# AI-Driven Intrusion Detection in IoT Networks: Enhancing Security through Machine Learning and Feature Selection Techniques

*Karamvir Kharinta[1], Kumar Harsh[2] and Deepak Paramhans[3]

[1,2,3] Department of Computer Science and Engineering, Chandigarh University, Mohali, India
[1]monukk2002@gmail.com, [2]21bcs11423@cuchd.in,[3]21bcs11386@cuchd.in

**Abstract.** The Internet of Things (IoT) has changed several industries by its rapid spread. It has made automation and seamless communication possible. IoT networks now have serious vulnerabilities as a result of their exponential growth, which makes them easy targets for cyberattacks. The specific issues presented by IoT environments cannot often be adequately addressed by traditional security mechanisms like firewalls and antivirus software. This calls for the development of more advanced security solutions. In this study, an AI-driven intrusion detection system (IDS) that is especially suited for Internet of Things networks is designed and put into operation. The proposed system leverages advanced machine learning algorithms to analyze network traffic and detect anomalies indicative of potential security breaches. To optimize the performance of the IDS, this study employs permutation importance for feature selection, a technique that enhances the interpretability and accuracy of machine learning models by identifying the most relevant features from the dataset. Through a thorough series of tests involving multiple machine learning models, such as Support Vector Machines (SVM), Random Forests, and Neural Networks, the efficacy of the suggested IDS is meticulously assessed. These models are trained and tested on a representative dataset to assess their capability in accurately identifying and mitigating diverse security threats prevalent in IoT networks, such as Distributed Denial of Service (DDoS) attacks, unauthorized access, and malware infiltration. The results of this study demonstrate that the integration of permutation selection in the feature selection process significantly improves the detection accuracy and operational efficiency of the IDS. This research not only contributes to the ongoing efforts in securing IoT infrastructures but also provides valuable insights into the application of machine learning techniques in enhancing the robustness of intrusion detection systems. The results have significant design implications for next-generation intrusion detection systems (IDS) that can protect the quickly growing IoT ecosystem from new cyber threats.

**Keywords:**—IoT Security, Intrusion Detection System (IDS), Machine Learning, Permutation Importance, Feature Selection, Cybersecurity, Anomaly Detection, AI-Driven Security, IoT Networks, Network Traffic Analysis.Introduction


N. Vyas et al. (eds.), *Proceedings of the International Conference on Advancements in Computing Technologies and Artificial Intelligence (COMPUTATIA 2025)*, Advances in Intelligent Systems Research 189,
https://doi.org/10.2991/978-94-6463-700-7_31

# 1 Introduction

A fundamental shift in how devices interact and communicate with one another has been brought on by the Internet of Things (IoT) leading to increased automation, efficiency, and relation to previously unheard-of levels across a variety of industries. By 2025, it is projected that there will be over 75 billion IoT devices globally, reflecting a compound annual growth rate (CAGR) of 26.9% from 2017 to 2025. These devices range from simple sensors and home appliances to complex industrial systems, all of them are interconnected within IoT networks to collect, exchange, and process vast amounts of data. While the benefits of IoT are undeniable, this rapid proliferation has also exposed significant security vulnerabilities, making IoT networks the most accessible tar gets for a slew of cybercrimes. The particular difficulties presented by IoT environments are frequently too complex for conventional security measures like firewalls and signature-based intrusion detection systems (IDS) [1]. They comprise a collection of heterogeneous devices with diverse processing capabilities and real-time data processing requirements, diverse networks with different platforms and protocols, and necessitate seamless interoperability among them. In addition, the decentralized nature of IoT networks also makes monitoring the whole network and securing them a challenging task because of the high number of devices connected. As a result, different cyber threats like Distributed Denial of Service (DDoS), unauthorized access, data breaches, and malware propagation have made IoT networks more and more susceptible.

To mitigate these challenges, the adoption of machine learning (ML) and artificial intelligence (AI) based solutions for enhancing IoT network security has become more common. Because of their ability to analyze network data, detect a signal, and detect anomalies that may indicate malicious activity in real-time, AI-powered intrusion detection systems or intrusion detection systems (IDS) has become a powerful alternative [18]. Unlike traditional IDS, which remain static, AI-driven IDS are dynamic and may learn from data to identify previously unknown vulnerabilities and threats. In accordance with the world, IDS relies on defined settlements and signatures. Feature selection is one of the challenges of the effective design of AI-based IDS for IoT Networks [6]. Intrusion detection does not rely on all of the features present in the IoT datasets, but rather only on a small subset of those features; the result is the so-called curse of dimensionality. In a machine learning model, such irrelevant or redundant features can be rendered infeasible, diminishing the accuracy of the detection and increasing the overhead of computation. To mitigate this tendency, this work incorporates permutation importance (a feature selection technique) into the IDS framework. By measuring how much removing a feature reduces the prediction accuracy [3], we can identify and rank features, and hence, this method can help to highlight the most relevant features. This research aims to develop a robust AI-based intrusion detection algorithm for IoT environments by establishing the suitability of machine learning techniques and feature selection methods. Some tests are performed on various machine learning models to study the effectiveness of the proposed IDS to detect and mitigate security attacks, including Support Vector Machines (SVM), Random Forests, and Neural networks [9]. Moreover, the study

demonstrates the benefits of the importance of permutation usage on IDS frameworks vs. models by comparing their performance independently of feature selection [10].

## 2 Methodological Analysis

The approach for implementing in conjunction and generating the dataset (corpus) required to develop and evaluate the proposed AI-driven Intrusion Detection System (IDS) designed for Internet of Things networks is described in this section. After data gathering, we describe the AI-based analysis used to enhance the IDS's efficiency and accuracy of detection. [8]

### 2.1 Gathering the Corpus

Given the need to showcase the capabilities of the proposed IDS, an extensive and representative dataset that demonstrates the varied and constantly shifting nature of Internet of Things network traffic had to be compiled. The corpus, consisting of network traffic data [2], provides the fundamental component for machine learning model development, testing, and validation and represents the basis of the IDS.

**1. Data Sources**:

•*Public IoT Datasets*: The primary data sources were the UNSW-NB15 and IoT-23 datasets, known for their comprehensive logging of IoT network traffic [15]. UNSW NB15, developed by the Australian Centre for Cyber Security, includes over 2.5 million records with 49 features, covering both normal and malicious traffic. IoT-23, from the Stratosphere Laboratory, focuses on malicious activities, particularly botnet traffic [19] [21].

• *Supplementary Data:* Additional data was gathered from industry-specific IoT environments, including smart home networks, industrial control systems (ICS), and healthcare applications. This data, provided by industry partners, captured a wide range of IoT devices and scenarios, enhancing the IDS's generalizability [14].

**2. Selection Criteria:**

• *Relevance to IoT Security:* The selection of datasets was determined by how pertinent they had been for IoT security, with a particular emphasis on datasets that featured a range of attack techniques, including unauthorized access, Distributed Denial of Service (DDoS), and data exfiltration. Additionally, the datasets selected covered a range of IoT network configurations, including hybrid, decentralized, and centralized architectures [4], which represents the multitude of deployment circumstances found in IoT systems in the real world.

• *Labeled Data Availability*: To facilitate supervised ma chine learning, priority was given to datasets that provided labeled data, where each record was annotated with a corresponding label indicating whether the traffic was benign or malicious. This labeling is crucial for training the IDS to distinguish between normal and suspicious activities. [5]

• *Diversity of Features*: The datasets were selected to ensure a rich variety of features, encompassing packet-level data (e.g., packet size, time-to-live, protocol), flow-level data, and application-level data (e.g., HTTP requests, DNS queries). This diversity of features allows the IDS to capture subtle variations in network traffic that may indicate an intrusion [16].

**3. Preprocessing**:

• *Data Cleaning*: A comprehensive data-cleaning process was applied to the raw datasets. Duplicate entries, in consistent data, and incomplete records were found and eliminated. Additionally, normalizing the numerical features to a common scale was part of this process. Support Vector Machines (SVM) and other algorithms that are sensitive to feature magnitudes will benefit greatly from this.

• *Feature Engineering:* Categorical features, such as protocol types and service names, were transformed into a numerical format using one-hot encoding. This method ensures that the categorical data is properly represented without introducing artificial relationships between different categories. Additionally, new features were engineered based on domain knowledge, such as calculating the average packet size per session or the ratio of inbound to outbound traffic.

• *Data Balancing:* To mitigate the impact of class imbalance, which is common in network traffic datasets where benign traffic often far outweighs malicious traffic, techniques such as Synthetic Minority Over-sampling Technique (SMOTE) were applied. This ensures that the machine learning models do not become biased towards the majority class and maintain high sensitivity to minority classes representing attacks.

• *Dimensionality Reduction:* The most informative aspects of the dataset were preserved while its dimensionality was decreased through the use of Principal Component Analysis (PCA). By getting rid of unnecessary and duplicated characteristics, this phase lowers computational complexity and improves IDS performance.

### 2.2. Applying Machine Learning Technique

The development of the AI-driven Intrusion Detection System (IDS) required the deployment of machine learning techniques once the dataset had been prepared through the previously described preprocessing processes. The approaches utilized, such as the feature selection procedure, model training, and model evaluation, are described in detail in this section. Figure 1 represents a model of IoT network traffic data collection and preprocessing workflow.

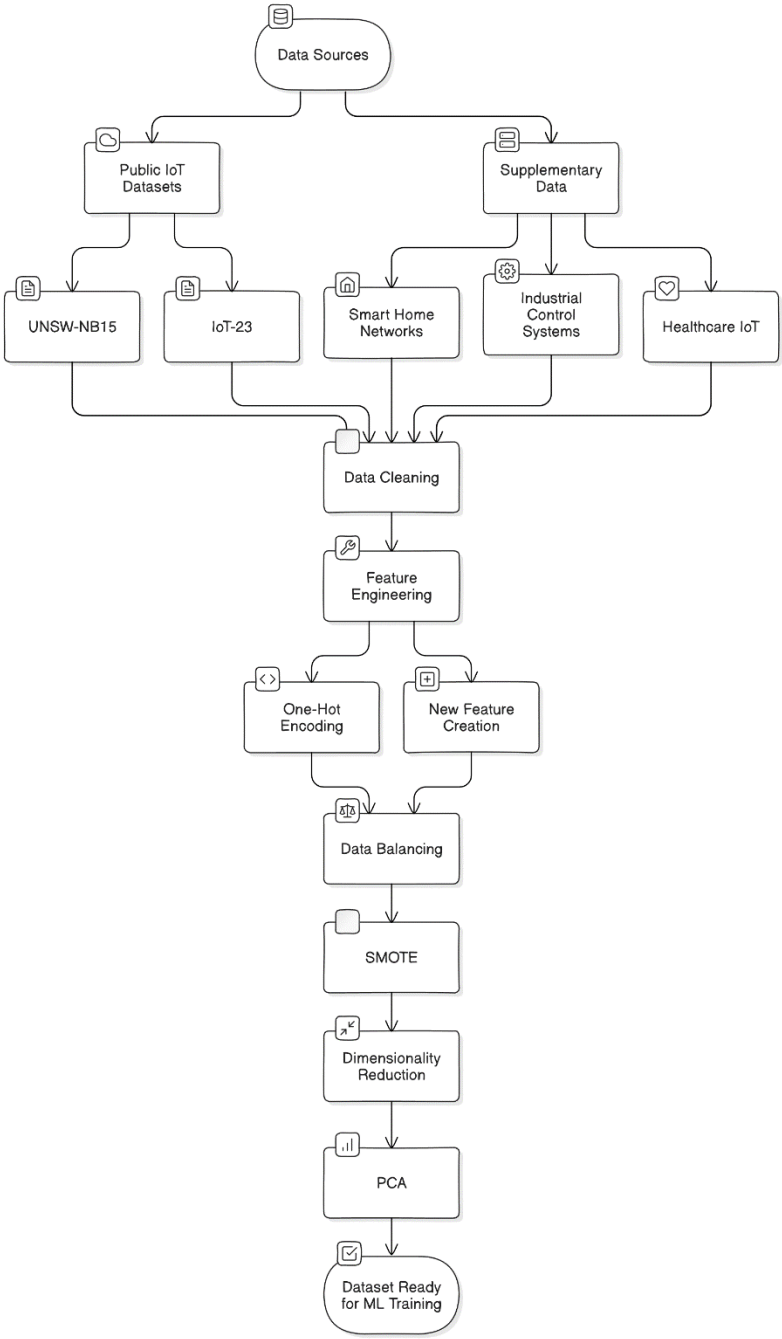


**Figure 1. IoT Network Traffic Data Collection and Preprocessing Workflow**

**1) Feature Selection Using Permutation Importance:**

As a feature selection method, we used permutation importance to improve the IDS's effectiveness. This method was used since it can measure the change in the accuracy of the model when the values of each feature are shuffled randomly, thus offering insights into the relevance for specific features.

• *Implementation:* We implemented permutation importance using the sci-kit-learn library in Python. The method was applied after the initial model training phase to identify the most prominent features influencing the detection rate. The dataset, now composed of normalized and engineered features, was passed through the machine learning model (e.g., Random Forest) to calculate the baseline accuracy. Following this, the efficacy of each feature was determined by shuffling its values and observing the corresponding change in model accuracy. [17]

• *Outcome:* The features with the highest permutation importance scores were retained for the final model training. These features included critical indicators such as packet size, protocol type, and the ratio of inbound to outbound traffic. Less significant features were either discarded or combined with other features to reduce dimensionality and improve model efficiency.

**2) Selection of Machine Learning Models:**

The IDS was developed using several machine learning models to evaluate and compare their performance in detecting intrusions in IoT networks [7]. The selected models summary is given in Table 1.

*Support Vector Machines (SVM):* SVM was selected given how effectively it fares in binary classification tasks and how well it handles high-dimensional data. Radial Basis Function (RBF) kernels are ideal for training SVMs because they can handle the non-linear characteristics of network traffic data.

• *Random Forest:* Selected due to its strong resistance to overfitting and capacity to manage huge datasets with plenty of characteristics. Because Random Forest is an ensemble model and improves the stability of significance scores, it was also utilized as the main model for applying permutation importance.

• *Neural Networks:* It was utilized to extract non-linear correlations and intricate patterns from the data. The rich feature set generated by the feature selection procedure was used to train a three-layered multi-layer perceptron (MLP).

**Table 1.** Summary of Machine Learning Models and Their Configuration

| Model | Configuration Details | Purpose |
|---|---|---|
| Support Vector Machine (SVM) | - Kernel: RBF<br>- Regularization (C): 1.0<br>- Gamma: 'scale' | - Binary classification of network traffic |
| Random Forest | - Number of Trees: 100<br>- Max Depth: 10<br>- Min Samples Split: 2 | - Primary model for feature importance analysis |
| Neural Network (MLP) | - Hidden Layers: 3<br>- Neurons per Layer: 128 | - Capturing complex non-linear |

- Activation Function: ReLU patterns

**3) Training and Evaluation of Machine Learning Models:**

With the dataset prepared and the machine learning models selected, the next phase involved training the models and evaluating their performance. This section describes the process followed for model training, the evaluation metrics used, and the comparative analysis of the model's performance.

**A.Model Training:** The processed dataset was used to train the Support Vector Machine (SVM), Random Forest, and Neural Network (MLP) machine learning models. The training procedure was meticulously crafted to maximize each model's performance while guaranteeing its robustness [20] to enable good generalization to untested data.

• *Training Procedure:*

– *Train-Test Split:*Two primary subsets of the dataset were created: 20% was set aside for testing and the remaining 80% was designated for training. This division made sure the models had enough data to learn from while keeping a sizeable amount for assessment.

– *Cross-Validation:* Five-fold cross-validation was implemented to confirm the models' performance in more detail and reduce the possibility of overfitting. Using this method, the training data is divided into five subsets. The model is trained on four of the subsets, and it is validated on the fifth. This process is repeated until all subsets have been used as validation sets

.– *Hyperparameter Tuning:* For each model, hyper parameters were tuned to find the optimal settings that maximized performance. Grid search was used for this purpose, exploring combinations of hyperparameters such as the regularization parameter (C) in SVM,the prominence of trees in Random Forest, and the learning rate and batch size in Neural Networks.

• *Training Environment:* A high-performance computing environment equipped with several GPUs was used to train the models, especially the neural network, in order to speed up the training process. The machine learning algorithms were implemented using sci-kit-learn, Tensor Flow, and Keras, various other libraries for Python.

**B. Evaluation Metrics:** Computation and assessment of the performance of the trained models were done through several evaluation metrics given in Table 2.

• *Accuracy:* It analyzes the ratio of correctly predicted cases—both true positives and true negatives—to the total number of instances to assess the overall correctness of

the model. However, in incomplete data sets where the majority class predominates, accuracy might not be enough on its own.

• *Precision:* Indicates the proportion of positive identifications (i.e., detected intrusions) that were actually correct. Precision is crucial in intrusion detection, where false positives (incorrectly flagged normal traffic) need to be minimized.

• *Recall (Sensitivity):* Reflects the model's ability to identify all actual positive cases (true positives). High recall is important to ensure that as many actual intrusions as possible are detected, even at the cost of more false positives.

• *F1-Score:* A balanced metric of the model's performance is the harmonic mean of precision and recall. Because it can account for both false positives and false negatives, the F1-Score is particularly useful for circumstances in which the distribution among classes is not consistent.

• *ROC-AUC Score:* The capacity of the model to distinguish across classes is determined by the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). This robust indicator factors into account the trade-off between false positive and true positive rates

**Table 2.** Evaluation Metrics for Model Performance.

| Metric | Description | Importance |
|---|---|---|
| Accuracy | Proportion of total correct predictions (TP + TN) / (Total Predictions) | Overall correctness of the model |
| Precision | Proportion of true positive identifications out of all positive identifications (TP / (TP + FP)) | Minimizing false positives |
| Recall | Proportion of true positives detected out of all actual positives (TP / (TP + FN)) | Ensuring all intrusions are detected |
| F1-Score | Harmonic mean of precision and recall (2 * (Precision * Recall) / (Precision + Recall)) | Balanced measure of precision and recall |
| ROC-AUC Score | Area under the ROC curve, indicating the model's discriminative power | Assessing overall model discrimination |

**C. Comparative Analysis:** After training and evaluating the models, a comparative analysis was conducted to determine which model performed best in detecting intrusions within IoT networks and figure 2 shows scores for selected features using permutation.

• *Performance Results*:
– SVM: Showed strong performance in precision but had a slightly lower recall, indicating that while it was good at minimizing false positives, it may have missed some actual intrusions
.– *Random Forest:* Balanced performance with high precision and recall, making it an effective choice for both identifying intrusions and minimizing false positives.
– *Neural Network (MLP):* Accomplished the best F1-Score and ROC-AUC, proving its exceptional capacity to discern between benign and malicious traffic, and excelled in capturing complex patterns.

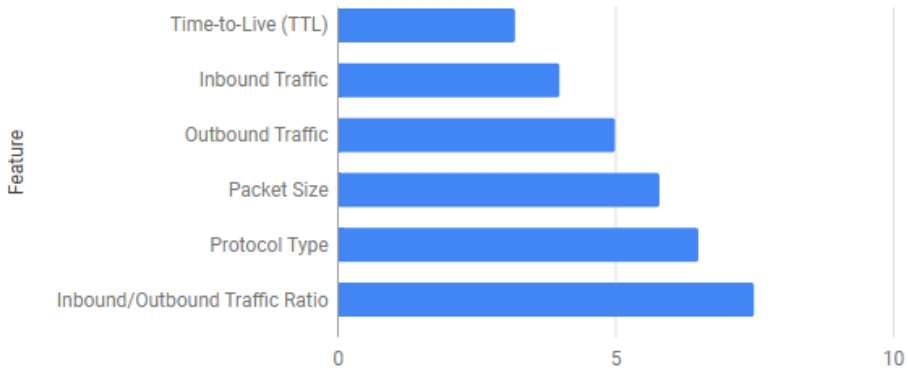


**Figure 2. Permutation Importance Scores for Selected Features**

## 3 Discussion

This section analyzes the positive and negative aspects of each machine-learning model by evaluating the outcomes of the training and evaluation phases. The discussion additionally offers suggestions for potential room for improvement and emphasizes the significance of these results for the creation of AI-driven Intrusion Detection Systems (IDS) in Internet of Things networks. [12]

**3.1) Performance Interpretation:** The comparative analysis of the models reveals several key findings that are critical for understanding their effectiveness in the context of IoT network security.
**• Support Vector Machine (SVM):**
– *Strengths:* This model demonstrates a high level of precision, which is particularly valuable in minimizing false positives. This is important in operational settings where frequent false alarms could lead to wasted resources and potential disregard for alerts.
– *Weaknesses:* The model's lower recall indicates that it may miss some intrusions, potentially allowing certain types of attacks to go undetected. This limitation could be problematic in environments where the detection of all possible threats is crucial [13].
**• Random Forest:**

– *Strengths:* Random Forest provided a balanced performance across all metrics, including precision, recall, and F1-Score. This suggests that it is a reliable model for general-purpose intrusion detection in IoT networks, capable of both accurately identifying intrusions and maintaining a low rate of false positives.
– *Weaknesses:* Even though Random Forest fared well overall, as the number of trees in the forest increases, so do its computational requirements. This might be a problem in Internet of Things setups with limited resources and computing capacity.
**3.2) Implications for IoT Network Security:** The results from this study [11] have several important implications for the design and deployment of IDS in IoT networks:
• *Model Selection:* The unique needs of the deployment environment should direct the model selection for an intrusion detection system (IDS) in Internet of Things networks. For example, SVM might be favored in settings where reducing false positives is crucial. On the other hand, albeit requiring more resources, a Neural Network might be better suitable in situations where identifying all possible dangers is essential.
• *Feature Selection:* The application of permutation importance proved to be an effective strategy for feature selection, enhancing model performance by focusing on the most relevant features. This approach not only improved detection accuracy but also reduced the computational burden by eliminating redundant features.
• *Scalability and Efficiency:* While high-performing models like Neural Networks offer significant advantages in detection capabilities, their scalability and efficiency in resource-constrained environments remain a challenge. Future research should explore lightweight neural network architectures or hybrid models that combine the strengths of different algorithms.

**3.3) Limitations and Future Work:** Regardless of the favorable results, it is essential to acknowledge the following study's limitations:
• *Dataset Diversity:* Although the dataset used in this study was comprehensive, it primarily relied on publicly available data. The inclusion of more diverse datasets, particularly those reflecting emerging IoT technologies and attack vectors, would provide a more robust evaluation of the IDS.
• *Real-Time Implementation:* The models were evaluated in an offline setting. Future work should focus on real time implementation and testing of the IDS in live IoT environments to assess its practical applicability and performance under real-world conditions.
• *Adaptive Learning:* Cyber threats are continually evolving, and static models may struggle to keep pace. Incorporating adaptive learning mechanisms that allow the IDS to assess and change its model data in response to new threats is a crucial area for future research.

## 4 Conclusion

As the increasing number of devices that are connected, expanding and changing at an accelerated rate, it is imperative to create an AI-driven intrusion detection system (IDS) specifically designed for Internet of Things networks. This study set out to

design, implement, and evaluate such a system, lever aging machine learning techniques to enhance its detection accuracy and efficiency. Through rigorous experimentation and analysis, several key insights have been gained, contributing to both the academic field of IoT security and its practical applications.

**4.1) Summary of Findings:** The comparative analysis of the machine learning models—Support Vector Machine (SVM), Random Forest, and Neural Network (MLP)—provided a clear understanding of their respective strengths and limitations:

• *Support Vector Machine (SVM):* SVM showed good precision but poor recall, thus it might not be the best choice for systems that need to detect every intrusion in detail. However, it might be appropriate in situations where reducing false positives is a top concern.

• *Random Forest:* This model offered a balanced performance across all metrics, making it a versatile and reliable option for general-purpose intrusion detection in IoT networks. Its ability to handle a large number of features and its resilience to overfitting make it a strong candidate for IDS deployment.

• *Neural Network (MLP):* The MLP model excelled in detecting complex patterns, reflected in its high F1-Score and ROC-AUC. However, its demand for computational resources suggests that its use may be best suited to more powerful IoT environments or hybrid systems where it can be combined with other models.

**4.2) Contributions to IoT Security:** This research makes several important contributions to the field of IoT security:

• *Enhanced Feature Selection:* By integrating permutation importance into the feature selection process, the study was able to improve the performance of the machine learning models while reducing unnecessary computational overhead. This approach is particularly beneficial in resource-constrained IoT environments.

• *Model Performance Insights:* The detailed analysis of the models' performance provides valuable insights for researchers and practitioners in selecting the appropriate machine-learning techniques for IDS in various IoT scenarios. The findings underscore the importance of considering the specific requirements of the deployment environment when choosing a model.

• *Framework for IDS Development:* The methodology presented in this study offers a comprehensive framework for developing and evaluating AI-driven IDS, which can be adapted and expanded upon in future research. The use of publicly available datasets and established ma chine learning techniques ensures that the findings are reproducible and can be built upon by other researchers.

**4.3) Future Directions:** While this study has achieved its objectives, several avenues for future research have been identified:

• *Real-Time IDS Implementation*: Future work should focus on implementing the IDS in real-time environments, particularly in live IoT networks. This would allow for the evaluation of the system's performance under real world conditions, including its ability to process data at scale and respond to dynamic threats.

• *Adaptive and Continuous Learning:* The incorporation of adaptive learning mechanisms into the IDS would allow it to evolve alongside emerging threats. Techniques such as online learning, where the model updates continuously as new data is received, could significantly enhance the system's resilience against novel attacks.

• *Exploration of Hybrid Models:* Combining the strengths of different machine learning models into a hybrid IDS could offer a more robust solution. For instance, integrating the high precision of SVM with the complex pattern recognition capabilities of Neural Networks might yield a system that balances accuracy with computational efficiency.
• *Broader Dataset Utilization:* Expanding the dataset to include more diverse and representative IoT traffic, especially from emerging technologies and underrepresented regions, would provide a more comprehensive assessment of the IDS's capabilities.

## References

1. A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.
2. Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," in Proceedings of the Network and Distributed System Security (NDSS) Symposium 2018.
3. N. Moustafa and J. Slay, "The UNSW-NB15 Dataset: A Comprehensive Benchmark Dataset for Network Intrusion Detection Systems (NIDSs)," in Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6, 2016.
4. S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, Privacy and Trust in Internet of Things: The Road Ahead," Computer Networks, vol. 76, pp. 146–164, 2015.
5. C. Zhang and M. Zulkernine, "Anomaly-Based Network Intrusion Detection with Unsupervised Outlier Detection," in Proceedings of the 2006 IEEE International Conference on Communications, vol. 5, pp. 2388 2393, 2006.
6. W. Zhou, Y. Zhang, P. Liu, and N. Memon, "The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1606–1616, 2018.
7. L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
8. H. Kaur and A. Singh, "A Survey on Machine Learning Techniques for Intrusion Detection Systems," Journal of Information Security and Applications, vol. 49, p. 102395, 2020.
9. A. A. Diro and N. Chilamkurti, "Distributed Attack Detection Scheme Using Deep Learning Approach for Internet of Things," Future Generation Computer Systems, vol. 82, pp. 761–768, 2018.
10. M. Chowdhury, A. Kayes, and A. Iqbal, "Towards an Intrusion Detection Framework for IoT Networks," in Proceedings of the 2020 International Conference on Advanced Computing and Applications (ACOMP), pp. 1–6, 2020.
11. R. Lu, L. Zhu, X. Liu, and J. Shae, "Achieving Efficient and Secure Data Acquisition for Cloud-Supported Internet of Things in Smart Grid," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1934–1944, 2017.
12. T. T. Le, H. H. Nguyen, and C. W. Ahn, "An Efficient Privacy-Preserving Data Aggregation Scheme for Smart Grid Based on Edge Computing," IEEE Access, vol. 8, pp. 123563–123574, 2020.
13. R. Karam and T. Melodia, "Lightweight Network Intrusion Detection for Military IoT Using Edge Computing," IEEE Communications Magazine, vol. 58, no. 10, pp. 55–61, 2020.

14. M. S. Al-Maadeed, “IoT and Machine Learning for Human Activity Recognition in Smart Home Environments,” IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7633–7645, 2020.
15. S. R. Islam, “Anomaly Detection in IoT Network Traffic for Smart City Applications Using Machine Learning,” IEEE Access, vol. 8, pp. 145778–145789, 2020.
16. T. T. Nguyen, “A Novel Approach for Network Anomaly Detection Based on Feature Selection and Autoencoder,” IEEE Access, vol. 6, pp. 70818–70827, 2018.
17. A. Singh, M. Kumar, and N. Kumar, “An Edge AI-Based Smart Traffic Monitoring System for Internet of Things,” IEEE Sensors Journal, vol. 20, no. 12, pp. 7115–7125, 2020.
18. Y. Yilmaz and S. Celik, “Online Learning-Based Network Anomaly Detection in IoT Using Hyperspherical Representation,” IEEE Transactions on Information Forensics and Security, vol. 16, pp. 4981–4993, 2021.
19. M. Zolanvari, “Explainable Deep Learning for IoT Security,” IEEE Internet of Things Magazine, vol. 4, no. 2, pp. 29–35, 2021.
20. C. Yin, Y. Zhu, J. Fei, and X. He, “A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks,” IEEE Access, vol. 5, pp. 21954–21961, 2017.
21. N. Moustafa and J. Slay, “The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Data Set and the Comparison with the KDD99 Data Set,” Information Security Journal: A Global Perspective, vol. 25, no. 1-3, pp. 18–31, 2016