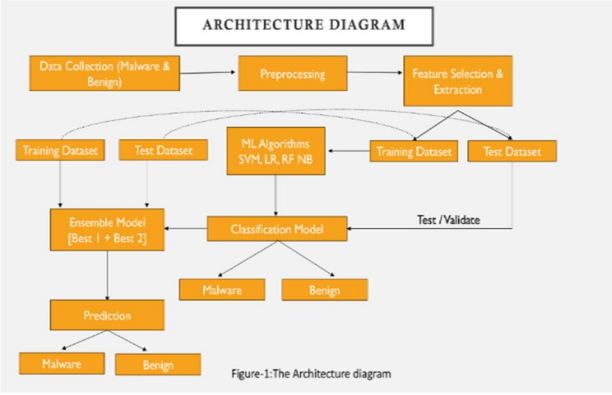


Figure 1: Architecture of the proposed system

D. Work Flow Of The Architecture

An explanation of the architecture as in fig. 1 includes-

- Data Collection (Benign & Malware) - Prepare datasets with both benign and malignant (malware) samples. The machine learning models are trained and tested on these datasets.
- Preprocessing - Clean and prepare the collected data by removing redundant features, missing values, and noise. Normalize and prepare the data to make it suitable for machine learning algorithms.
- Feature Selection & Extraction - Extract relevant features (e.g., file signatures, network usage, and system calls) from the raw data. Lower the dimensionality of the dataset using feature selection methods to identify what features are most important for identifying malware.
- Dataset Splitting - Split pre-processed data into training and test datasets. The machine learning models are trained on the training dataset, and the test dataset evaluates their performance.
- Algorithms for Machine Learning (SVM, LR, RF, NB) - These individual machine learning models, SVM (Support Vector Machine), LR (Logistic Regression), RF (Random Forest), and NB (Naive Bayes), can be trained. Each of the algorithms is trained to be able to identify between benign and malicious data.
- Classification Model - The models classify input data as Malware and Benign upon training. The classification results of each model are analysed to check how good it is on its own.
- Best 1 + Best 2 Ensemble Model - Select the best-performing algorithms (for instance: the top two classifiers based on metrics such as accuracy or F-score). To enhance overall classification performance, aggregate their predictions using an ensemble method (for instance, weighted voting).

The last prediction is provided by the ensemble model, which classifies newly obtained input data as Malware or Benign. For enhanced precision and robustness, this prediction utilizes the synergistic benefits of the selected approaches

E. Data Flow Diagram

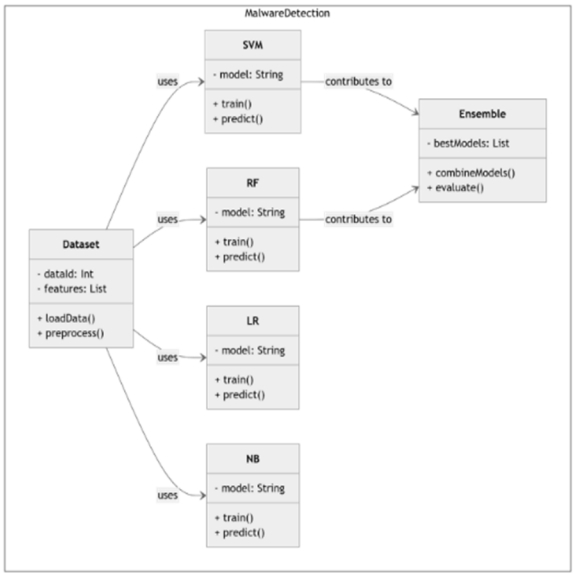


Figure 2: Data Flow diagram

An explanation of the Data Flow diagram as in fig. 2 includes-

F. Essential Elements

Dataset: This serves as the system's core. It contains the unprocessed data needed to train and assess the malware detection models.
Characteristics -

- features: A collection of characteristics taken from the malware samples (e.g., file size, API calls, byte frequencies)
- dataId: A distinct identification for every dataset.

Techniques -

- Preprocess(): Gets the data ready for model training (e.g., cleaning, normalisation, feature scaling).
- loadData(): Loads the raw malware data.
- Models for Malware Detection: Four distinct machine learning models are employed by the system to classify malware are Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB)

The characteristics and techniques of each model are as follows - The particular algorithm either LR, NB, RF, or SVM.

- train(): Uses the pre-processed dataset to train the model.
- Predict(): Forecasts fresh, untested malware strains.
- Ensemble: To increase overall accuracy, this component integrates the predictions of the separate models.
- Features: bestModels: A compilation of the top-performing models chosen using evaluation measures.
- Methods: combineModels(): This function aggregates the predictions of the chosen models (e.g., by stacking, voting, or averaging).
- evaluate(): Assesses the ensemble model's performance using measures such as recall, accuracy, and precision.

The entire procedure includes -

- Preprocessing and Data Loading: The dataset loads unprocessed malware data.and prepares it for use.
- Model Training: The preprocessed data is used to train each malware detection model.
- Ensemble Creation: The Ensemble incorporates the predictions of the top-performing models.
- Evaluation: The Ensemble's malware detection accuracy is assessed.

The total accuracy and resilience of malware detection may be increased by using this system design, which gives users freedom in selecting and mixing several machine learning models.

IV. PERFORMANCE METRICS

There are various metrics which we can use to evaluate the performance of ML algorithms, classification as well as regression algorithms.

- True Positive (TP) = Observation is positive, and is predicted to be positive.
- False Negative (FN) = Observation is positive, but is predicted negative.
- True Negative (TN) = Observation is negative, and is predicted to be negative.
- False Positive (FP) = Observation is negative, but is predicted positive.

A. Accuracy

For binary label classification, the accuracy is calculated as:

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

B. Recall

For binary label classification, the recall is calculated as:

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

C. Precision

For binary label classification, the precision is calculated as:

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

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

D. F – Score

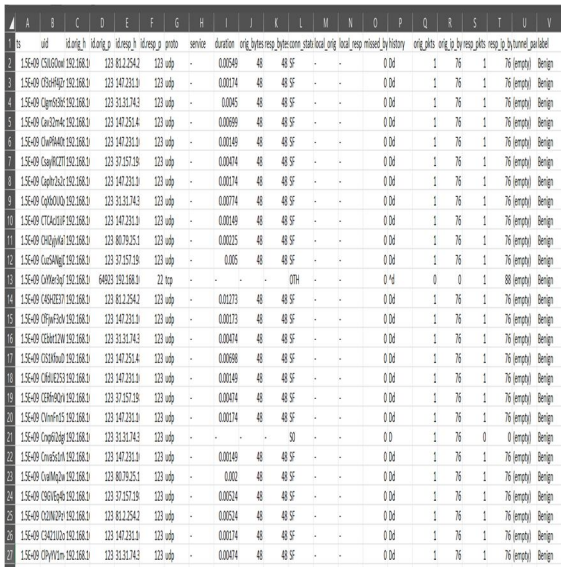
This score will give us the harmonic mean of precision and recall. F1 score is having equal relative contribution of precision and recall.

*F1 = 2 * (precision * recall) / (precision + recall)*

V. EXPERIMENTAL SETUP

A. Dataset

The "IoT-23 Malware Traffic Dataset" is a collection of network traffic logs primarily focused on distinguishing between benign and malicious activities in IoT-based networks. This dataset was gathered from Kaggle and contains 175,097 records with 22 attributes, representing various aspects of network communication. It serves as a valuable resource for analysing network behaviour, detecting anomalies, and developing malware detection models



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	proto	service	duration	orig_bytes	resp_bytes	conn_state	local_orig	local_resp	missed_bytes	history	orig_pkts	orig_ip_by	resp_pkts	resp_ip_by	tunnel_pa	label
2	1.5E+09	CSL6O2oel	192.168.1(	123	81.2.254.2	123	udp	-	0.00549	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
3	1.5E+09	CEbH4Qr	192.168.1(	123	147.231.1(	123	udp	-	0.00174	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
4	1.5E+09	CJgmSr26f	192.168.1(	123	31.31.74.3	123	udp	-	0.0045	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
5	1.5E+09	Cao82mr4c	192.168.1(	123	147.251.4(	123	udp	-	0.00699	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
6	1.5E+09	CIaPR4dQr	192.168.1(	123	147.231.1(	123	udp	-	0.00149	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
7	1.5E+09	CsayJRCZTI	192.168.1(	123	37.157.19(	123	udp	-	0.00474	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
8	1.5E+09	CqqhrZs2c	192.168.1(	123	147.231.1(	123	udp	-	0.00174	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
9	1.5E+09	CqfbOiXQy	192.168.1(	123	31.31.74.3	123	udp	-	0.00774	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
10	1.5E+09	CTCAo1IiF	192.168.1(	123	147.231.1(	123	udp	-	0.00149	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
11	1.5E+09	CHQpVaI	192.168.1(	123	80.79.25.1	123	udp	-	0.00025	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
12	1.5E+09	CucSANgE	192.168.1(	123	37.157.19(	123	udp	-	0.005	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
13	1.5E+09	CoYKe3qI	192.168.1(	64823	192.168.1(	22	tcp	-	-	-	-	OTH	-	-	0	vd	0	0	1	88	(empty)	Benign
14	1.5E+09	C49rH2E97	192.168.1(	123	81.2.254.2	123	udp	-	0.01273	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
15	1.5E+09	CPjyFI3ch	192.168.1(	123	147.231.1(	123	udp	-	0.00173	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
16	1.5E+09	CE8nt12W	192.168.1(	123	31.31.74.3	123	udp	-	0.00474	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
17	1.5E+09	CG1tXou0	192.168.1(	123	147.251.4(	123	udp	-	0.00698	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
18	1.5E+09	CJf6IJCE5	192.168.1(	123	147.231.1(	123	udp	-	0.00149	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
19	1.5E+09	CEHn9QrV	192.168.1(	123	37.157.19(	123	udp	-	0.00474	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
20	1.5E+09	CVmnFn1S	192.168.1(	123	147.231.1(	123	udp	-	0.00174	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
21	1.5E+09	Cnqv0Jdgr	192.168.1(	123	31.31.74.3	123	udp	-	-	-	-	S0	-	-	0	D	1	76	0	0	(empty)	Benign
22	1.5E+09	Cmx6s1rN	192.168.1(	123	147.231.1(	123	udp	-	0.00149	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
23	1.5E+09	CraMq7w	192.168.1(	123	80.79.25.1	123	udp	-	0.002	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
24	1.5E+09	C9GYEe4b	192.168.1(	123	37.157.19(	123	udp	-	0.00524	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
25	1.5E+09	C2DNZ0Pr	192.168.1(	123	81.2.254.2	123	udp	-	0.00524	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
26	1.5E+09	C3421UDz	192.168.1(	123	147.231.1(	123	udp	-	0.00174	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign
27	1.5E+09	CDyfVI1m	192.168.1(	123	31.31.74.3	123	udp	-	0.00474	48	48	SF	-	-	0	Dd	1	76	1	76	(empty)	Benign

Figure 3: Data set

B. Dataset Features

1) Timestamp (ts) – Represents the time of the network event.
2) Unique ID (uid) – A unique identifier for each connection session.
3) Source & Destination IPs (id.orig_h, id.resp_h) – Identifies the communicating devices.
4) Source & Destination Ports (id.orig_p, id.resp_p) – Specifies port numbers used for communication.
5) Protocol (proto) – Indicates the communication protocol (e.g., TCP, UDP).
6) Service (service) – Specifies the network service in use.
7) Duration (duration) – The time length of a network session.
8) Traffic Statistics: Includes orig_bytes, resp_bytes, orig_pkts, resp_pkts, which provide insights into packet exchange and data flow.
9) Connection State (conn_state) – Defines the status of the connection.
10) Missed Bytes (missed_bytes) – Number of lost bytes during transmission.
11) History (history) – Represents the sequence of events in a connection.
12) Local Origin & Response (local_orig, local_resp) – Flags indicating whether the communication originated locally.
13) Tunnel Parents (tunnel_parents) – Identifies any encapsulating network tunnels.
14) Label (label) – Classifies each connection as either Benign or associated with a specific malware type.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

C. Data Handling and Preprocessing

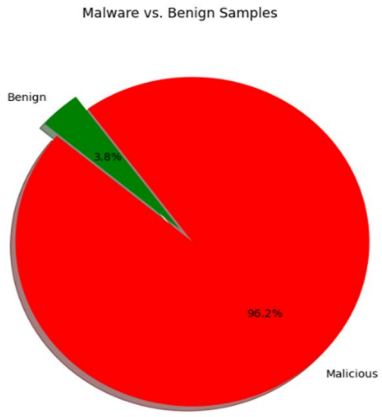


Figure 4: Pie Chart

The Fig. 4 above shows that the dataset has more malicious samples and very less benign samples. The dataset is an imbalanced one. In order to balance this dataset, we will use one balancing technique called SMOTE, after a while.
To ensure the dataset is suitable for machine learning models, various data cleaning and preprocessing steps were performed:

1) Handling Missing Values:
- Checked for null values using df.isnull().sum().
- Removed rows containing null values using df.dropna().
- Dataset shape was recorded before and after dropping null values.

2) Target Column Analysis:
- Examined the unique values in the label column.
- Confirmed that it represents a binary classification problem with two classes: Benign and Malicious.

3) Target Column Mapping:
- Converted string labels ('Benign', 'Malicious') into numerical values (0 and 1) using mapping and the map() function.

4) Feature Selection Using Correlation:
- Computed the correlation between numerical features using df.corr().
- Visualized the correlation matrix using a heatmap (seaborn.heatmap).
- Identified and dropped highly correlated features to reduce redundancy and enhance model performance using df.drop().

5) Feature Engineering Using Label Encoding:
- Converted categorical features (e.g., 'proto') into numerical representations using Label Encoding.
- Ensured compatibility with machine learning algorithms that require numerical input.

6) Data Balancing:
- Addressed class imbalance using SMOTE (Synthetic Minority Over-sampling Technique) from the imblearn library.
- Generated synthetic samples of the minority class to create a more balanced dataset.

7) Dataset Splitting:
- Divided the dataset into training and testing sets using train_test_split from sklearn.model_selection.
- Ensured proper evaluation of the model on unseen data.

VI. RESULTS

We compared the performance of a number of machine learning algorithms based on accuracy, precision, recall, and F1-score after applying data preprocessing methods, dataset balancing using SMOTE, and training. The results illustrate how effectively different classifiers detect IoT malware.

Naïve Bayes was accurate but slightly less so due to its feature independence assumption, while Random Forest and Support Vector Machine (SVM) had the highest accuracy and stability among the models that were tested. Leaning on the strength of multiple classifiers, the Voting Classifier, which combined a number of models, reflected better overall performance.

The complete comparison of each model's performance is given below:

	ACCURACY	PRECISION	RECALL	F1-SCORE
SUPPORT VECTOR MACHINE(SVM)	0.96	0.96	1.00	0.98
RANDOM FOREST(RF)	0.98	0.99	1.00	0.99
LOGISTIC REGRESSION(LR)	0.84	0.99	0.85	0.91
NAIVE BAYES(NB)	0.39	1.00	0.37	0.54
ENSEMBLE	0.99	0.99	0.99	0.98

Figure 5: Table of performance comparison

Based on the results from the above Fig. 5, we analyse the performance of each machine learning algorithm used in the project:

A. Support Vector Machine (SVM)

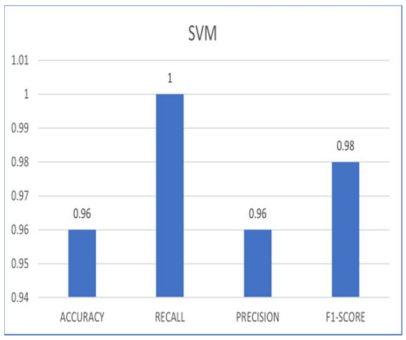


Figure 6: Performance graph of SVM

As mentioned in the above Fig 6 -

- Accuracy: **0.96**, Precision: **0.96**, Recall: **1.00**, F1-Score: **0.98**
- SVM performed exceptionally well with high precision and perfect recall, meaning it correctly identified all malicious instances. The high F1-score confirms a strong balance between precision and recall.

B. Random Forest (RF)

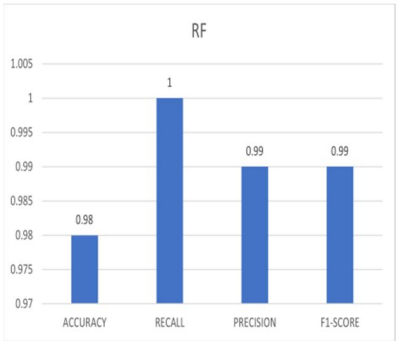


Figure 7: Performance graph of Random Forest

As we can see in the above Fig. 7 -

- Accuracy: **0.98**, Precision: **0.99**, Recall: **1.00**, F1-Score: **0.99**
- Random Forest achieved the second-highest accuracy, performing slightly better than SVM. Its **high precision and recall** indicate that it effectively distinguishes between benign and malicious traffic.

C. Logistic Regression (LR)

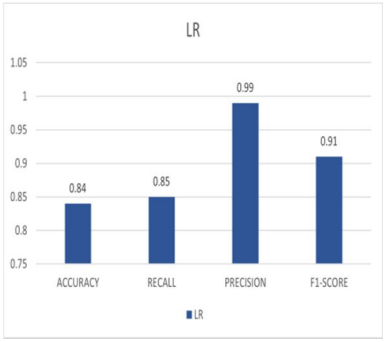


Figure 8: Performance graph of Logistic Regression

In the above Fig. 8 -

- Accuracy: **0.84**, Precision: **0.99**, Recall: **0.85**, F1-Score: **0.91**
- While Logistic Regression has a high precision of **0.99**, its recall is lower (**0.85**), meaning it misses some malicious instances. This suggests it is more prone to false negatives compared to RF and SVM.

D. Naïve Bayes (NB)

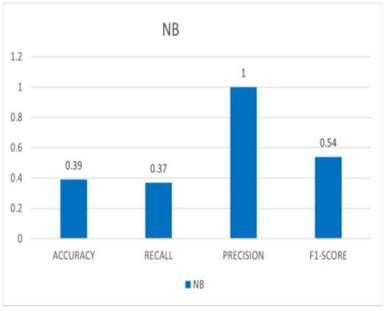


Figure 9: Performance graph of Naïve Bayes

The above Fig. 9 shows -

- Accuracy: 0.39, Precision: 1.00, Recall: 0.37, F1-Score: 0.54
- Naïve Bayes shows a high precision but very low recall, meaning it correctly identifies malicious instances when it predicts them, but it misses a significant number of actual malware cases. This indicates that Naïve Bayes is not reliable for this dataset.

E. Ensemble Model

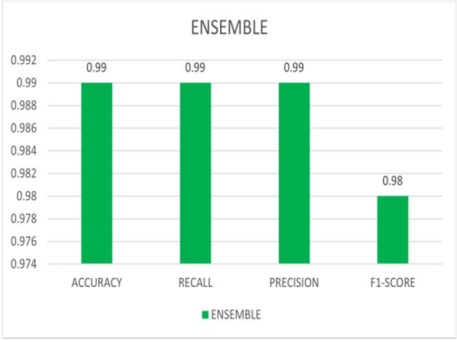


Figure 10: Performance of Ensemble Model

The Fig. 10 includes -

- Accuracy: 0.99, Precision: 0.99, Recall: 0.99, F1-Score: 0.98
- The Ensemble model performed the best, achieving the highest accuracy and a well-balanced precision-recall trade-off. By combining multiple classifiers, it effectively reduces errors and improves generalization.

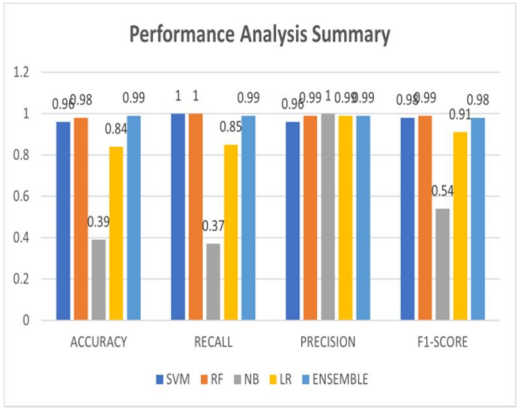


Figure 11: Final graph

The above Fig. 10 offers, a thorough summary of each algorithm's performance that was assessed for this project. It makes it simpler to assess the advantages and disadvantages of each model by combining its primary metrics—Accuracy, Precision, Recall, and F1-Score—into a single graphic with the best overall performance, Ensemble Model is in first place, closely followed by Random Forest. The Ensemble Model is the most dependable option for classifying IoT malware since it achieves the ideal balance of accuracy, precision, recall, and F1-score.

In terms of F1-score, Random Forest and Ensemble models surpass Support Vector Machine (SVM), which performs well, particularly in recall.

The performance of logistic regression is mediocre; it has a high precision but a low recall, which leads to more false negatives.

Bayes Naïve (NB) Despite having perfect precision, it has large lags in accuracy and recall, indicating that it is not appropriate for this dataset.

The Ensemble Model and Random Forest are shown as the leading candidates for use in IoT malware detection systems in this final graph, which aggregates the results of the individual algorithms.

In order to support the ultimate selection of the top-performing models for additional study and application, this visual comparison is essential.

VII. CONCLUSION AND FURTHER ENHANCEMENT

Our research shows that ensemble machine learning techniques greatly improve the resilience and accuracy of IoT malware categorization. The suggested model outperforms individual models by combining several methods, including SVM, Random Forest, Naive Bayes, and Logistic Regression. The model's superiority is confirmed by performance metrics such as accuracy, precision, recall, F1-score, AUC, and ROC. It performs better than single models with an accuracy of 99%. This advancement strengthens the security of IoT systems by enabling early detection and efficient response to emerging malware threats.

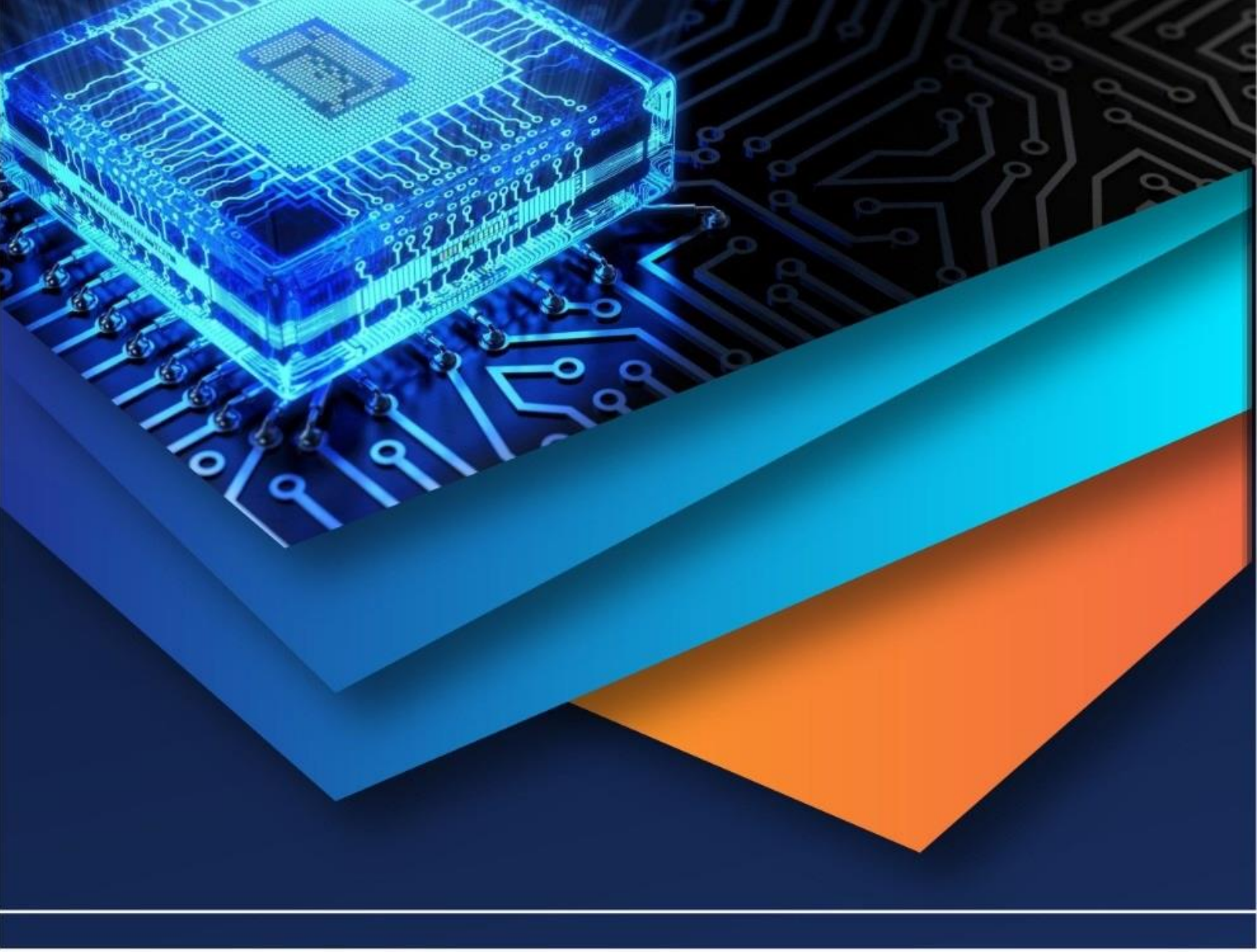
Advanced techniques like deep learning, graph-based approaches, and hybrid ensemble models can greatly improve malware classification in order to counter the growing threat of IoT malware. While graph-based methods capture the complicated relationships between malware operations, deep learning models, such CNNs and RNNs, are excellent at spotting complex patterns. Hybrid ensemble models provide enhanced robustness and performance. Explainable AI techniques can increase model transparency and confidence, while real-time detection and continuous learning are essential for responding to changing risks. The goal of these developments is to produce IoT security solutions that are more effective.

REFERENCES

[1] Kumar and T. J. Lim, "EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques," *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, 2019, pp. 289-294, doi: 10.1109/WF-IoT.2019.8767194. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8767194&isnumber=8767167

[2] Y. Meidan *et al*., "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," in *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12-22, Jul.-Sep. 2018, doi: 10.1109/MPRV.2018.03367731. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8490192&isnumber=8490161

[3] Gandotra, Ekta, Divya Bansal, and Sanjeev Sofat. "Malware analysis and classification: A survey." *Journal of Information Security* 2014 (2014). https://www.scirp.org/html/4-7800194_44440.htm
[4] Jamal, A., Hayat, M. F., & Nasir, M. (2022). Malware detection and classification in IoT network using ANN. Mehran University Research Journal Of Engineering & Technology, 41(1), 80–91. https://search.informit.org/doi/10.3316/informit.263296849285942
[5] NGiang L. Nguyen, Braulio Dumba, Quoc-Dung Ngo, Hai-Viet Le, Tu N. Nguyen, A collaborative approach to early detection of IoT Botnet,Computers & Electrical Engineering,Volume 97, 2022, 107525,ISSN 0045-7906 https://www.sciencedirect.com/science/article/abs/pii/S0045790621004717
[6] Sebastián García, Vojtěch Uhlíř, and Martin Rehak. 2014. Identifying and modeling botnet C&C behaviors. In Proceedings of the 1st International Workshop on Agents and CyberSecurity (ACySE '14). Association for Computing Machinery, New York, NY, USA, Article 1, 1–8. https://doi.org/10.1145/2602945.2602949
[7] Samantray, Om & Tripathy, Satya. (2021). An Opcode-Based Malware Detection Model Using Supervised Learning Algorithms. International Journal of Information Security and Privacy. 15. 10.4018/IJISP.2021100102.. https://www.sciencedirect.com/science/article/pii/S016740482300295X
[8] O. P. Samantray and S. Narayan Tripathy, "A Knowledge-Domain Analyser for Malware Classification," *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, Gunupur, India, 2020, pp. 1-7, doi: 10.1109/ICCSEA49143.2020.9132916. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9132916&isnumber=9132837
[9] M. Dib, S. Torabi, E. Bou-Harb and C. Assi, "A Multi-Dimensional Deep Learning Framework for IoT Malware Classification and Family Attribution," in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1165-1177, June 2021, doi: 10.1109/TNSM.2021.3075315. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9411822&isnumber=9450206
[10] B. Sharma, R. Kumar, A. Kumar, M. Chhabra and S. Chaturvedi, "A Systematic Review of IoT Malware Detection using Machine Learning," 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2023, pp. 91-96. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10112581&isnumber=10111838
[11] A. A. Ali, O. Al-Blooshi, R. A. Ali and H. A. Hamadi, "Securing the Internet of Things (IoT) Application: Best Practices and Challenges in IoT Software," 2024 Advances in Science and Engineering Technology International Conferences (ASET), Abu Dhabi, United Arab Emirates, 2024, pp. 1-7, doi: 10.1109/ASET60340.2024.10708648. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10708648&isnumber=10708636
[12] M. A. Khatun, S. F. Memon, C. Eising and L. L. Dhirani, "Machine Learning for Healthcare-IoT Security: A Review and Risk Mitigation," in IEEE Access, vol. 11, pp. 145869-145896, 2023, doi: 10.1109/ACCESS.2023.3346320. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10371310&isnumber=10005208
[13] Nakahara, M., Okui, N., Kobayashi, Y., Miyake, Y., & Kubota, A. (2022). Malware detection for IoT devices using hybrid system of whitelist and machine learning based on lightweight flow data. Enterprise Information Systems, 17(9). https://doi.org/10.1080/17517575.2022.2142854
[14] Basharat Ahmad, Zhaoliang Wu, Yongfeng Huang, Sadaqat Ur Rehman, Enhancing the security in IoT and IIoT networks: An intrusion detection scheme leveraging deep transfer learning, Knowledge-Based Systems, Volume 305, 2024, 112614, ISSN 0950-7051. https://www.sciencedirect.com/science/article/pii/S0950705124012486
[15] Z. Li, B. Perez, S. A. Khan, B. Feldhaus and D. Zhao, "A New Design of Smart Plug for Real-time IoT Malware Detection," 2021 IEEE Microelectronics Design & Test Symposium (MDTS), Albany, NY, USA, 2021, pp. 1-6, doi: 10.1109/MDTS52103.2021.9476113. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9476113&isnumber=9476084
[16] A. D. B, M. K. S and P. Joshi, "Advancing IoT Security: A Stacked Hybrid AI Approach for Anomaly Detection," 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2024, pp. 1-6, doi: 10.1109/CONECCT62155.2024.10677130. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10677130&isnumber=10677018
[17] D. Nanthiya, P. Keerthika, S. B. Gopal, S. B. Kayalvizhi, T. Raja and R. S. Priya, "SVM Based DDoS Attack Detection in IoT Using Iot-23 Botnet Dataset," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2021, pp. 1-7, doi: 10.1109/i-PACT52855.2021.9696569. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9696569&isnumber=9696444
[18] Odeh, Ammar, and Anas Abu Taleb. 2023. "Ensemble-Based Deep Learning Models for Enhancing IoT Intrusion Detection" Applied Sciences 13, no. 21: 11985. https://doi.org/10.3390/app132111985 https://www.mdpi.com/2076-3417/13/21/11985
[19] A. S. Ahanger, S. M. Khan and F. S. Masoodi, "Intrusion Detection System for IoT Environment using Ensemble Approaches," 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2023, pp. 935-938. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10112382&isnumber=10111838



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY