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.2025.70168

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 V May 2025- Available at www.ijraset.com

IoT-Based Smart Home Security System with AI-Powered Intrusion Detection

Kunal Singh Bisht, Shreyansh Dwivedi, Mr. Harendra Singh, Dr. Aanchal Gaba, Dr. Sureshwati
[1, 2]*Department of Computer application, Greater Noida Institute of Technology (Engg. Institute), Greater Noida, India*
[3]*Deputy Head of Department, Greater Noida Institute of Technology (Engg. Institute), Greater Noida, India*
[4]*Assistant Professor, Department of computer application, Greater Noida Institute of Technology (Engg. Institute), Greater Noida, India*
[5]*Assistant Professor, Department of Computer Applications, Greater Noida Institute of Technology (Engg. Institute), Greater Noida, India*

Abstract: ***This paper explores the development and implementation of an IoT-based smart home security system enhanced by AI-powered intrusion detection. This system leverages the interconnectedness of IoT devices, such as smart cameras and sensors, to collect real-time data from the home environment. Artificial intelligence algorithms, particularly machine learning and deep learning, analyze this data to identify patterns and anomalies indicative of potential security threats. The system aims to provide homeowners with enhanced security through real-time monitoring, intelligent alerts, and automated responses. We discuss the core concepts, functionalities, challenges, and future trends associated with this technology, emphasizing the importance of addressing privacy and security concerns.***

I. IMPORTANCE

The pervasive integration of Internet of Things (IoT) devices has dramatically reshaped various sectors, including the domestic sphere, where smart home systems have become increasingly prevalent. The rapid advancement of information and communication technologies has led to the proliferation of sensors, hardware, and software applications, enabling the automation and remote control of home environments. While these systems offer unparalleled convenience and efficiency, they simultaneously introduce significant security vulnerabilities. The expanding attack surface of IoT networks, coupled with the resource-constrained nature of many IoT devices, presents a unique challenge for cybersecurity professionals. Malicious actors are increasingly targeting these systems, necessitating the development of robust and adaptive security mechanisms.

Traditional security approaches often prove inadequate in addressing the complexities of IoT environments. Consequently, the application of Artificial Intelligence (AI) and, specifically, machine learning (ML) algorithms has emerged as a promising avenue for enhancing IoT security through the development of intelligent Intrusion Detection Systems (IDS). Numerous studies have explored the efficacy of ML techniques in detecting anomalies and malicious activities within IoT networks. Research efforts, such as those by Hasan et al. [1], Latif et al. [2], and Kumar et al. [3], have demonstrated the potential of various ML models, including Random Forest, Neural Networks, and Support Vector Machines, in building effective IDS. These studies have addressed diverse threats, ranging from Denial of Service (DoS) attacks to sophisticated data probing and malicious control scenarios.

This research aims to contribute to the ongoing efforts in securing IoT-based smart homes by conducting a comprehensive evaluation of different ML algorithms for intrusion detection using the DS2oS dataset. We investigate the impact of feature selection techniques on IDS performance, specifically focusing on reducing feature dimensionality to optimize resource utilization without compromising detection accuracy. Furthermore, we explore methods to achieve high performance with minimal energy consumption, addressing the resource constraints of typical IoT devices. By employing common evaluation criteria and rigorously analyzing the results, we aim to provide valuable insights into the effectiveness of ML-based IDS for smart home security. This paper evaluates the performance of different machine learning models, and aims to determine the models that provide the best performance for IOT based smart home security.

II. BACKGROUND

This section provides the essential information needed to understand our research. First, we'll explain how Intrusion Detection Systems (IDS) work in the context of IoT networks. Then, we'll introduce the DS2oS dataset, a commonly used resource for analyzing IoT security, and discuss its key features. Next, we'll describe the Machine Learning (ML) models we used to build our IDS. We'll also cover the methods we employed to select the most important features from the dataset. Finally, we'll explain the performance metrics we used to measure how well our models performed.

©IJRASET: All Rights are Reserved

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

A. Intrusion Detection Systems for IoT

IoT devices have become a vital part of our daily lives, offering a wide range of services. These devices communicate with each other, simplifying tasks and even making decisions for us. As IoT networks continue to grow, we can expect an increase in the variety and complexity of cyberattacks. Basic security measures, like encryption and authentication, often fall short due to the limited resources of IoT devices. This leads to significant security weaknesses.

Therefore, we need more sophisticated security systems that can effectively detect and prevent attacks. It's crucial that these systems are designed to be lightweight, meaning they don't require a lot of processing power, as IoT devices typically have limited computational resources compared to traditional computers.

In today's world, many security systems use Artificial Intelligence (AI) and Machine Learning (ML) to automatically identify threats. These systems can even make decisions to block attacks. For instance, ML can detect unusual behavior in IoT network traffic, which might indicate an attack. When an anomaly is detected, the system can send alerts, and apps can be programmed to automatically take action to stop the threat. In essence, ML algorithms continuously analyze IoT device communication data to find anything that looks out of place.

B. Overview of the DS2oS Dataset

We used the DS2oS dataset for our research. This dataset was created to help improve the privacy and security of IoT users. It's publicly available on Kaggle. Essentially, it was built by recording how IoT devices in a home normally behave, and then adding in examples of abnormal or attack-like activity.

To create this dataset, researchers monitored sensors and IoT devices in different home environments. They tracked things like light controls, motion sensors, thermostats, and even smartphones. The system they used allowed these devices to share information with each other, and users could control them through a central system, either a website or a mobile app. Every action taken by these devices and users was recorded.

Each device or "node" in the system has a specific address, like /kaName/serviceName/variableName. We know the types of these nodes (e.g., SmartDoors, Batteries, LightController) and their locations (e.g., entrance, kitchen, bathroom). Each connection is described by four key pieces of information: the service ID, the node's address, the operation performed (read or write), and a timestamp.

The dataset includes information from four different types of places: a house, two-room apartment, three-room apartment, and an office. They recorded data for a full day, capturing how things worked in each place. We noticed that the most common type of attack in the dataset was DDoS, while the least common was a "wrong setup" attack.

C. Overview of the DS2oS Dataset

The DS2oS dataset is a valuable resource for IoT security research, designed to help improve the privacy and security of IoT users. It's publicly available on Kaggle. This dataset was created by recording normal and abnormal activity from various IoT devices in a home environment.

Researchers monitored devices like light controls, motion sensors, thermostats, washing machines, and smartphones across different home settings, including houses and apartments. These devices were connected in a way that allowed them to share information, and users could control them remotely. All actions were logged, providing a detailed record of device and user behavior.

Each device in the system has a unique address, and its type and location are known. Each connection is described by four key pieces of information: the service ID, the device's address, the action (read or write), and a timestamp.

The dataset contains a total of 357,952 samples, with 347,935 representing normal activity and 10,017 representing abnormal or attack behavior. The dataset includes various attack types, such as Denial of Service (DoS), network scans, malicious operations, spying, data probing, wrong setups, and malicious control.

- DoS: Attacks that overwhelm IoT devices with traffic to disrupt their services.
- Network Scans: Attacks that probe IoT devices to find vulnerabilities.
- Malicious Control: Attacks that gain unauthorized access to IoT devices.
- Malicious Operation: Attacks that cause IoT devices to perform unexpected actions.
- Spying: Attacks that exploit system weaknesses to access sensitive information.
- Data Probing: Attacks that search for vulnerabilities in IoT devices.
- Wrong Setup: Attacks that exploit incorrect system configurations.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The dataset provides a realistic representation of IoT network traffic and attack patterns, making it useful for developing and testing AI-powered intrusion detection systems for smart home environments.

D. Machine Learning Models and Feature Selection

In this section, we'll explain the Machine Learning (ML) models we used to classify the DS2oS dataset and how we selected the most important features. We used Naive Bayes, J48, Random Forest, Bagging, and K-Star algorithms.

- Naive Bayes: This algorithm uses probability to classify data. It assumes that the features are independent of each other.
- J48: This is a decision tree algorithm that creates "if-then" rules to classify data.
- Random Forest: This algorithm builds multiple decision trees and combines their predictions to improve accuracy and reduce errors.
- Bagging: This technique improves prediction by creating multiple versions of the model, and then combines the results.
- K-Star: An algorithm useful for feature selection, it looks at the k-nearest neighbors to determine feature importance.

E. Feature Selection

We used a technique called Information Gain (IG) to select the most relevant features from the DS2oS dataset. This reduced the number of features from 12 to 6, which helps improve the speed and efficiency of our ML models. The selected features, ranked by importance, were:

- sourceAddress
- accessedNodeAddress
- destinationServiceAddress
- sourceID
- value
- sourceType

F. Performance Evaluation

To evaluate how well our models performed, we used several metrics:

- Accuracy: The percentage of correct predictions.
- Recall: The ability of the model to find all the actual positive cases.
- Precision: The ability of the model to avoid labeling a negative case as positive.
- F1-Measure: A combined measure of precision and recall.

We used a confusion matrix to calculate these metrics. This matrix helps us understand how many predictions were correct (True Positives and True Negatives) and how many were incorrect (False Positives and False Negatives).

III. EXPERIMENTAL EVALUATIONS

In this section, we explain how we tested our machine learning models using the DS2oS dataset. We created two modified datasets from the original: one with the timestamp removed (11 features) and another with only the most important six features, selected using Information Gain.

A. Experimental Setup

We used the Weka software to train and test our models. Our computer setup included a Windows 10 operating system, an i5 processor, 16 GB of RAM, and an NVIDIA graphics card.

We compared the performance of our models using the original 12-feature dataset, the 11-feature dataset (no timestamp), and the 6-feature dataset. We noticed that the original dataset, with the timestamp, led to unrealistically high accuracy because the models basically memorized the data based on the unique timestamps.

We used two testing methods:

- Percentage split: 80% of the data was used for training, and 20% for testing.
- K-fold cross-validation: The data was divided into 10 parts, and each part was used for testing in turn.

We trained and tested Naive Bayes, J48, Random Forest, Bagging and K-Star algorithms.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

B. Performance Comparisons

We compared the performance of the algorithms across the datasets and testing methods. Here's a summary of our findings:
The original 12-feature dataset showed very high accuracy, but this was likely due to overfitting on the timestamp.

- Removing the timestamp (11 features) resulted in more realistic accuracy scores.
- The 6-feature dataset, selected using Information Gain, still provided good accuracy, showing that we could reduce the number of features without losing much performance.
- The Random Forest algorithm had the highest accuracy for the 6-feature dataset.
- The J48 algorithm performed well, but was prone to overfitting when the timestamp was included.
- The performance of Naive Bayes was consistent across different datasets.
- K-fold cross-validation and percentage split methods showed similar results.

In general the 6 feature dataset still provided very high accuracy, which means that we were able to reduce the computational complexity, with out losing a lot of accuracy.

IV. DISCUSSION

We compared our results with other recent studies that used the DS2oS dataset (Table 8). Many of these studies also achieved high accuracy, around 99%, similar to our findings. When we tested our models without removing the timestamp feature, we got results close to 100% accuracy. However, because timestamps are unique, these results were not realistic; the models were essentially memorizing the data. We then used Information Gain to select the six most important features and achieved a 99.30% accuracy with the Random Forest algorithm. This is significant because it shows we can achieve high performance even with a reduced dataset, which is crucial for IoT devices with limited processing power. In summary, our results are consistent with other research, and we demonstrated that effective intrusion detection is possible even with a smaller, more efficient dataset, which is vital for real-world IoT applications.

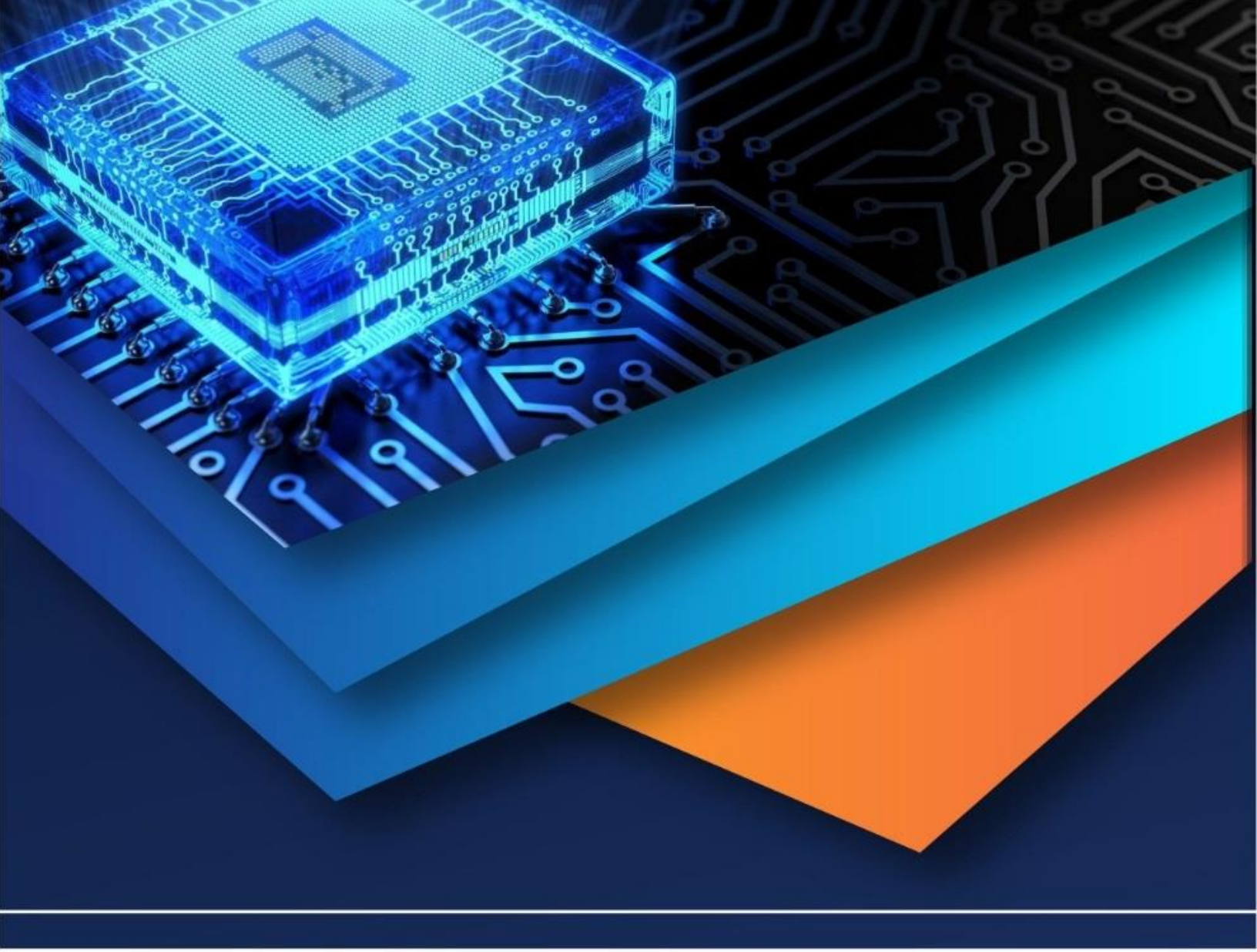
V. CONCLUSIONS

In this research, we compared different machine learning (ML) methods to build an intrusion detection system (IDS) for IoT networks, using the DS2oS dataset. We explored the dataset, explained how IDS works, and described the various attack types it contains. We used a feature selection technique called Information Gain to create different versions of the dataset: one with 12 features, one with 6 important features, and one with 11 features (where we removed the timestamp). We then tested these datasets with various ML algorithms. Our results showed that the Random Forest algorithm achieved the best accuracy, reaching 99.42% on the 6-feature dataset. We also found that including the timestamp as a feature led to unrealistic results, so we excluded it.
The tests on the 6-feature dataset, created using Information Gain, proved that we can achieve good performance with fewer features, which is important for IoT devices with limited resources. Random Forest consistently outperformed other algorithms across all datasets. For future work, we suggest addressing the imbalance in the dataset, as some attack types have very few samples. We believe that using advanced techniques like GAN-based re-samplers could help improve the performance of our models for these less common attack types.

REFERENCES

[1] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. Hashem, “Attack and anomaly detection in iot sensors in iot sites using machine learning approaches,” Internet of Things, vol. 7, p. 100059, 2019.
[2] S. Latif, Z. Zou, Z. Idrees, and J. Ahmad, “A novel attack detection scheme for the industrial internet of things using a lightweight random neural network,” IEEE Access, vol. 8, pp. 89 337–89 350, 2020.
[3] P. Kumar, G. P. Gupta, and R. Tripathi, “A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 10, pp. 9555–9572, 2021.
[4] M.-O. Pahl and F.-X. Aubet, “All eyes on you: Distributed multi-dimensional iot microservice anomaly detection,” in 2018 14th International Conference on Network and Service Management (CNSM). IEEE, 2018, pp. 72–80.
[5] F. Aubet and M. Pahl, “Ds2os traffic traces,” 2018. [Online]. Available: https://www.kaggle.com/datasets/francoisxa/ds2ostraffic
[6] S. Jadhav, H. He, and K. Jenkins, “Information gain directed genetic algorithm wrapper feature selection for credit rating,” Applied Soft Computing, vol. 69, pp. 541–553, 2018.
[7] N. Japkowicz and M. Shah, Evaluating learning algorithms: a classification perspective. Cambridge University Press, 2011.
[8] T. R. Patil, “Mrs. ss sherekar,” performance analysis of j48 and j48 classification algorithm for data classification”,” International Journal of Computer Science And Applications, vol. 6, no. 2, 2013.
[9] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, “An improved method to construct basic probability assignment based on the confusion matrix for classification problem,” Information Sciences, vol. 340, pp. 250–261, 2016.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)