iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70207

Securing IoT Devices with Advanced Cyber Defense Using Random Forest and Django

Dr. R. Bharathi[1], Sudhan S[2], Mohammed Riswan P M[3], Sivesh Kumaar B[4]
Department of Computer Science and Engineering with Specialization in Cyber Security, SRM Institute of Science and technology, Ramapuram , Chennai, Tamil Nadu

Abstract: Combining machine learning with the Django framework significantly enhances intrusion detection in Internet of Things (IoT) environments. This system incorporates powerful classification models Random Forest, Bagging, and Ridge to improve detection precision and resilience against cyberattacks. Random Forest utilizes multiple decision trees to accurately identify diverse and complex attack patterns across large datasets. Bagging enhances the model's robustness by lowering variance through model aggregation, ensuring reliable performance in different intrusion scenarios. Ridge Classifier adds regularization to minimize overfitting, which is especially valuable when handling high-dimensional network data. Django serves as the backbone of the application, offering a user-friendly and scalable interface for real-time intrusion monitoring and response. The synergy between Django and these machine learning models creates a responsive, efficient solution for dynamic IoT security needs. This architecture provides a well-rounded defense mechanism capable of adapting to evolving threats, ensuring comprehensive protection for interconnected IoT systems.
Keywords: IoT Cybersecurity, Machine Learning, Cyberattacks, Anomaly Detection, Threat Prediction, Network Defense, Vulnerability Exploitation, Adversarial Machine Learning.

I. INTRODUCTION

In today's hyper-connected world, the Internet of Things (IoT) has revolutionized how devices communicate and operate. However, this connectivity comes with increased exposure to cyber threats. As more smart devices are added to networks, the risk of sophisticated attacks such as DDoS, botnets, and unauthorized data access grows significantly. To counter these evolving threats, there is a need for intelligent, real-time security systems that can keep up with the complexity of IoT networks. This project proposes a Machine Learning-Django-based framework designed to detect and respond to such threats effectively.

Machine learning offers the advantage of recognizing patterns and anomalies in large volumes of network data, making it ideal for identifying suspicious activities before they escalate. When coupled with Django a powerful and flexible web development framework the system gains a practical, user-accessible front end that allows for real-time monitoring and control. The integrated approach supports the classification of various attack types and enables immediate response through automation.

The framework's real-time data evaluation capabilities help in identifying potential threats as they occur, rather than relying on post-incident analysis. Django's scalability ensures that the system remains responsive even as the network expands. The interface is designed for ease of use, providing administrators with clear visualizations and actionable insights. This combined architecture addresses the rising demand for robust, intelligent security systems in IoT environments and lays the foundation for scalable, proactive intrusion detection that evolves alongside modern cyber threats.

A. Real-World Applications

A crucial real-world application of IoT cybersecurity using machine learning is industrial IoT networks. In industries such as manufacturing and energy, IoT-enabled devices monitor and control operations remotely, increasing efficiency but also exposing networks to cyber threats. Machine learning enhances security by detecting anomalies in device communication and network traffic. AI-driven threat prediction analyzes patterns to prevent cyberattacks, ensuring uninterrupted operations and safeguarding sensitive data. This approach strengthens industrial networks against evolving security risks.

Imagine a smart factory where IoT sensors regulate machinery operations. If a cyberattack attempts to manipulate critical processes, machine learning- powered security systems detect unusual commands and isolate compromised devices before damage occurs. Automated alerts enable swift response, preventing disruptions and securing the industrial IoT ecosystem. This proactive security framework ensures real-time protection and resilience in connected environments.

B. Data Science

Data science is a multidisciplinary domain that brings together scientific techniques, workflows, algorithms, and systems to uncover meaningful insights from both structured and unstructured data. It integrates mathematics, statistics, programming, and domain expertise to uncover patterns and make informed decisions across diverse applications. Peter Naur first introduced the term 'data science' in 1974, suggesting it as a substitute for the term 'computer science', gaining traction in the 1990s when data science gained further recognition when the International Federation of Classification Societies highlighted it as a key topic. The term became widely popular in 2008, thanks to D.J. Patil and Jeff Hammerbacher, who were instrumental in advancing data analytics at LinkedIn and Facebook. Over time, data science evolved into a distinct discipline, merging applied statistics with computer science to address real-world problems and predict outcomes using big data. Its significance lies in enabling organizations to analyze vast datasets, enhance decision-making, and drive innovation across fields including medicine, banking, and digital commerce

C. Artificial Intelligence

Artificial intelligence (AI) is an ever-evolving field focused on enabling machines to mimic human intelligence and behavior, enabling them to learn, problem-solve, and act similarly to humans. Founded as an academic discipline in 1956, AI has evolved through various approaches, including statistical machine learning, which has proven highly successful in solving complex problems. AI encompasses various use cases, spanning from expert systems and language understanding to more advanced solutions, self-driving cars, and strategic game systems. It operates by analyzing large datasets to identify patterns and make predictions, frequently surpassing human capabilities in tasks that require repetition and precision.

AI's importance is in its ability to provide enterprises with new insights and enhance operational efficiency. It can analyze large volumes of data with speed and precision, which makes it especially useful for tasks like reviewing legal documents or optimizing business processes. AI systems function by processing vast amounts of labeled training data, identifying patterns and relationships within it, and using those insights to forecast future outcomes. This approach mirrors cognitive abilities like learning, reasoning, and adapting through self- correction, enabling AI to continually fine-tune its algorithms for more accurate results.

As AI advances, it also raises philosophical and ethical questions about creating intelligent machines. The field draws from multiple disciplines, including computer science, psychology, philosophy, and linguistics, and continues to evolve with technologies like neural networks and deep learning. Despite the hype surrounding AI, its true potential lies in its capacity to augment human capabilities and solve complex problems across industries. AI technologies are likely to continue to have a significant impact on shaping the direction of industries and society, from healthcare and finance to education and transportation. By enhancing decision-making and automating processes, AI can help organizations become more efficient and innovative, ultimately driving growth and progress. As AI continues to evolve, its impact on both business and society will only continue to grow.

D. Machine Learning

Machine learning (ML), a crucial component of artificial intelligence (AI), enables computers to learn from data and enhance their performance over time, eliminating the need for direct programming. The primary goal of ML is to develop algorithms capable of detecting patterns and correlations in large datasets, which can then be used to make predictions or decisions. ML models learn by analyzing historical data, adjusting internal parameters during training, and applying this knowledge to forecast outcomes for new data. Machine learning is typically categorized into three types: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, models are trained on labeled data to predict outcomes accurately. Unsupervised learning focuses on finding hidden structures or patterns in unlabeled data. Reinforcement learning trains an agent to make decisions through interactions with an environment, where feedback in the form of rewards or penalties guides its decision-making process and continuous improvement.

ML algorithms are widely applied across industries for tasks such as fraud detection, recommendation systems, speech recognition, autonomous vehicles, and predictive maintenance. Classification is a fundamental method in supervised learning, where the goal is to assign data into predefined categories based on input feature models that predict discrete outputs (e.g., spam or non-spam emails). Applications such as recognizing speech, identifying individuals through biometrics, and categorizing documents.Python is frequently used for ML implementation due to its extensive libraries and tools. Data scientists leverage ML algorithms to uncover actionable insights by training models on labeled datasets and testing them on new data. This iterative process ensures improved accuracy over time. As ML evolves, it continues to drive innovation across domains like healthcare, finance, and transportation by automating processes and enhancing decision- making capabilities.

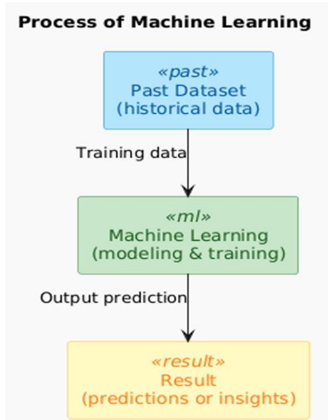

Fig.1 Machine Learning Process

II. RELATED WORK

This study highlights the significant advancements and existing methodologies i**n** IOT networks interconnected with Machine Learning.

1) Siddiqui A.J. and Boukerche A. explore the implementation of adaptive ensembles of autoencoders for unsupervised intrusion detection in IoT ecosystems. Their methodology involves the dynamic integration of multiple autoencoders to align with the shifting patterns of network behavior, thereby improving the accuracy of anomaly detection. Utilizing unsupervised learning, the system effectively identifies new and unknown threats without the need for labeled data, which is often scarce in IoT environments. Their findings reveal that the ensemble approach significantly surpasses single autoencoder models in recognizing complex intrusion patterns.
2) Verma A. and Ranga V. assess the performance of Machine Learning (ML) techniques in identifying Denial of Service (DoS) attacks in IoT networks. Their study evaluates several classification models, including Decision Trees, Random Forest, and Support Vector Machines, using benchmark datasets like CIDDS-001, UNSW-NB15, and NSL-KDD. Emphasis is placed on the critical roles of feature selection and preprocessing in boosting detection precision. To compare classifiers effectively, statistical tools such as Friedman and Nemenyi tests are employed. Although some algorithms show promising results, the study points out ongoing challenges in maintaining consistent performance across various datasets
3) Mahadevappa P. et al. conduct a comparative evaluation of traditional Machine Learning (ML) classifiers for detecting intrusions in edge-centric IoT environments. Using the NSL-KDD dataset, the study assesses the performance of Multi-Layer Perceptron (MLP), Decision Trees, and Support Vector Machines (SVM). Results show that MLP achieves an optimal trade-off between accuracy and training duration, reaching a testing accuracy of 79% with just 1.2 seconds of training time. The authors underline the necessity of selecting ML algorithms that consider both detection effectiveness and computational efficiency, particularly in edge computing scenarios where resources are limited. They encourage further exploration of lightweight algorithms suitable for these conditions.
4) Gueriani A. et al. compile an extensive survey on applying Deep Reinforcement Learning (DRL) methods for intrusion detection in IoT frameworks. The paper categorizes modern DRL-based intrusion detection systems into segments such as Wireless Sensor Networks (WSN), Deep Q-Networks (DQN), healthcare, hybrid approaches, and others. It presents detailed analysis using performance indicators such as accuracy, precision, recall, false positive rate (FPR), false negative rate (FNR), and F-measure. The survey also reviews datasets employed across the literature and discusses current obstacles and future prospects for deploying DRL in IoT security solutions.
5) Al-Hawawreh M. et al. present a thorough review of intrusion detection systems tailored for Internet of Things environments, focusing on detection strategies, deployment models, evaluation methods, types of attacks, public datasets, and prevailing challenges. IDS techniques are categorized into several types, including statistics-based, pattern-based, rule-based, state-based, and heuristic-based approaches. The study draws attention to the expanding attack surface in IoT due to its rapid proliferation and underlines the necessity of strong IDS mechanisms for securing Device communications. Various attack types and detection capabilities are examined, emphasing the importance of efficient and adaptive intrusion detection in securing IoT infrastructures.

6) Soe J.M. et al. undertake a systematic literature review of Machine Learning (ML) applications in IoT security. The review covers diverse ML techniques applied to intrusion detection in IoT, analyzing their effectiveness and constraints. It highlights the prominence of anomaly-based detection for identifying unknown or zero-day threats and stresses the increasing reliance on AI for cyber defense. The study advocates for embedding strong IDS solutions within IoT systems to protect data integrity and network functionality. Challenges discussed include privacy issues, hardware limitations, and the demand for real-time threat identification in constrained environments.
7) R. Bharathi investigates the application of optimization algorithms for analyzing bank loan repayment histories. The study compares Bat Algorithm, Particle Swarm Optimization (PSO), and Grey Wolf Optimization (GWO) in predicting user repayment behaviors. These methods are applied to financial datasets to evaluate their effectiveness in loan prediction tasks. The research outlines each algorithm's unique strengths—Bat Algorithm's adaptability, PSO's speed and efficiency, and GWO's strategic modeling. Findings indicate that optimization techniques enhance financial forecasting and support risk assessment in banking. Despite improved predictive accuracy, challenges such as computational complexity and real-time adaptability remain key limitations in practical implementation.
8) R. Bharathi et al. (2022) conducted a study on sentiment analysis of Amazon unlocked mobile reviews using supervised machine learning techniques. The process involved text preprocessing steps such as handling negations, removing punctuation, stemming, and filtering stop words. TF-IDF vectorization was used to convert the textual data into numerical format. Three models Gaussian Naïve Bayes (GNB), Logistic Regression (LR), and Support Vector Machine (SVM) were applied to classify sentiments as positive, negative, or neutral. Based on experiments using a Kaggle dataset, the SVM model achieved superior performance across all metrics. The study suggests future work on deep learning and clustering methods.
9) Ferrag M.A. et al. carry out an in-depth review of Machine Learning (ML) and Deep Learning (DL) approaches for enhancing security in Internet of Things (IoT) networks. The paper classifies attack types such as DoS, spoofing, and malware injection, and maps these to corresponding ML and DL models like Random Forest, Naive Bayes, and Convolutional Neural Networks (CNNs). The study includes comparative analyses of detection performance across benchmark datasets such as NSL-KDD and CICIDS2017, demonstrating that DL models often outperform their ML counterparts in handling complex attack signatures. Despite these strengths, the authors address limitations such as imbalanced data, scarcity of real-world datasets, and challenges in deploying these models on edge devices. Recommendations include designing adaptive and lightweight models capable of real-time threat detection, contributing to the development of scalable and intelligent IoT defense systems.
10) R. Bharathi et al. (2024) explored deep learning-based sentiment analysis on Amazon Kindle book reviews. The study emphasizes classifying user opinions as positive, negative, or neutral using natural language processing and computational linguistics. Unlike traditional lexicon-based or machine learning methods, deep learning offers better accuracy with minimal feature engineering. The proposed model integrates feature extraction techniques such as N-grams and GloVe embeddings. It employs cascaded recurrent neural networks (CRNN) and convolutional neural networks (CNN) for classification. Experimental results show the Glove-CNN model achieves high accuracy (98.00%), precision (96.95%), recall (96.13%), and F-score (96.52%), outperforming existing sentiment analysis approaches.
11) R. Bharathi et al. (2024) present a study enhancing sentiment analysis of online book reviews through deep learning methods. The research utilizes Convolutional Neural Networks (CNN) and Cascaded Recurrent Neural Networks (CRNN) along with word representation models such as N-grams and Global Vectors (GloVe) to improve sentiment polarity classification. Based on Amazon Kindle review data, the study analyzes how linguistic patterns affect classification performance. GloVe embeddings enhance contextual understanding, while CNN extracts features and CRNN handles sequential data. Experimental outcomes show the model achieves 98% precision, surpassing traditional methods. Despite promising results, challenges remain in preprocessing complexity and adaptability to diverse review styles.

III. PROPOSED WORK

The Intrusion Detection System (IDS) developed for securing IoT networks combines the capabilities of machine learning algorithms Random Forest, Bagging, and Ridge Classifier within a Django-based web platform. This system is designed to enhance real-time detection accuracy and responsiveness by harnessing the distinct advantages of each model. It actively scans ongoing network traffic to identify anomalies and classify threats such as DoS attacks, unauthorized intrusions, and data compromises. Specifically tailored for IoT infrastructure, the IDS offers timely and effective threat identification, ensuring robust and continuous cybersecurity protection.

The Random Forest Classifier, a well-known ensemble method, forms the first layer of defense by constructing multiple decision trees during training and combining their outputs through a majority voting process. This approach not only boosts prediction accuracy but also reduces the likelihood of overfitting, thereby improving performance on unseen data. Its ability to handle large feature spaces and complex relationships makes it ideal for diverse and heterogeneous IoT data. The Bagging Classifier, another ensemble method, contributes to the model by aggregating predictions from several weak learners trained on randomly sampled subsets of the data. This technique reduces the variance in predictions and increases the model's robustness, particularly when faced with noisy or inconsistent input, which is often the case in real-world IoT network traffic. Complementing these models, the Ridge Classifier brings in the advantage of regularization. It addresses challenges like multicollinearity and highly correlated features, which are common in high-dimensional IoT datasets, by penalizing large coefficients. This helps in maintaining generalization performance while preventing the model from becoming overly complex. All these classifiers are tightly integrated into a Django-powered web application, which serves as the user interface for the system. This web-based dashboard allows users to visualize predictions, monitor anomalies, and track threat severity levels in real time. By providing an intuitive and interactive platform, Django supports streamlined decision-making and prompt responses to detected security incidents.

In conclusion, this IDS framework integrates the powerful predictive abilities of ensemble machine learning models with the versatile and scalable Django web platform. It provides a dynamic, real-time security solution, tailored to address the evolving challenges of securing resource-constrained and ever-changing IoT environments.

A. *Existing System*

With the Over the past decade, the rapid adoption of the Internet of Things across numerous sectors has highlighted the critical need for securing these interconnected devices. Many IoT systems manage sensitive data and are often vulnerable to cyber threats due to limited built-in security measures. This research presents an innovative intrusion detection strategy that relies on side-channel analysis specifically monitoring the power consumption patterns of devices to identify abnormal behaviors. Unlike traditional intrusion detection systems, this approach does not interfere with regular device operations. By integrating machine learning techniques, the system can accurately detect unauthorized activities, even those that were not previously encountered during training. The use of machine learning enables the system to learn and recognize deviations in power usage that may signal threats such as data breaches, unauthorized access, or denial-of-service (DoS) attacks. Extensive testing reveals that the system maintains high accuracy in a range of conditions, including real-time environments and custom datasets. One of its primary advantages is its lightweight and portable architecture, which ensures low resource consumption and ease of deployment on constrained IoT devices. Its modular design also allows seamless integration across various IoT infrastructures, making it adaptable to different network configurations. To address varying power limitations and system requirements, multiple deployment models are supported, offering scalability and flexibility based on the target environment. The proposed intrusion detection framework combines the predictive capabilities of ensemble-based machine learning models with the scalability and modularity of the Django web framework. It provides a proactive, intelligent, and efficient solution tailored to the unique challenges posed by dynamic IoT ecosystems. With its minimal computational overhead, user-friendly implementation, and ability to detect both known and novel threats, this system represents a practical and effective tool for enhancing IoT security in real-world scenario and all IoT system requires security.

Moreover, In summary, this IDS framework combines the predictive strength of ensemble machine learning models with the flexibility and scalability of Django. It offers an intelligent, adaptive, and real-time security solution specifically tailored for the dynamic and resource-constrained nature of IoT ecosystems.

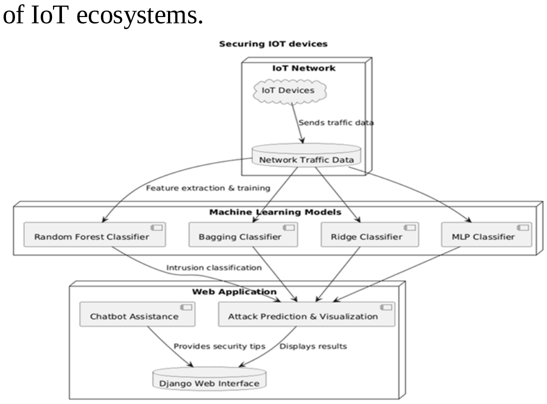

Fig.3 Architecture Diagram of the Proposed System

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

Enhanced Security: Machine learning enables real- time threat detection, preventing cyberattacks, unauthorized access, and data breaches, ensuring robust IoT security.

Scalability and Adaptability: Designed for various IoT environments, ensuring consistent protection for smart homes, healthcare systems, and industrial infrastructures.

IV. MODULES

The Module Description for IOT Cyber Network Attacks using Machine Learning explaining the functionalities are as follows:

1) Module 1: Data Pre-processing

The data preprocessing module plays a critical role in preparing raw IoT network data for machine learning applications. It begins by addressing missing values to prevent incomplete information from affecting model accuracy. Duplicate entries are identified and removed to ensure data integrity, while feature normalization is applied to bring all variables onto a common scale, improving model performance. Additionally, categorical data is transformed into numerical format using encoding techniques, allowing algorithms to effectively interpret and process the information. These combined steps ensure the dataset is clean, consistent, and reliable. Proper preprocessing enhances the accuracy, training efficiency, and overall effectiveness of the machine learning models used for cyber threat detection, making it a foundational component of the intrusion detection system for IoT networks.

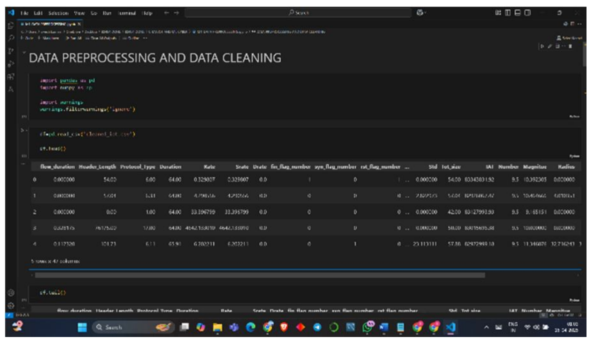

Fig 4.1.1 Data pre-processing and cleaning

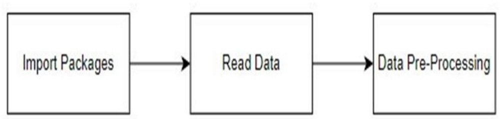

Fig 4.1.2 Module Diagram of Data pre-processing

2) Module 2: Data and Visualization

This module leverages statistical analysis techniques along with visualization tools such as heatmaps, histograms, and correlation plots to explore and interpret IoT network traffic data. By analyzing these visual representations, the system can uncover hidden patterns, behavioral trends, and relationships between various features within the data. Special emphasis is placed on identifying unusual or unexpected activity, which often signals potential security threats or intrusions. These insights enable early detection of anomalies, allowing for a proactive response to cyber threats before they can escalate. Visualization not only simplifies complex data but also enhances situational awareness, making it easier for analysts to understand traffic behavior and detect deviations. By highlighting suspicious patterns, this module significantly contributes to the overall effectiveness of the intrusion detection system, supporting timely and accurate identification of threats within IoT environments. Its role is essential in transforming raw data into actionable intelligence for maintaining network security.

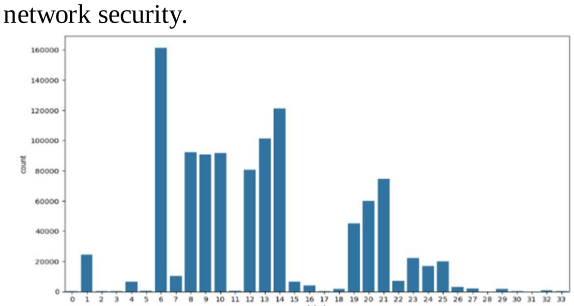

Fig 4.2.1 Bar graph representation of processed data

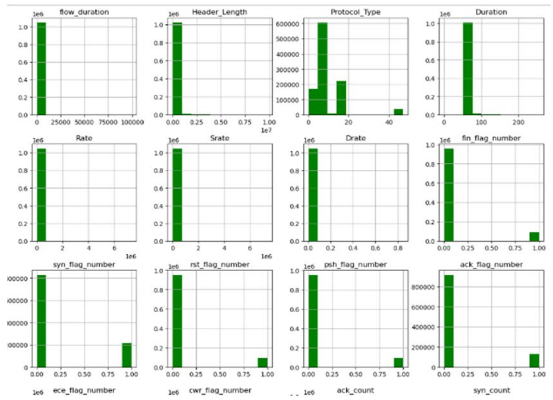

Fig 4.2.2 Graph representation of Input fields

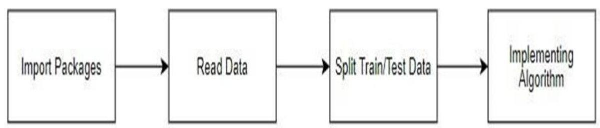

Fig 4.2.3 Module Diagram of Data Analysis and Visualization

3) *Module 3: Ridge Classifier Algorithm*

The Ridge Classifier utilizes Ridge Regression with L2 regularization to reduce the risk of overfitting by applying penalties to large coefficients. This method is particularly effective when working with high-dimensional IoT network data, where many features may be redundant or irrelevant. By limiting the influence of such features, the Ridge Classifier enhances the stability and generalization of the model. This leads to more accurate predictions and improved detection of complex intrusion patterns. Its strength lies in managing noisy or feature-rich datasets, which are common in IoT environments. By maintaining model simplicity without sacrificing performance, the Ridge Classifier becomes a reliable tool for identifying subtle threats within the data.

Overall, it provides a balanced approach to intrusion detection, handling large-scale feature sets efficiently while preserving the model's predictive power in real-world cybersecurity applications.

```
from sklearn.metrics import accuracy_score
a = accuracy_score(y_test,predicted)
print("THE ACCURACY SCORE OF RidgeClassifierCV IS :",a*100)
[19]
THE ACCURACY SCORE OF RidgeClassifierCV IS : 1.5058661762305656

from sklearn.metrics import hamming_loss
hl = hamming_loss(y_test,predicted)
print("THE HAMMING LOSS OF RidgeClassifierCV IS :",hl*100)
[20]
THE HAMMING LOSS OF RidgeClassifierCV IS : 98.49413382376943
```

Fig 4.3.1 Ridge Classifier Accuracy

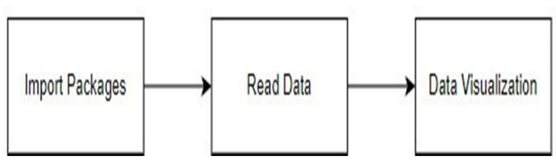

Fig 4.3.2 Confusion Matrix of Ridge Classifier

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 V May 2025- Available at www.ijraset.com

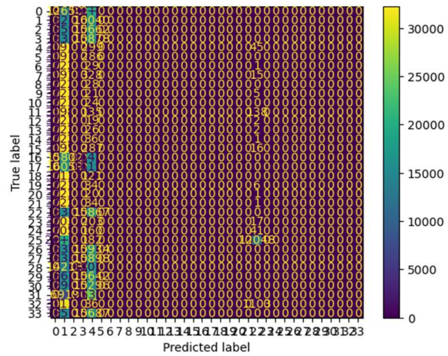

Fig 4.3.3 Module Diagram of Bagging Classifier

4) Module 4: Bagging Classifier Algorithm

Bagging Classifier enhances prediction accuracy by combining multiple weak learners, typically decision trees, into a unified, more powerful model through ensemble learning. It employs bootstrap sampling to create diverse subsets of the training data, allowing each base model to learn from different data points. This technique effectively reduces variance, strengthens generalization, and minimizes the risk of overfitting. In the context of IoT intrusion detection, Bagging proves especially valuable due to its ability to maintain high accuracy across diverse and complex network traffic. Its ensemble nature ensures consistent performance, even in noisy or unpredictable environments, making it a dependable solution for identifying a broad spectrum of security threats in real-time.

```
from sklearn.metrics import accuracy_score
a = accuracy_score(y_test,predicted)
print("THE ACCURACY SCORE OF BAGGING CLASSIFIER IS :",a*100)
```

```
THE ACCURACY SCORE OF BAGGING CLASSIFIER IS : 99.99379964275
```

```
from sklearn.metrics import hamming_loss
hl = hamming_loss(y_test,predicted)
print("THE HAMMING LOSS OF BAGGING CLASSIFIER IS :",hl*100)
```

```
THE HAMMING LOSS OF BAGGING CLASSIFIER IS : 0.006200357249995668
```

Fig 4.4.1 Accuracy Of Bagging Classifier

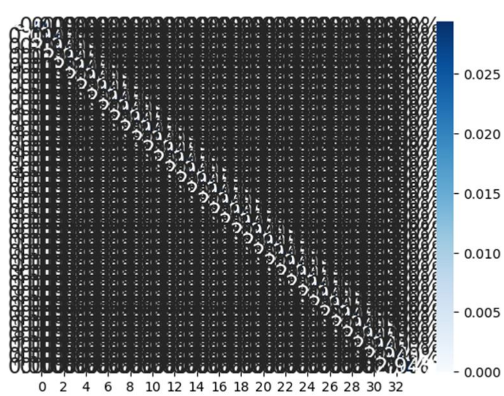

Fig 4.4.2 Confusion Matrix of Bagging Classifier

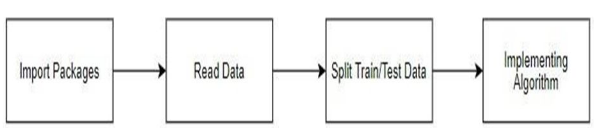

Fig 4.4.3 Module Diagram of Gaussian Naive Bayes

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 V May 2025- Available at www.ijraset.com

5) Module 5: Random Forest Algorithm

Random Forest enhances model performance by building an ensemble of decision trees, making it a robust choice in various machine-learning applications to boost prediction accuracy. It employs "bagging" to create diverse training sets by randomly sampling data, training a decision tree on each set, and combining their predictions through voting or averaging. This ensemble approach effectively reduces overfitting and enhances model generalization by balancing individual tree errors. Random Forest also evaluates feature importance, offering valuable insights into the most influential variables. By combining data and feature sampling, it crafts a resilient model suitable for diverse applications, this makes it highly effective for both classification and regression tasks, as it can capture complex data patterns and highlight the most important features.

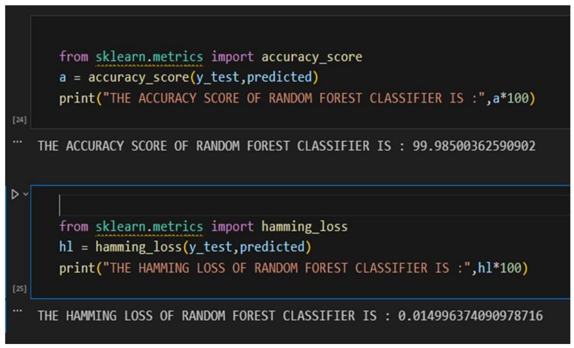

Fig 4.5.1 Accuracy of Random Forest Classifier

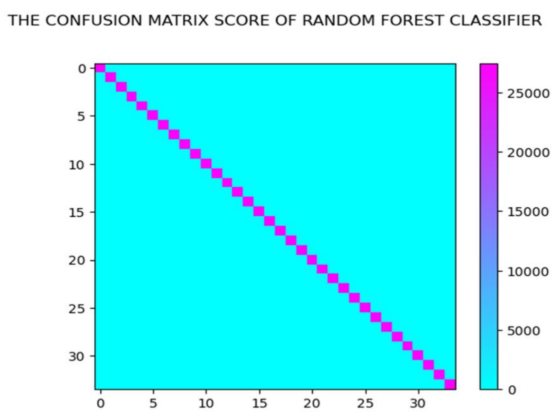

Fig 4.5.2 Confusion Matrix of Random Forest Classifier

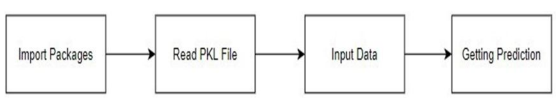

Fig 4.5.3 Module Diagram of Random Forest Algorithm.

Prediction Performance Based on Accuracy:

The algorithms use a linear model to make predictions, with logistic regression commonly employed to ensure high accuracy.

False Positives (FP): These occur when the model incorrectly labels a non-defaulter as a defaulter, resulting in an inaccurate prediction.

False Negatives (FN): This happens when the model fails to recognize a true defaulter, wrongly predicting them as a non-defaulter or misclassifying survival as death.

True Positives (TP): The model correctly identifies a defaulter or accurately predicts survival when both the actual and predicted outcomes align.

True Negatives (TN): This refers to the accurate prediction of non-defaulters, where both the actual and predicted results indicate no default or non-surviv

True Positive Rate (TPR) = TP / (TP + FN) False Positive Rate (FPR) = FP / (FP + TN)

Accuracy: It shows the overall proportion of correct predictions, reflecting how often the model correctly classifies both defaulters and non-defaulters.

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 V May 2025- Available at www.ijraset.com

Accuracy calculation:
Accuracy = (TP + TN) / (TP + TN + FP + FN)
Precision: This metric evaluates the proportion of true positive predictions among all positive predictions made by the model.
Precision = TP / (TP + FP)
Recall: It determines the ability of the model to correctly identify all actual defaulters.
Recall = TP / (TP + FN) General Formula:
F- Measure = 2TP / (2TP + FP + FN)
F1-Score Formula:
F1 Score = 2*(Recall * Precision) / (Recall + Precision)

6) Module 6: Chatbot Data Preprocessing

The chatbot module processes user input by tokenizing and cleaning the text to ensure accurate interpretation. It links various cybersecurity concerns to relevant responses, enabling the system to deliver precise guidance. Designed for real-time interaction, the chatbot provides users with instant security tips and preventive strategies, enhancing engagement and offering accessible, responsive cybersecurity assistance within an intrusion detection framework. The chatbot module enhances cybersecurity support by accurately interpreting user input through text tokenization and cleaning.

It connects specific queries to relevant responses, offering timely advice. With its real-time interaction capability, the chatbot improves user engagement and ensures quick access to preventive measures within the intrusion detection system.

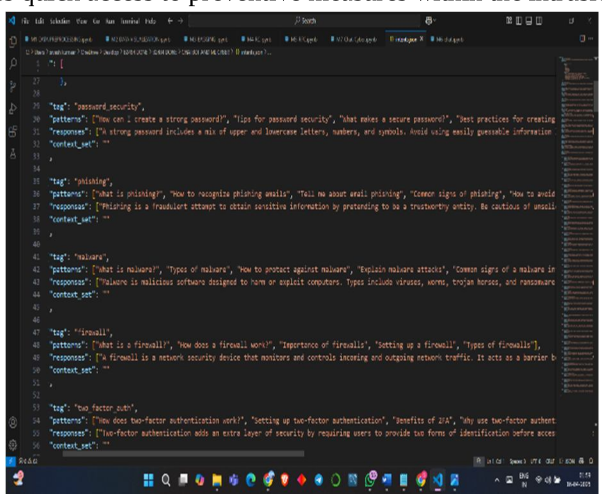

Fig 4.6.1 Chatbot Data Preprocessing

7) Module 7: MLP(Multi-Layer Perceptron)

The system analyzes IoT network data to uncover complex patterns, using a deep learning model to improve intrusion detection. By learning hierarchical representations of potential threats, the model enables more accurate classification of various cyberattacks. This approach enhances detection accuracy and resilience by capturing both low-level and high-level features. The deep learning technique is particularly effective in defending IoT systems against sophisticated infiltration attempts, as it adapts well to evolving and dynamic threats, making it a crucial component for robust cybersecurity in IoT environments.

8) Module 8: Deployment

The system integrates machine learning models into a Django-based web application with an intuitive user interface, enabling real-time intrusion detection. This seamless integration allows users to monitor and respond to risks effectively within IoT networks. Designed for scalability and high performance, the platform efficiently handles large volumes of network data while maintaining both accuracy and speed. As a result, it serves as an effective and practical solution for IoT security monitoring and threat management, ensuring reliable protection against evolving security threats while supporting the growing demands of IoT environments.

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 V May 2025- Available at www.ijraset.com

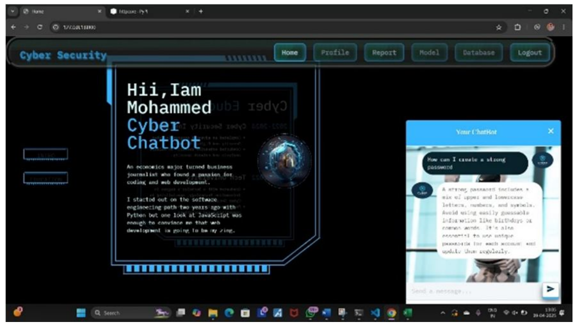

Fig 4.6.2 User Interface

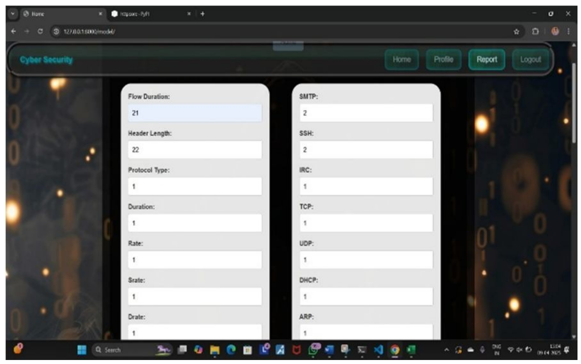

Fig 4.6.3 User Input page

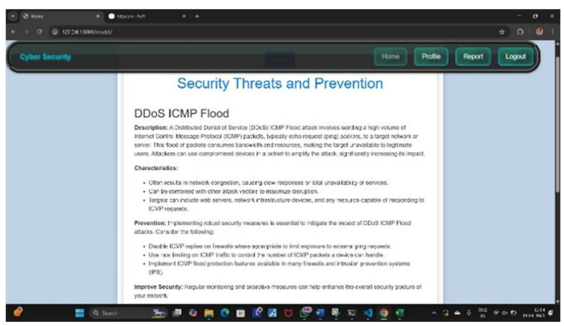

Fig 4.6.4 Output Page

V. RESULTS AND DISCUSSION

The proposed system's performance was evaluated through various tests, with results showing a substantial improvement over the existing system. The existing method, which uses side-channel power consumption analysis, demonstrated limited detection capabilities, especially for covert attacks that do not cause significant power spikes. It also struggled with scalability in dynamic IoT environments, with accuracy decreasing as network complexity increased. In comparison, the proposed system, which integrates Random Forest, Bagging, and Ridge Classifiers within a Django-based framework, significantly improved detection accuracy. The machine learning classifiers achieved high precision and recall, with Random Forest scoring 99% in both. This method not only handled diverse attack types but also provided real-time monitoring through an interactive Django interface, making it a scalable and flexible solution for various IoT environments.

The system's efficiency is further demonstrated by its ability to classify multiple attacks in real-time, providing immediate results to users via the web interface. Additionally, it showed improved scalability and adaptability, accommodating varying IoT network sizes and dynamic attack patterns. By leveraging machine learning's predictive capabilities, the proposed system ensures better detection rates and lower false positives, offering a more reliable solution for securing IoT networks. Overall, the proposed system provides a robust, scalable, and user-friendly solution for IoT security, addressing the limitations of the current power-based detection methods and offering enhanced real-time threat analysis.

A. *Proposed System Vs Existing System*

The current intrusion detection system for IoT uses side-channel analysis, primarily monitoring device power consumption to detect unusual activity. While lightweight and non-intrusive, this method is limited to identifying attacks that cause significant power fluctuations, making it ineffective against stealthy or subtle threats. Moreover, it lacks scalability and may struggle in complex network environments. In contrast, the proposed system employs machine learning classifiers—Random Forest, Bagging, and Ridge—within a Django web framework. It analyzes real-time network traffic to detect intrusions such as DoS attacks and unauthorized access, improving accuracy and detection range. Django also provides an intuitive interface for real-time monitoring and threat management.

Key Differences Between the Existing and Proposed Systems

Technological Advancements:

The existing system relies on side-channel analysis using power consumption to detect threats, which limits detection to significant physical anomalies. In contrast, the proposed system leverages machine learning integrated with Django to analyze real-time network behavior. This approach enables smarter, data-driven detection of complex attacks, enhancing the technological depth and responsiveness of intrusion detection compared to static or hardware-bound legacy methods.

Performance and Efficiency

While the existing system performs well in simple scenarios, its detection capabilities are limited and may produce false negatives. The proposed system enhances performance using advanced classifiers like Random Forest and Bagging, which efficiently identify threats with higher accuracy. It reduces processing delays and improves response time, making it more reliable and suitable for real-time applications in resource-sensitive IoT environments.

Scalability and Adaptability

The existing model struggles with scalability and becomes less effective in dynamic or large-scale IoT environments. The proposed system addresses this through a modular, machine learning-driven design within the Django framework. It can adapt to various network sizes and changing threat patterns, offering retrainable models that ensure continued effectiveness and broader deployment capabilities across evolving IoT infrastructures.

Usability and Accessibility

The existing system lacks user-friendly interfaces and requires technical knowledge to interpret power consumption data. The proposed solution improves accessibility through a Django-based interface that presents threats visually and intuitively. It supports remote access, real-time updates, and requires minimal training, making it far more usable for both technical and non-technical users to manage IoT security effectively.

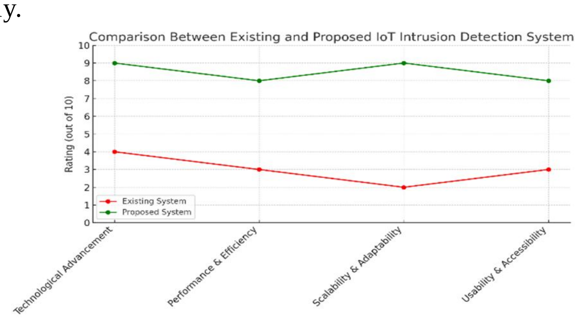

Figure 5.1: Comparison of the proposed system with the existing system

VI. CONCLUSIONS

In conclusion, integrating machine learning techniques into the Django framework presents a robust approach to mitigating security threats within Internet of Things (IoT) environments. Through the deployment of intelligent algorithms, the system is capable of effectively distinguishing between normal and malicious network behavior. Django's role as a high-level web framework ensures the deployment of a scalable and responsive interface that supports real-time threat detection and user interaction.

The incorporation of machine learning models enables accurate classification of various types of cyberattacks, such as Flooding, Time Division Multiple Access (TDMA), Blackhole, and Grayhole attacks. This fusion of machine learning and Django provides a highly responsive and adaptive intrusion detection solution, capable of continuous monitoring, dynamic analysis, and clear data visualization. It also ensures that the system remains effective even when confronted with new or evolving attack vectors. Furthermore, the modular nature of this architecture supports scalability and customization, making it suitable for a variety of IoT network configurations. The proposed system significantly enhances security by offering an intelligent, real-time defense mechanism that adapts to changing threat landscapes. It not only promotes early detection but also facilitates efficient response to potential intrusions, thereby strengthening the overall security posture of IoT infrastructures.

REFERENCES

[1] R. Bharathi, R. Bhavani, & R. Priya. “Leveraging Deep Learning with Sentiment Analysis for Online Book Reviews Polarity Classification Model”, Multimedia Tools and Applications, 17 October 2024, pp 1-20.

[2] H. Al-Alami, A. Hadi, and H. Al-Bahadili, “Vulnerability scanning of IoT devices in Jordan using Shodan,” in Proc. 2nd Int. Conf. Appl. Inf. Technol. Developing Renew. Energy Processes Syst. (IT-DREPS), Dec. 2017

[3] X. Ma, J. Qu, J. Li, J. C. S. Lui, Z. Li, and X. Guan, “Pinpointing hidden IoT devices via spatial-temporal traffic fingerprinting,” in Proc. IEEE INFOCOM Conf. Comput. Commun., Jul. 2020

[4] Jahanzaib Latif, Chuangbai Xiao, Shanshan Tu, Sadaqat Ur Rehman, Azhar Imran, Anas Bilal T. Dai, and H. Shulman, “SMap: Internet-wide scanning for spoofing,” in Proc. Annu. Comput. Secure. Appl. Conf., Dec. 2021,

[5] M. Hastings, J. Fried, and N. Heninger, “Weak keys remain widespread in network devices,” in Proc. Internet Meas. Conf., Nov. 2016

[6] Z. Durumeric, “Fast internet-wide scanning: A new security perspective,” Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. Michigan, Ann Arbor, MI, USA, 2017.

[7] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, “IoT SENTINEL: Automated device-type identification for security enforcement in IoT,” in Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jun. 2017

[8] R. Bharathi, R. Bhavani, and R. Priya, “Twitter text sentiment analysis of Amazon unlocked mobile reviews using supervised learning techniques”, Indian J. Comput. Sci. Eng., vol. 13, no. 4, pp. 1242-1251, 2022. [Online].

[9] F. Murtagh and P. Contreras, “Algorithms for hierarchical clustering: An overview,” WIREs Data Mining Knowl. Discovery, vol. 2, no. 1, pp. 86–97, Jan. 2012.

[10] Abomhara, Mohamed, and G. M. Kien. "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks." Journal of Cyber Security 4 (2015)

[11] Rowe, Dale C., Barry M. Lunt, and Joseph J. Ekstrom. "The role of cyber-security in information technology education." Proceedings of the 2011 conference on Information technology education.ACM, 2011.

[12] “Internet Security Threat Report Internet Report “VOLUME 21, APRIL 2016https://www.symantec.com/content/dam/s ymantec/docs/reports/istr-21-2016

[13] Detection and Prevention of Passive Attacks in Network Security” ISSN: 2319-5967 ISO 9001:2008 Certified International Journal of Engineering Science and Innovative Technology (IJESIT)

[14] Al-Mohannadi, Hamad, et al. "Cyber-Attack Modeling Analysis Techniques: An Overview." Future Internet of Things and Cloud Workshops (FiCloudW), IEEE International Conference on. IEEE, 2016.

[15] R.Bharathi, "Study of Comparison between Bat Algorithm, Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO) for user's bank loan and their related due history," International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), vol. 3, issue 5, pp. 1168-1176, May-June 2018.

[16] Razzaq, Abdul, et al. "Cyber security: Threats, reasons, challenges, methodologies and state of the art solutions for industrial applications. “Autonomous Decentralized Systems (ISADS), 2013 IEEE Eleventh International Symposium on. IEEE, 2013.

[17] “Cyber security: risks, vulnerabilities, and countermeasures to prevent social engineering attacks” International Journal of Advanced Computer Research, Vol 6(23).

[18] R. Bharathi, R. Bhavani, and R. Priya, “Leveraging deep learning with sentiment analysis for Online Book reviews polarity classification model, Multimed. Tools Appl.”, 2024

[19] Ten, Chee-Wooi, Chen-Ching Liu, and Govindarasu Manimaran. "Vulnerability assessment of cyber security for SCADA systems." IEEE Transactions on Power Systems 23.4 (2008).

[20] “Cyber Crime-Its Types, Analysis, and Prevention Techniques”, Volume 6, Issue 5, May 2016 ISSN: 2277 128X www.ijarcsse.com