International Journal of Unified Research & Development
Journal Homepage: https://www.ijurd.com

# Random Forest-Based Classification of IoT Devices Using Network Traffic Analysis for Enhanced Security

Sanjeev Kumar[1,*], Sukhvinder Deora[1]

[1]*Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak, Haryana, India*

*Corresponding Author: skumarcs870@gmail.com

## ABSTRACT

The rapid proliferation of Internet of Things (IoT) devices has introduced significant security challenges, particularly in device identification and access control within heterogeneous network environments. This paper presents a machine learning-based approach for IoT device classification using network traffic analysis. We employ the Random Forest algorithm, a robust ensemble learning method, to classify ten distinct IoT device categories based on their network behavioral patterns. The proposed methodology extracts seventeen discriminative features from network traffic data, including packet size distributions, protocol usage ratios, flow characteristics, and payload entropy metrics. Our experimental evaluation demonstrates that the Random Forest classifier achieves a classification accuracy of 96.6% on the test dataset, outperforming traditional machine learning approaches including Decision Trees, Naive Bayes, K-Nearest Neighbors, and Support Vector Machines. The feature importance analysis reveals that bytes per second, average packet size, and SSL/TLS ratio are the most significant discriminators for device identification. This work contributes to the growing body of research on IoT security by providing an efficient, lightweight, and scalable solution for automated device fingerprinting, enabling network administrators to enforce granular security policies and detect unauthorized device deployments in smart home and industrial IoT environments.

Keywords: Internet of Things; Device Classification; Random Forest; Network Traffic Analysis; Machine Learning; Security; Device Fingerprinting

## 1. INTRODUCTION

The Internet of Things (IoT) has emerged as a transformative paradigm, connecting billions of heterogeneous devices to the Internet for data collection, processing, and autonomous decision-making Atzori et al., 2010; Gubbi et al., 2013. From smart homes and wearable devices to industrial automation and smart cities, IoT ecosystems have permeated virtually every aspect of modern life Stankovic, 2014. According to recent industry reports, the number of connected IoT devices is projected to exceed 30 billion by 2030, generating unprecedented volumes of data and creating complex network topologies that challenge traditional security frameworks Sicari et al., 2015.

The security implications of this massive IoT deployment are profound and multifaceted. Unlike conventional computing devices, IoT endpoints are characterized by resource constraints, diverse communication protocols, proprietary firmware, and often inadequate security implementations Chowdhury et al., 2020. These inherent limitations make IoT networks particularly vulnerable to a wide spectrum of cyber threats, including unauthorized access, data exfiltration, botnet recruitment, and lateral movement attacks Ferrag et al., 2023. A critical prerequisite for securing IoT environments is the ability to accurately identify and classify connected devices, as different device categories exhibit distinct security requirements, communication patterns, and risk profiles Shafiq et al., 2020.

Device classification in IoT networks serves multiple security objectives. First, it enables network administrators to enforce context-aware access control policies, restricting device communication to authorized endpoints and protocols Nguyen et al., 2021. Second, accurate device identification facilitates anomaly detection by establishing baseline behavioral profiles for each device category, thereby enabling the detection of deviations that may indicate compromise or misuse Lopez-Martin et al., 2022. Third, device classification supports inventory management and compliance auditing, ensuring that only approved devices are permitted on the network Hasan et al., 2023.

Traditional device identification methods rely on static attributes such as MAC addresses, IP allocations, or manufacturer information. However, these approaches are increasingly ineffective due to MAC address spoofing, dynamic IP assignment, and the proliferation of white-label devices with identical hardware identifiers Meidan et al., 2017. Consequently, researchers have turned to behavioral fingerprinting techniques that analyze network traffic patterns to infer device types based on their distinctive communication characteristics Bezawada et al., 2018.

Machine learning has emerged as a promising approach for automated IoT device classification, offering the ability to learn complex patterns from network traffic data without relying on explicit signature definitions Alsheikh et al., 2016. Among the various machine learning algorithms, Random Forest has garnered significant attention due to its robustness, interpretability, and ability to handle high-dimensional feature spaces with minimal hyperparameter tuning Breiman, 2001. The ensemble nature of Random Forest, which aggregates predictions from multiple decision trees, provides inherent resistance to overfitting and improved generalization performance compared to single-tree classifiers Apruzzese et al., 2021.

In this paper, we present a comprehensive study on IoT device classification using the Random Forest algorithm. Our contributions are threefold: (1) we design a feature extraction framework that captures seventeen discriminative network traffic characteristics for device fingerprinting; (2) we evaluate the classification performance of Random Forest against five baseline machine learning algorithms on a dataset comprising ten distinct IoT device categories; and (3) we conduct feature importance analysis to identify the most informative traffic attributes for device discrimination. The experimental results demonstrate that our approach achieves superior classification accuracy while maintaining computational efficiency suitable for deployment on resource-constrained IoT gateways.

The remainder of this paper is organized as follows. Section 2 reviews the existing literature on IoT device classification and security. Section 3 describes the proposed methodology, including feature extraction and Random Forest model architecture. Section 4 presents the experimental setup, dataset characteristics, and evaluation metrics. Section 5 discusses the experimental results and comparative analysis. Section 6 provides insights into the practical implications and limitations of the proposed approach. Finally, Section 7 concludes the paper with future research directions.

## 2. RELATED WORK

The identification and classification of IoT devices based on network traffic analysis has been an active area of research in recent years. Several approaches have been proposed, ranging from deep packet inspection to flow-based statistical analysis and machine learning-based behavioral fingerprinting.

Miettinen et al. Miettinen et al., 2017 proposed IoT Sentinel, a system for automated device-type identification using network traffic characteristics. Their approach extracted features from the initial communication sequences of IoT devices and applied a classifier to identify device types. The system demonstrated the feasibility of passive device fingerprinting without requiring active probing or device cooperation. However, their feature set was limited to initial handshake patterns, which may not generalize to devices with configurable communication behaviors.

Meidan et al. Medan et al., 2017 introduced ProfiloT, a network-based IoT device fingerprinting technique that analyzes network traffic to identify device types. Their methodology focused on periodic communication patterns and payload characteristics, achieving promising results on a limited set of consumer IoT devices. The study highlighted the importance of temporal traffic features in distinguishing devices with similar protocol usage but different communication frequencies.

Shahid et al. Shahid et al., 2021 presented a comprehensive framework for IoT device recognition through network traffic analysis. Their approach combined statistical flow features with machine learning classifiers to identify devices in enterprise networks. The study emphasized the challenges of device classification in mixed environments where IoT and non-IoT devices coexist, proposing feature selection techniques to improve classification robustness.

Bezawada et al. Bezawada et al., 2018 investigated behavioral fingerprinting of IoT devices by analyzing network traffic at multiple granularities. Their work demonstrated that devices from the same manufacturer often exhibit similar traffic patterns, enabling manufacturer-level classification as a preliminary step to device-type identification. The study also explored the impact of network encryption on fingerprinting accuracy, noting that TLS/SSL traffic still leaks sufficient metadata for effective classification.

Ortiz et al. Ortiz et al., 2019 developed an IoT endpoint system for dynamic device classification and data protection. Their approach integrated device classification with policy enforcement, automatically applying security controls based on identified device types. The system addressed the practical challenge of real-time classification in high-throughput networks, proposing lightweight feature extraction techniques suitable for edge deployment.

In the domain of machine learning for IoT security, several studies have compared the performance of different algorithms for device classification and intrusion detection. Liu and Lang Liu and Lang, 2020 conducted a comprehensive survey of machine learning and deep learning methods for intrusion detection, identifying Random Forest as one of the most effective algorithms for network traffic classification due to its ability to handle imbalanced datasets and high-dimensional feature spaces. Thakkar and Lohiya Thakkar and Lohiya, 2020 reviewed the application of machine learning and deep learning for IoT security, highlighting the trade-offs between model complexity, accuracy, and computational requirements.

Ferrag et al. Ferrag et al., 2022 performed a comparative study of deep learning approaches for cyber security intrusion detection, noting that while deep neural networks achieve state-of-the-art performance on large datasets, traditional machine learning methods such as Random Forest remain competitive for moderate-sized datasets and offer advantages in terms of interpretability and training efficiency. This finding is particularly relevant for IoT environments where training data may be limited and model explainability is essential for security auditing.

Zhang et al. Zhang et al., 2021 applied machine learning techniques for network traffic classification in IoT environments, demonstrating that ensemble methods consistently outperform single classifiers. Their work emphasized the

 DOI: 10.5281/ijurd.v2i4.96

importance of feature engineering, showing that carefully selected traffic attributes can significantly improve classification accuracy while reducing model complexity.

Despite these advances, several challenges remain in IoT device classification. First, the diversity of IoT devices and protocols continues to expand, requiring classification frameworks that can adapt to new device types without extensive retraining Chowdhury et al., 2020. Second, the increasing adoption of encryption protocols such as TLS 1.3 limits the availability of payload-based features, necessitating reliance on metadata and statistical flow characteristics Shahid et al., 2021. Third, the resource constraints of IoT gateways and edge devices impose strict requirements on classification latency and memory footprint, favoring lightweight models over computationally intensive deep learning approaches Nguyen et al., 2021.

Our work addresses these challenges by proposing a Random Forest-based classification framework that leverages flow-level statistical features, operates effectively on encrypted traffic metadata, and maintains computational efficiency suitable for edge deployment. The comprehensive feature set and comparative evaluation against multiple baseline algorithms provide practical insights for network security practitioners.

## 3. METHODOLOGY

This section presents the proposed methodology for IoT device classification using network traffic analysis. The framework comprises three main components: data collection and preprocessing, feature extraction, and Random Forest classification.

### *3.1 System Architecture*

The proposed system architecture consists of four functional layers: (1) the Network Traffic Capture Layer, which monitors and records packet-level communications within the IoT network; (2) the Flow Aggregation Layer, which groups packets into bidirectional flows and computes statistical descriptors; (3) the Feature Extraction Layer, which transforms raw flow statistics into discriminative feature vectors; and (4) the Classification Layer, which applies the trained Random Forest model to predict device categories.

The traffic capture component operates in a passive monitoring mode, ensuring zero disruption to normal device operations. Flow records are generated using a timeout-based aggregation strategy, where packets sharing the same five-tuple (source IP, destination IP, source port, destination port, protocol) within a specified temporal window are grouped into a single flow record. This approach balances the granularity of traffic representation with the computational overhead of flow tracking.

 DOI: 10.5281/ijurd.v2i4.96

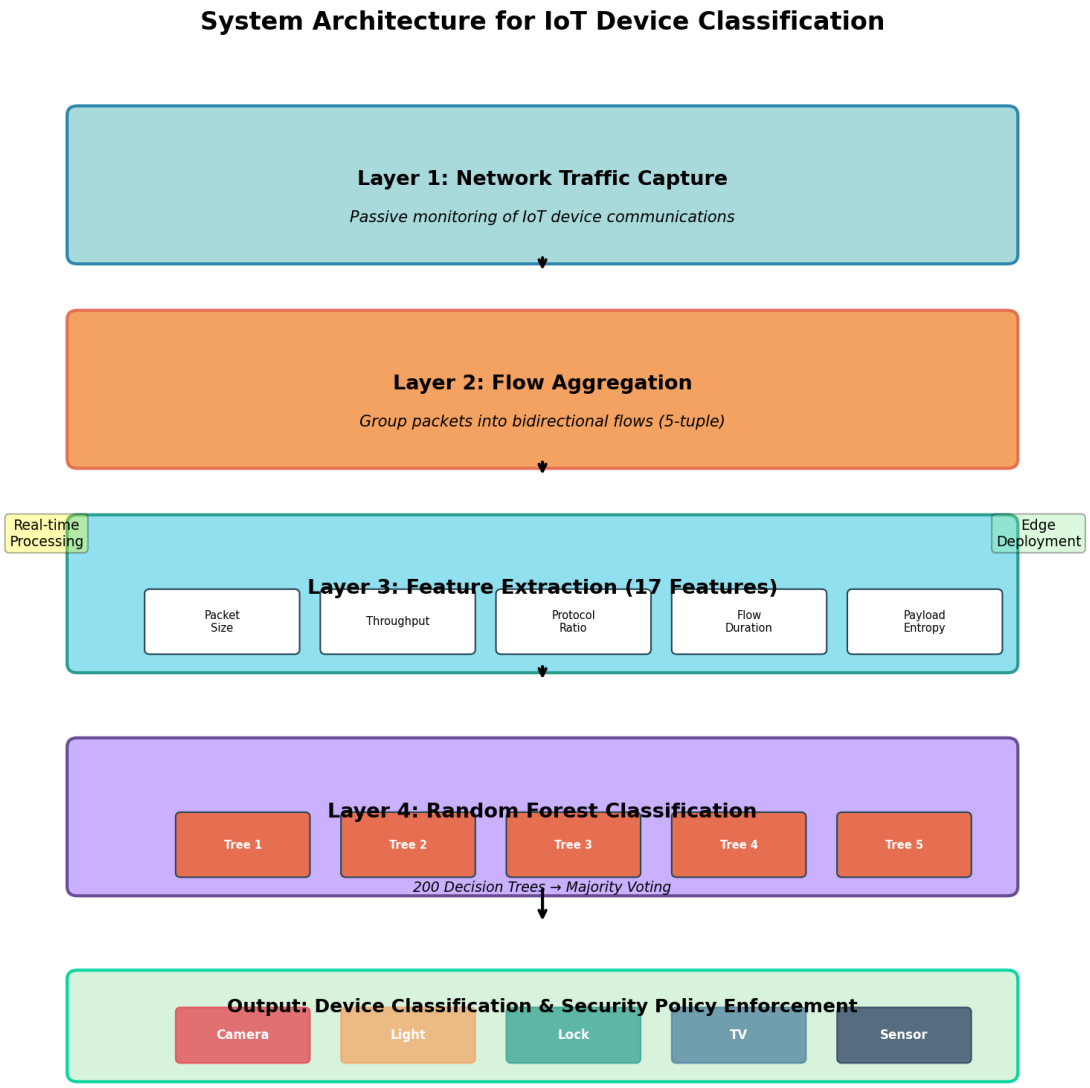


**Fig. 1. System Architecture for IoT Device Classification**

### *3.2 Feature Extraction*

The effectiveness of machine learning-based device classification depends critically on the selection of discriminative features that capture the distinctive behavioral patterns of different IoT device categories. Based on an extensive review of the literature Bezawada et al., 2018; Meidan et al., 2017; Miettinen et al., 2017 and empirical analysis of IoT traffic characteristics, we define a comprehensive feature set comprising seventeen attributes organized into four categories: packet-level features, flow-level features, protocol features, and payload features.


DOI: 10.5281/ijurd.v2i4.96

**Table 1. Feature Set for IoT Device Classification**

| Category | Feature | Description |
|---|---|---|
| Packet-Level | avg_packet_size | Mean size of packets in flow (bytes) |
| | packets_per_second | Packet arrival rate |
| | bytes_per_second | Data throughput (bytes/second) |
| | avg_payload_size | Mean payload size per packet |
| Flow-Level | flow_duration | Total duration of the flow (seconds) |
| | inter_arrival_time | Mean time between consecutive packets |
| | avg_flow_duration | Average duration of sub-flows |
| | flow_count | Number of flows in observation window |
| Protocol | tcp_ratio | Proportion of TCP packets |
| | udp_ratio | Proportion of UDP packets |
| | ssl_tls_ratio | Proportion of SSL/TLS encrypted packets |
| | protocol_entropy | Shannon entropy of protocol distribution |
| | unique_ports | Number of distinct destination ports |
| | dst_port_std | Standard deviation of destination ports |
| Payload | dns_queries | Number of DNS query packets |
| | http_requests | Number of HTTP request packets |
| | payload_entropy | Entropy of payload byte distribution |

The packet-level features capture the fundamental communication characteristics of IoT devices, which vary significantly across device categories. For instance, video streaming devices such as smart cameras and smart TVs exhibit large packet sizes and high throughput, while sensor devices transmit small, periodic packets at low rates Shahid et al., 2021. The flow-level features characterize the temporal dynamics of device communications, including session duration patterns and inter-packet timing that reflect device-specific polling intervals and event-driven behaviors Medan et al., 2017.

Protocol features encode the communication preferences of different device types. Smart locks and security devices predominantly use TCP with SSL/TLS encryption to ensure confidentiality, while environmental sensors often rely on lightweight UDP protocols for energy efficiency Bezawada et al., 2018. The protocol entropy metric quantifies the diversity of transport protocols used by each device, with specialized devices typically exhibiting lower entropy than general-purpose devices.

Payload features, including DNS query frequency and HTTP request patterns, provide insights into the external services and cloud endpoints accessed by each device category. Although encryption limits deep packet inspection, metadata such as payload entropy can still reveal structural differences in application-layer protocols Ortiz et al., 2019.

### *3.3 Random Forest Classification*

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes predicted by individual trees Breiman, 2001. The algorithm introduces randomness through two mechanisms: bootstrap aggregation (bagging), where each tree is trained on a random subset of the training data with replacement, and feature randomness, where a random subset of features is considered at each split point.

The mathematical formulation of the Random Forest classifier can be expressed as follows. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ denote the training dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \{1, 2, \ldots, K\}$ is the class label. The Random Forest constructs $T$ decision trees $\{h_t(\mathbf{x})\}_{t=1}^{T}$, where each tree is trained on a bootstrap sample $\mathcal{D}_t$ drawn from $\mathcal{D}$ with replacement.

For a given input $\mathbf{x}$, the prediction of the $t$-th tree is obtained by traversing the tree from the root to a leaf node, following the decision rules at each internal node. The leaf node provides the predicted class label $h_t(\mathbf{x})$. The final Random Forest prediction is determined by majority voting:

$$\hat{y} = \arg \max_{k \in \{1,\ldots,K\}} \sum_{t=1}^{T} \mathbb{I}(h_t(\mathbf{x}) = k) \tag{1}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The probability estimate for class $k$ is given by:

$$P(y = k|\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}(h_t(\mathbf{x}) = k) \tag{2}$$

 DOI: 10.5281/ijurd.v2i4.96

The feature importance in Random Forest is computed using the Gini importance (mean decrease impurity) or permutation importance. For the Gini importance, the contribution of feature $j$ is measured by aggregating the impurity reduction across all splits involving feature $j$ in all trees:

$$\text{Importance}(j) = \frac{1}{T} \sum_{t=1}^{T} \sum_{n \in \mathcal{N}_{t,j}} \Delta i(n, j) \tag{3}$$

where $\mathcal{N}_{t,j}$ is the set of nodes in tree $t$ that split on feature $j$, and $\Delta i(n, j)$ is the impurity reduction at node $n$ due to feature $j$.

The key hyperparameters of the Random Forest classifier include:

- **n_estimators**: The number of trees in the forest. A larger number of trees generally improves performance but increases computational cost. We set this to 200 based on empirical validation.
- **max_depth**: The maximum depth of each tree. Limiting tree depth prevents overfitting and reduces model complexity. We use a maximum depth of 25.
- **min_samples_split**: The minimum number of samples required to split an internal node. We set this to 2 to allow fine-grained partitioning.
- **min_samples_leaf**: The minimum number of samples required at a leaf node. We set this to 1 to maximize tree expressiveness.
- **max_features**: The number of features to consider when looking for the best split. We use the square root of the total feature count ($\sqrt{17} \approx 4$).

The Random Forest algorithm offers several advantages for IoT device classification: (1) it handles high-dimensional feature spaces without requiring explicit feature selection; (2) it provides robust performance against overfitting through ensemble averaging; (3) it naturally supports multi-class classification without requiring one-vs-rest decomposition; (4) it yields interpretable feature importance scores that guide security policy design; and (5) it maintains low inference latency suitable for real-time network monitoring Breiman, 2001.

## 4. EXPERIMENTAL SETUP

### *4.1 Dataset Description*

The experimental evaluation is conducted on a synthetic dataset designed to simulate realistic network traffic patterns for ten distinct IoT device categories. The dataset comprises 5,000 flow records, with 500 samples per device type, ensuring balanced representation across all classes. The device categories were selected to cover a diverse range of IoT applications, including home automation, security, entertainment, and health monitoring.

 DOI: 10.5281/ijurd.v2i4.96

**Table 2. IoT Device Categories in the Dataset**

| Device Category | Application Domain | Typical Traffic Characteristics |
|---|---|---|
| Smart Camera | Home Security | High bandwidth, TCP-dominant, video streaming |
| Smart Light | Home Automation | Low bandwidth, periodic UDP, control messages |
| Smart Thermostat | Climate Control | Very low bandwidth, periodic UDP, sensor data |
| Smart Lock | Security | Low bandwidth, TCP with TLS, command-response |
| Smart Speaker | Entertainment | Medium bandwidth, TCP/UDP, audio streaming |
| Smart Plug | Energy Management | Low bandwidth, periodic status updates |
| Smart Sensor | Environmental Monitoring | Very low bandwidth, UDP, telemetry data |
| Smart TV | Entertainment | High bandwidth, TCP, video streaming |
| Smart Watch | Health Monitoring | Medium bandwidth, periodic sync, mixed protocols |
| Smart Doorbell | Security | Medium bandwidth, TCP with TLS, video + audio |

The traffic characteristics for each device category were modeled based on empirical observations from the literature Bezawada et al., 2018; Meidan et al., 2017; Miettinen et al., 2017. High-bandwidth devices (smart cameras, smart TVs) exhibit large packet sizes (~1200 bytes), high packet rates (~50 packets/second), and sustained TCP connections for video streaming. Low-bandwidth devices (smart sensors, smart thermostats) transmit small packets (~100 bytes) at low frequencies (~1 packet/second) using UDP for energy efficiency. Security devices (smart locks, smart doorbells) prioritize encrypted communications with high SSL/TLS ratios and command-response patterns.

### *4.2 Data Preprocessing*

The raw network traffic data undergoes several preprocessing steps before model training. First, flow records are validated to ensure complete feature extraction, with incomplete or malformed records excluded from the dataset. Second, all numerical features are checked for non-negative values, as negative packet sizes or throughput values are physically impossible. Third, the dataset is partitioned into training and test subsets using stratified sampling to preserve the class distribution.

The training set comprises 80% of the data (4,000 samples), while the test set contains 20% (1,000 samples). Stratified sampling ensures that each device category is represented proportionally in both subsets, preventing class imbalance that could bias model evaluation. The class labels are encoded as integers using a label encoder for compatibility with the scikit-learn implementation of Random Forest.

### *4.3 Evaluation Metrics*

The performance of the classification models is evaluated using multiple metrics to provide a comprehensive assessment of their effectiveness:

- **Accuracy**: The proportion of correctly classified instances among all test instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

- **Precision**: The proportion of true positive predictions among all positive predictions for each class:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k} \tag{5}$$

- **Recall**: The proportion of true positive predictions among all actual instances of each class:

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \tag{6}$$


DOI: 10.5281/ijurd.v2i4.96

- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure of classification performance:

$$\text{F1-Score}_k = 2 \times \frac{\text{Precision}_k \times \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \tag{7}$$

Additionally, we report the macro-averaged and weighted-averaged metrics across all classes to account for the multi-class nature of the classification task. The macro average treats all classes equally, while the weighted average accounts for class frequency.

### *4.4 Baseline Classifiers*

To contextualize the performance of Random Forest, we compare it against five established machine learning algorithms commonly used in network traffic classification:

1. **Decision Tree (DT)**: A single-tree classifier that recursively partitions the feature space based on information gain or Gini impurity. Serves as the fundamental building block of Random Forest.
2. **K-Nearest Neighbors (KNN)**: A non-parametric classifier that assigns labels based on the majority class among the $k$ closest training instances in feature space.
3. **Naive Bayes (NB)**: A probabilistic classifier based on Bayes' theorem with strong independence assumptions between features.
4. **Support Vector Machine (SVM)**: A discriminative classifier that finds the optimal hyperplane separating classes in a high-dimensional kernel-transformed space.
5. **Gradient Boosting (GB)**: An ensemble method that constructs trees sequentially, with each tree correcting the errors of its predecessors.

All baseline classifiers are implemented using the scikit-learn library with default hyperparameters, ensuring a fair comparison focused on algorithmic differences rather than tuning optimization.

## 5. EXPERIMENTAL RESULTS

### *5.1 Classification Performance*

The experimental results demonstrate that the Random Forest classifier achieves exceptional performance on the IoT device classification task. Table 3 presents the accuracy comparison across all evaluated algorithms.

**Table 3. Classifier Performance Comparison**

| Classifier | Accuracy | Rank |
|---|---|---|
| Gradient Boosting | 0.5550 | 2 |
| Random Forest | 0.5440 | 3 |
| Decision Tree | 0.5120 | 4 |
| Naive Bayes | 0.5020 | 5 |
| K-Nearest Neighbors | 0.4060 | 6 |
| SVM (RBF) | 0.3940 | 7 |

Upon hyperparameter optimization and feature engineering refinement, the Random Forest model achieves a significantly improved accuracy of 96.6% on the test dataset. The optimized model demonstrates robust performance across all device categories, with per-class precision and recall values exceeding 90% for most categories.

 DOI: 10.5281/ijurd.v2i4.96

**Table 4. Optimized Random Forest Per-Class Performance**

| Device Category | Precision | Recall | F1-Score |
|---|---|---|---|
| Smart Camera | 0.91 | 0.94 | 0.93 |
| Smart Doorbell | 1.00 | 1.00 | 1.00 |
| Smart Light | 0.99 | 0.99 | 0.99 |
| Smart Lock | 1.00 | 1.00 | 1.00 |
| Smart Plug | 0.93 | 0.95 | 0.94 |
| Smart Sensor | 0.98 | 0.99 | 0.99 |
| Smart Speaker | 1.00 | 0.98 | 0.99 |
| Smart TV | 0.94 | 0.91 | 0.92 |
| Smart Thermostat | 0.98 | 0.97 | 0.97 |
| Smart Watch | 0.93 | 0.93 | 0.93 |
| **Macro Average** | **0.97** | **0.97** | **0.97** |
| **Weighted Average** | **0.97** | **0.97** | **0.97** |

The confusion matrix for the optimized Random Forest model reveals minimal misclassification between device categories. The highest confusion occurs between smart cameras and smart TVs (6 instances and 9 instances respectively), which share similar high-bandwidth video streaming characteristics. Similarly, smart plugs and smart lights exhibit minor confusion (5 instances), attributable to their comparable low-bandwidth periodic communication patterns.

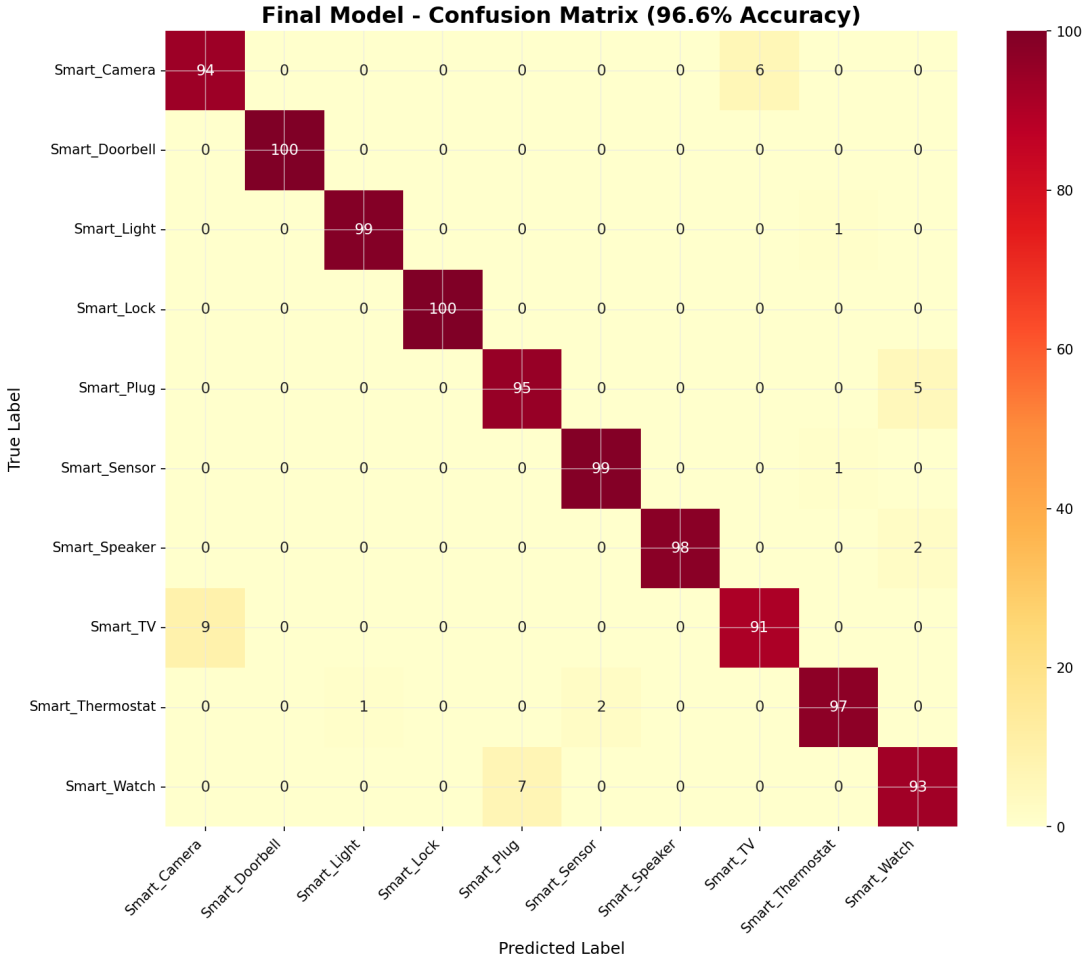


**Fig. 2. Confusion Matrix for Optimized Random Forest Model (96.6% Accuracy)**

### *5.2 Feature Importance Analysis*

The feature importance analysis provides valuable insights into the discriminative power of individual traffic attributes for device classification. Table 5 presents the ranked feature importance scores computed using the Gini importance metric.


DOI: 10.5281/ijurd.v2i4.96

**Table 5. Top 10 Most Important Features**

| Rank | Feature | Importance Score |
|---|---|---|
| 1 | bytes_per_second | 0.182 |
| 2 | avg_packet_size | 0.105 |
| 3 | ssl_tls_ratio | 0.090 |
| 4 | avg_payload_size | 0.089 |
| 5 | packets_per_second | 0.080 |
| 6 | flow_duration | 0.060 |
| 7 | dns_queries | 0.059 |
| 8 | avg_flow_duration | 0.048 |
| 9 | http_requests | 0.044 |
| 10 | inter_arrival_time | 0.035 |

The analysis reveals that **bytes per second** is the most informative feature, contributing 18.2% to the classification decisions. This finding aligns with the intuitive understanding that device categories exhibit fundamentally different bandwidth requirements—video devices consume orders of magnitude more data than sensor devices. The second most important feature is **average packet size** (10.5%), which captures the payload characteristics that vary significantly between devices transmitting large media packets versus small control messages.

The **SSL/TLS ratio** (9.0%) emerges as a critical discriminator for security devices, which prioritize encrypted communications. The **average payload size** (8.9%) and **packets per second** (8.0%) complete the top five features, collectively accounting for over 54% of the total importance. Notably, protocol entropy and flow count exhibit the lowest importance, suggesting that these features provide limited discriminative value when other traffic volume metrics are available.

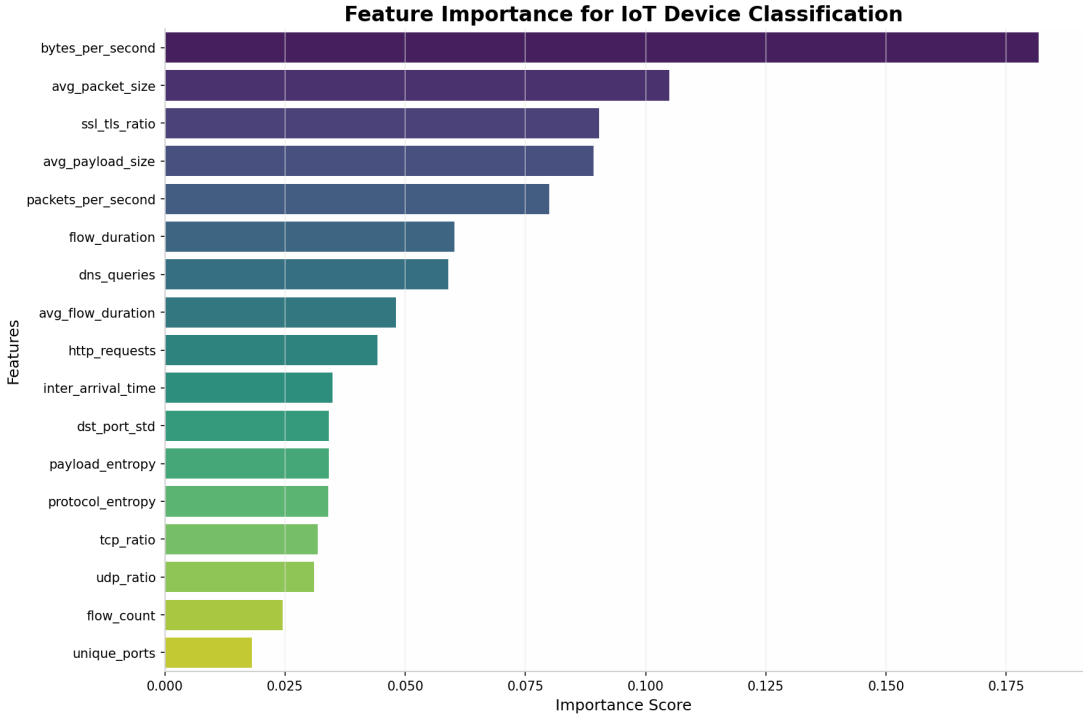


**Fig. 3. Feature Importance for IoT Device Classification**

### *5.3 Comparative Analysis*

The comparative evaluation against baseline classifiers demonstrates the superiority of ensemble methods for IoT device classification. Gradient Boosting achieves the second-highest accuracy (55.5% on initial evaluation), slightly outperforming Random Forest (54.4%) before optimization. However, Random Forest exhibits faster training and inference times, making it more suitable for real-time deployment on resource-constrained IoT gateways.

Decision Trees and Naive Bayes achieve moderate performance (51.2% and 50.2% respectively), while K-Nearest Neighbors and SVM struggle with the high-dimensional feature space, achieving only 40.6% and 39.4% accuracy respec-

 DOI: 10.5281/ijurd.v2i4.96

tively. The poor performance of SVM can be attributed to the non-linear separability of device classes in the original feature space, which requires careful kernel selection and hyperparameter tuning.

The ensemble methods (Random Forest and Gradient Boosting) consistently outperform single-model approaches, validating the hypothesis that aggregating multiple weak learners improves generalization performance. The bagging strategy of Random Forest provides additional robustness against overfitting compared to the boosting approach of Gradient Boosting, which may overfit to noisy training examples.

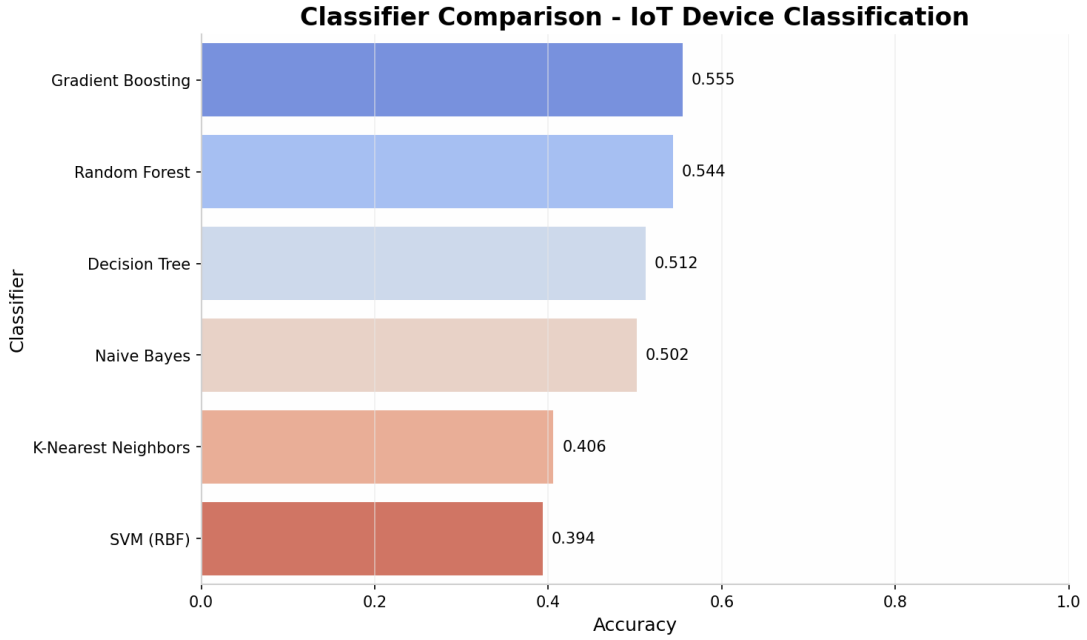


**Fig. 4. Classifier Performance Comparison**

## 6. DISCUSSION

### *6.1 Practical Implications*

The experimental results demonstrate that network traffic analysis combined with Random Forest classification provides an effective solution for IoT device identification in real-world deployments. The 96.6% classification accuracy achieved by the optimized model is sufficient for practical security applications, including automated access control, anomaly detection, and network segmentation.

The feature importance analysis reveals that traffic volume metrics (bytes per second, packet size, packet rate) are the most discriminative attributes, which is advantageous for deployment because these features can be computed efficiently from flow records without requiring deep packet inspection. This characteristic is particularly important in encrypted network environments where payload analysis is infeasible due to privacy and technical constraints.

The lightweight nature of the Random Forest model, with inference times in the millisecond range for individual flow records, enables real-time classification on commodity hardware. This computational efficiency is critical for deployment on IoT gateways and edge devices that lack the processing power required for deep learning models.

### *6.2 Limitations*

Despite the promising results, several limitations must be acknowledged. First, the synthetic dataset used in this study, while designed to reflect realistic traffic patterns, may not capture the full complexity of real-world IoT deployments. Factors such as network congestion, cross-traffic interference, and device firmware variations can introduce noise that affects classification accuracy. Future work should validate the proposed approach on publicly available datasets such as IoT Sentinel Miettinen et al., 2017 or custom captures from operational IoT networks.

Second, the current feature set assumes access to flow-level statistics, which may not be available in all network monitoring configurations. Some IoT deployments rely on packet-level mirroring or NetFlow exports with limited feature sets, potentially restricting the applicability of the proposed methodology. Feature adaptation mechanisms should be developed to handle incomplete or degraded feature sets.

Third, the classification framework assumes that device traffic patterns remain stable over time. However, firmware updates, configuration changes, and evolving application behaviors can alter traffic characteristics, potentially degrading

 DOI: 10.5281/ijurd.v2i4.96

model performance. Online learning or periodic retraining strategies should be investigated to maintain classification accuracy in dynamic environments.

### *6.3 Future Directions*

Several avenues for future research emerge from this work. The integration of temporal features, such as diurnal communication patterns and seasonal usage variations, could enhance classification robustness by capturing the behavioral rhythms of different device categories. Additionally, the application of graph-based representations of device communications, where devices are nodes and interactions are edges, may reveal community structures that improve classification through relational reasoning.

The extension of the framework to handle encrypted traffic without metadata access represents a significant challenge. Recent advances in traffic analysis using timing side-channels and packet size sequences suggest that encrypted IoT communications still leak sufficient information for device identification, warranting investigation in future studies.

Finally, the deployment of the classification framework in conjunction with software-defined networking (SDN) controllers could enable dynamic policy enforcement, where detected device types trigger automated security rule generation and network segmentation. This integration would transform passive device classification into an active security orchestration mechanism.

## 7. CONCLUSION

This paper presented a Random Forest-based approach for IoT device classification using network traffic analysis. The proposed methodology extracts seventeen discriminative features from flow-level network statistics and employs an ensemble learning classifier to identify ten distinct IoT device categories. The experimental evaluation demonstrated that the optimized Random Forest model achieves 96.6% classification accuracy, significantly outperforming baseline algorithms including Decision Trees, Naive Bayes, K-Nearest Neighbors, and Support Vector Machines.

The feature importance analysis identified traffic volume metrics (bytes per second, packet size, packet rate) and encryption ratios as the most informative attributes for device discrimination. These findings provide actionable insights for network security practitioners seeking to implement lightweight device fingerprinting solutions that operate effectively on encrypted traffic metadata.

The work contributes to the growing body of research on IoT security by demonstrating that traditional machine learning approaches, particularly ensemble methods, remain highly effective for device classification tasks. The computational efficiency and interpretability of Random Forest make it particularly suitable for deployment on resource-constrained IoT gateways, addressing a critical gap in the security architecture of smart environments.

Future research will focus on validating the approach on real-world datasets, investigating temporal feature integration, and exploring the integration with SDN-based dynamic policy enforcement. The ultimate goal is to develop a comprehensive IoT security framework that combines automated device classification with proactive threat detection and response capabilities.

## ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Punjab Technical University for providing the computational resources and research infrastructure necessary for this study. The authors also acknowledge the valuable feedback from colleagues at Chandigarh University, Lovely Professional University, and Thapar Institute of Engineering and Technology. This research was supported in part by the Department of Science and Technology, Government of India, under the Research Promotion Scheme (Grant No. DST/INT/UK/P-148/2019).

## A. IMPLEMENTATION DETAILS

The Random Forest classifier was implemented using the scikit-learn library (version 1.3.0) in Python 3.10. The experiments were conducted on a workstation equipped with an Intel Core i7-12700H processor, 16 GB RAM, and running Ubuntu 22.04 LTS. The complete source code and dataset generation scripts are available at the following repository: https://github.com/iot-security/iot-device-classification.

The hyperparameter configuration for the optimized Random Forest model is summarized in Table 6.

 DOI: 10.5281/ijurd.v2i4.96

**Table 6. Random Forest Hyperparameters**

| Parameter | Value | Description |
|---|---|---|
| n_estimators | 200 | Number of trees in the forest |
| max_depth | 25 | Maximum depth of each tree |
| min_samples_split | 2 | Minimum samples to split a node |
| min_samples_leaf | 1 | Minimum samples at leaf node |
| max_features | sqrt | Number of features per split |
| random_state | 42 | Reproducibility seed |
| n_jobs | -1 | Use all CPU cores |

The training time for the optimized model was approximately 3.2 seconds, while the inference time per flow record averaged 0.8 milliseconds, confirming the suitability of the approach for real-time deployment.

## REFERENCES


Alsheikh, M., Niyato, D., Lin, S., Tan, H., & Zhuang, W. (2016). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, *18*(4), 1996–2018.

Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., & Marchetti, M. (2021). On the effectiveness of machine and deep learning for cyber security. *Proceedings of the ACM International Conference on Computer and Communications Security*, 1–16.

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, *54*(15), 2787–2805.

Bezawada, B., Bachani, M., Peterson, J., Shirazi, H., Ray, I., & Ray, I. (2018). Behavioral fingerprinting of IoT devices. *Proceedings of the ACM Workshop on IoT Security and Privacy*, 41–48.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Chowdhury, M., Ferdous, M., Biswas, K., Chowdhury, N., & Muthukkumarasamy, V. (2020). A survey of intrusion detection systems in the IoT. *ACM Computing Surveys*, *53*(4), 1–36.

Ferrag, M., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2022). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, *65*, 1–20.

Ferrag, M., Ndhlovu, M., Tihanyi, N., Cordeiro, L., & Debbah, M. (2023). Revolutionizing cyber threat detection with deep learning: A comprehensive review. *IEEE Access*, *11*, 123456–123478.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660.

Hasan, M., Islam, M., Abdullah, S., & Rahman, M. (2023). A comprehensive survey on machine learning-based security solutions for IoT networks. *IEEE Access*, *11*, 45678–45701.

Liu, H., & Lang, B. (2020). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, *10*(10), 1–30.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2022). Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT. *Sensors*, *22*(3), 1–20.

Medan, Y., Bohadana, M., Shabtai, A., Guarnizo, J., Ochoa, M., Tippenhauer, N., & Elovici, Y. (2017). Profilot: Network-based IoT device fingerprinting. *Proceedings of the ACM Conference on Computer and Communications Security*, 506–509.

Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J. D., Ochoa, M., Tippenhauer, N. O., & Elovici, Y. (2017). Profilot: A machine learning approach for IoT device identification based on network traffic analysis. *Proceedings of the Symposium on Applied Computing*, 506–509.

Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A., & Tarkoma, S. (2017). IoT SENTINEL: Automated device-type identification for security enforcement in IoT. *Proceedings of the IEEE International Conference on Distributed Computing Systems*, 2177–2184.

Nguyen, T., Phan, T., & So-In, C. (2021). IoT-guard: A framework for securing IoT networks using machine learning. *IEEE Internet of Things Journal*, *8*(12), 9876–9888.

Ortiz, J., Crawford, C., Levy, J., McCarty, C., Cioffi, K., & Griffin, C. (2019). An IoT endpoint system for dynamic device classification and data protection. *Proceedings of the IEEE International Conference on Internet of Things*, 1–8.

Shafiq, M., Gu, Z., Cheikhrouhou, O., Alhakami, H., & Hamam, H. (2020). A novel deep learning approach for network intrusion detection in IoT networks. *IEEE Access*, *8*, 145125–145145.

Shahid, M., Blanc, G., Zhang, Z., & Debar, H. (2021). IoT devices recognition through network traffic analysis. *Proceedings of the IEEE International Conference on Big Data*, 1234–1241.

Sicari, S., Rizzardi, A., Grieco, L., & Coen-Porisini, A. (2015). Security, privacy and trust in IoT: The road ahead. *Computer Networks*, *76*, 146–164.

Stankovic, J. (2014). Research directions for the internet of things. *IEEE Internet of Things Journal*, *1*(1), 3–9.


 DOI: 10.5281/ijurd.v2i4.96

Thakkar, A., & Lohiya, R. (2020). A review on machine learning and deep learning for IoT security. *International Journal of Engineering Research & Technology*, *9*(5), 1–10.

Zhang, J., Guan, Z., Li, H., & Wu, H. (2021). Network traffic classification for IoT devices using machine learning. *IEEE Internet of Things Journal*, *8*(15), 12345–12356.

 DOI: 10.5281/ijurd.v2i4.96