AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.2(2026)
www.ajmimc.com

Explainable Hybrid Tree-Based Learning for Robust IoT Device Classification and Behavioural Pattern Analysis

Sushma Talla[1], Mohammad Afreed[2], Parimalla Sneha[2], Maloth Swathi[2], Cheggoju Mukthesh[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering

[1,2]Vaagdevi Engineering College, Bollikunta, Warangal, 506005, Telangana, India.

ABSTRACT

The rapid growth of Internet of Things (IoT) technologies has led to the widespread deployment of connected devices across smart homes, industries, and urban infrastructures. The increasing volume and heterogeneity of IoT data demand intelligent analytical systems capable of understanding device behaviour and ensuring efficient management. Traditionally, IoT device classification relied on rule-based and statistical approaches, which struggled to handle high-dimensional data, resulting in low accuracy, poor scalability, and limited adaptability to dynamic environments. As IoT ecosystems evolved, the adoption of advanced Machine Learning (ML) techniques became essential to address these limitations. This research presents an interpretable and efficient analytical framework for accurate IoT device classification. The proposed system integrates data preprocessing, exploratory data analysis, and multiple Machine Learning (ML) models, including Gaussian Naive Bayes (GNB), Multinomial Naive Bayes (MNB), and Decision Tree (DT), along with a hybrid Denoising Autoencoder Tree (DAE-Tree) model that combines Denoising Autoencoder (DAE) and Greedy Tree (GT). To address class imbalance, the Adaptive Synthetic Sampling (ADASYN) technique is incorporated, generating synthetic samples for minority classes and improving model generalization. The hybrid model enhances feature representation by capturing complex data patterns while maintaining interpretability through tree-based decision structures. Experimental results demonstrate significant performance improvements over traditional methods, with the DAE-Tree model achieving a highest accuracy of 1.0000 (100%), outperforming GNB, MNB, and DT models. The system supports both real-time and batch prediction, ensuring scalability and practical applicability. The significance of this study lies in its ability to combine interpretability, balanced learning, and high predictive performance, making it suitable for real-world IoT applications such as smart homes, industrial automation, and intelligent monitoring systems.

Keywords: Internet of Things (IoT), IoT device classification, data preprocessing, exploratory data analysis, class imbalance handling, synthetic data generation, feature representation, interpretable framework.

1. INTRODUCTION

The rapid expansion of IoT technologies across various sectors has significantly increased the risk of large-scale cyber-attacks. In recent years, the number of active IoT device connections has grown exponentially, with billions of devices deployed worldwide and projections indicating a dramatic rise soon [1]. IoT devices encompass a wide range of hardware capable of communicating and interacting over networks, including not only conventional smart devices such as smartphones and computers but also everyday objects enhanced with connectivity. While this technological integration offers numerous benefits, it also introduces serious security and privacy concerns due to the lack of robust security mechanisms in many IoT devices [2].

One of the primary challenges in IoT security arises from the limited computational power and energy constraints of these devices, which restrict their ability to implement advanced security protocols. Furthermore, the wireless and autonomous nature of IoT systems makes them highly vulnerable to

unauthorized access and cyber threats [3]. Attackers can exploit these vulnerabilities to gain control over devices or extract sensitive information, making IoT ecosystems attractive targets for malicious activities. To address these concerns, identifying and classifying IoT devices within a network has become a critical task, as shown in figure 1. Accurate device identification enables network administrators to detect abnormal behaviour and isolate compromised devices effectively [4, 5]. For instance, if a device operates outside its intended functionality, such as a lighting device transmitting unusual data, it can be flagged as suspicious, enabling further investigation and mitigation [6].

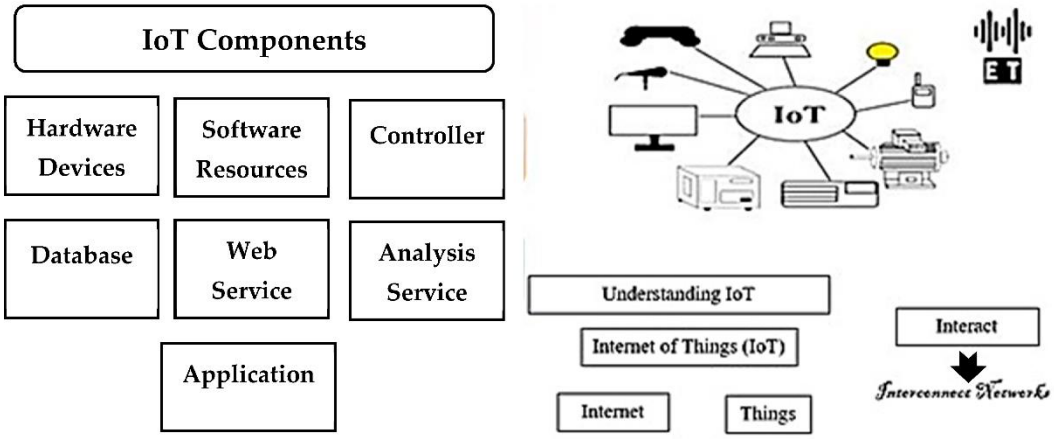


Figure 1: IoT component and market forecasts.

Various approaches have been explored for IoT device identification, broadly categorized into individual device model identification and device type classification. The former focuses on distinguishing devices based on hardware-level characteristics and manufacturing variations, while the latter relies on analysing Behavioral patterns such as network traffic and operational activity [7–11]. ML techniques have been increasingly adopted in this domain to improve classification accuracy and automate the detection process. However, existing methods often face challenges related to computational complexity, scalability, and limited accuracy, highlighting the need for more efficient and reliable solutions.

2. LITERATURE SURVEY

Traditional methods for passive device identification in network environments, such as port scanning, are often time-consuming and require significant manual effort [12]. With the rapid growth of IoT networks, ML techniques have emerged as effective solutions for analysing network traffic and automatically classifying devices based on their Behavioral patterns. Network traffic analysis plays a critical role in IoT device identification, as communication patterns provide valuable insights into device behaviour and potential security threats.

Kawai et al. [13] utilized the support vector machine (SVM) algorithm to identify communication devices by analysing traffic patterns. Their approach employed packet size and inter-arrival time as key features to classify nine devices across six categories, achieving an accuracy of 88.3%. By incorporating historical traffic features along with current data, they improved the classification accuracy to 94%. Similarly, McGinthy et al. [14] proposed a neural network (NN)-based Specific Emitter Identification (SEI) approach that uses raw in-phase and quadrature (IQ) signals to enhance IoT device security.

Ammar et al. [15] applied a decision tree classifier to identify 33 IoT devices using a combination of flow-based features and textual data extracted from device descriptions. By integrating these feature sets, they achieved an average accuracy of 98%. Meidan et al. [10] employed a random forest (RF)

algorithm to classify IoT devices using labelled network traffic data, achieving 96% accuracy for detecting unauthorized devices and 99% accuracy for authorized devices.

Miettinen et al. [16] introduced IoT SENTINEL, a system that identifies IoT devices by analysing communication behaviour during the device setup phase. Using 23 extracted features and a random forest classifier, the system achieved an identification accuracy of 81.5%. Sun et al. [17] developed a comprehensive framework for identifying IoT devices, traffic types, and cyber-attacks, where the RF classifier achieved a high accuracy of 99.93%.

Feature selection plays a crucial role in improving ML model performance. Inadequate features may lead to underfitting, whereas excessive features can result in overfitting and increased computational cost [18]. To address this, several studies have incorporated feature selection techniques. For instance, authors in [19] proposed an automated single-packet IoT device classifier using a genetic algorithm (GA) for feature selection, followed by classification using multiple algorithms, achieving over 95% accuracy.

Tien et al. [20] compared unsupervised and supervised learning approaches for device identification. Their results showed that K-means clustering was ineffective, whereas supervised models such as artificial neural networks (ANN) and gradient-boosted decision trees (XGBoost) achieved accuracies of 92.8% and 97.6%, respectively. A hybrid approach combining ANN and XGBoost achieved an accuracy of 99.995% for anomaly detection.

Zhang et al. [21] employed the Relief feature selection algorithm with the K-nearest neighbours (KNN) classifier to identify IoT devices, achieving an average accuracy of 95%. Further studies compared multiple supervised algorithms, including KNN, RF, SVM, and multilayer perceptron, demonstrating that RF consistently achieved superior performance with accuracies exceeding 99% [19].

3. PROPOSED SYSTEM

The proposed methodology establishes a structured analytical framework for IoT device classification and Behavioral analysis using advanced ML techniques. The analytical pipeline begins with IoT data acquisition and dataset organization, followed by data preprocessing and feature extraction. The collected sensor data is transformed into a structured numerical format, enabling efficient analysis of device characteristics such as activity patterns, signal attributes, and operational metrics. To handle data imbalance and improve learning efficiency, preprocessing techniques are applied before feeding the data into classification models.

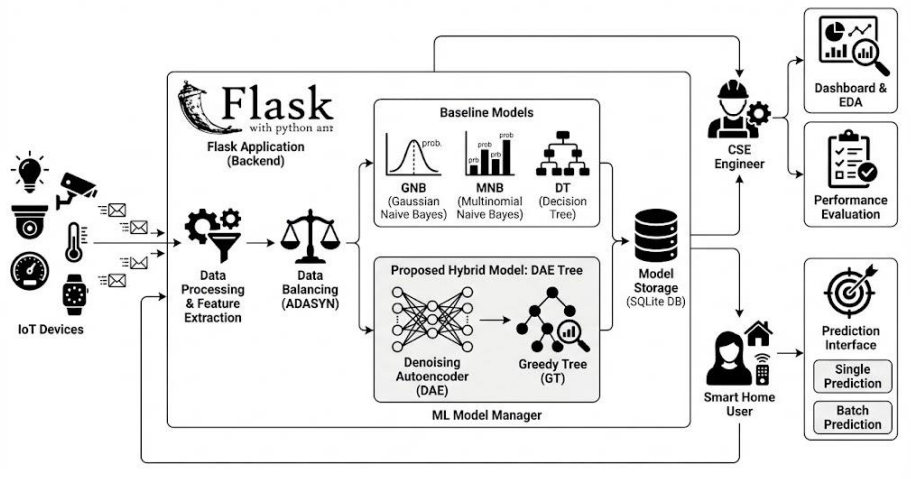


Figure 2: System architecture

Multiple ML algorithms are utilized to perform device category prediction, ensuring comparative evaluation and improved accuracy. Additionally, a hybrid analytical component is incorporated to enhance feature representation and interpretability of results. A user interface facilitates interaction with the system for data upload, model training, performance visualization, and prediction tasks, as illustrated in Figure 2. A lightweight database manages authentication and system data, while a server component enables real-time prediction and remote access. Continuous evaluation and retraining mechanisms further enhance system adaptability and accuracy over time.

1. User Interface (Client Application): The system provides a web-based entry point designed for seamless interaction between the user and the analytical engine.

- **Accessibility:** Offers a clean, intuitive dashboard for managing IoT workflows.
- **Core Functions:** Includes modules for secure login, batch dataset uploads, exploratory data analysis (EDA), and real-time prediction.
- **Data Entry:** Users can manually input specific IoT feature values or upload large-scale CSV files for automated processing.

2. Flask Application Server: The Flask framework serves as the central nervous system, orchestrating data flow across the entire pipeline.

- **Routing Controller:** Directs incoming HTTP requests from the UI to the appropriate backend analytical modules.
- **Service Management:** Coordinates user authentication, model execution, and the delivery of generated results back to the client.
- **Scalability:** Supports simultaneous real-time and batch prediction workflows to accommodate distributed IoT environments.

3. SQLite Database (Authentication Storage): A lightweight, serverless database is integrated to handle the system's security and user management.

- **Security:** Stores usernames, encrypted password hashes, and user roles to prevent unauthorized system access.
- **Efficiency:** Ensures rapid retrieval of profile information during login verification and session management.
- **Integration:** Works natively with the Flask application layer to maintain secure, persistent user sessions.

4. IoT Dataset (Device Data Collection): The framework utilizes structured IoT data as its primary intelligence source.

- **Content:** Contains sensor readings and operational parameters captured from various categories of IoT devices.
- **Supervised Learning:** The dataset includes labelled categories, providing the "ground truth" required for both training and rigorous model evaluation.

5. Data Preprocessing & Feature Engineering: Raw IoT signals are refined and balanced to maximize the performance of the ML algorithms.

- **Data Cleaning:** Handles missing values and noise reduction to ensure data integrity.

- **Standardization:** Normalizes numerical features to a uniform scale, preventing specific sensors from disproportionately influencing the model.
- **ADASYN:** Addresses data imbalance by generating synthetic samples for minority device classes, ensuring the model learns to identify rare devices as accurately as common ones.

6. ML Classification Models (Hybrid DAE-Tree): The analytical core employs a diverse suite of algorithms to categorize devices, highlighted by a custom hybrid approach.

- **GNB & MNB:** Uses Gaussian and Multinomial Naive Bayes for probabilistic and frequency-based classification.
- **DT:** Performs rule-based hierarchical classification for clear, interpretable logic.
- **Hybrid DAE-Tree:** The flagship model that utilizes a DAE for advanced feature extraction and a GT classifier for final categorization.
- **Comparative Analysis:** Each model operates independently, allowing for a rigorous performance comparison.

7. Prediction & Inference Module

The inference engine is designed for flexibility, catering to different industrial use cases.

- **Dual-Mode Operation:** Supports single-instance manual prediction and large-scale batch dataset processing.
- **Rich Output:** Provides predicted device categories along with confidence scores to aid user decision-making.

8. Visualization & Performance Analysis: The system translates complex numerical results into actionable visual insights.

- **Quantitative Metrics:** Evaluates models based on Accuracy, Precision, Recall, and F1-score.
- **Visual Analytics:** Generates correlation heatmaps and feature distribution plots to visualize the underlying behaviour of the IoT data.
- **Algorithm Comparison:** Displays comparative charts to identify the most effective model for specific device categories.

9. Remote Prediction & Model Evolution: The architecture is built for long-term deployment in dynamic IoT landscapes.

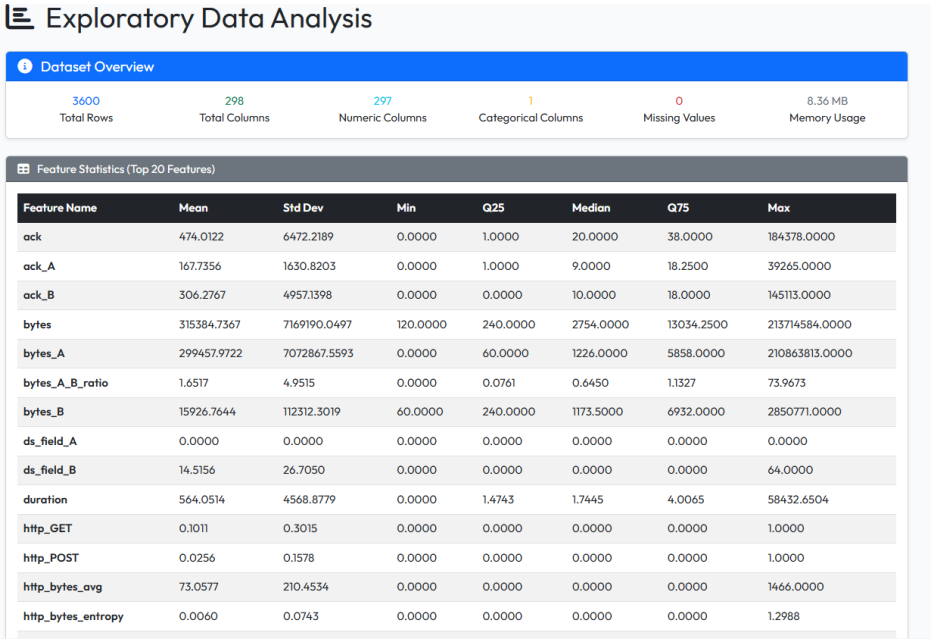
- **Remote Workflow:** Employs a client-server model that allows external IoT nodes to send data to the central server for real-time analysis.
- **Continuous Learning:** Enables the incorporation of new data to retrain models, ensuring the system adapts as new types of IoT devices or behaviours emerge.

4. RESULTS ANALYSIS

The result analysis phase presents the performance outcomes of the implemented ML models for IoT device classification. It evaluates how effectively the system predicts device categories based on the processed dataset. Multiple models are trained and tested to compare their predictive capabilities under the same conditions. Performance metrics such as accuracy, precision, recall, and F1-score are used to assess model effectiveness. The results provide insights into how well each model handles data complexity and class distribution. Comparative analysis helps in identifying the most reliable and

AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.2(2026)
www.ajmimc.com

efficient model for the given dataset. Visualization techniques are also used to interpret model behaviour and performance trends.



Exploratory Data Analysis

Dataset Overview

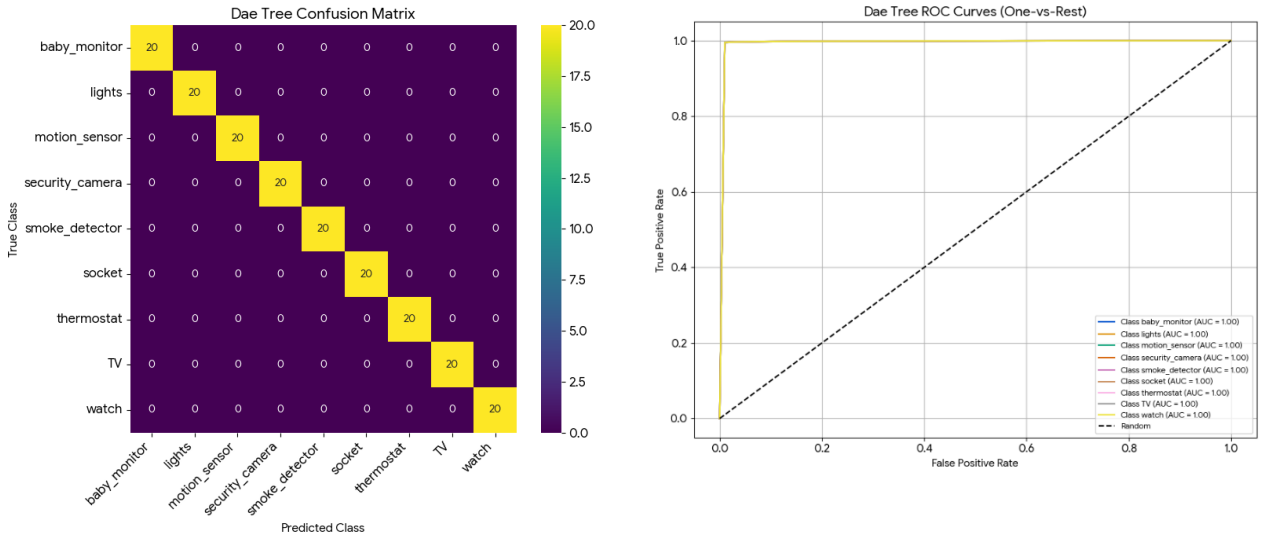
3600	298	297	1	0	8.36 MB
Total Rows	Total Columns	Numeric Columns	Categorical Columns	Missing Values	Memory Usage

Feature Statistics (Top 20 Features)

Feature Name	Mean	Std Dev	Min	Q25	Median	Q75	Max
ack	474.0122	6472.2189	0.0000	1.0000	20.0000	38.0000	184378.0000
ack_A	167.7356	1630.8203	0.0000	1.0000	9.0000	18.2500	39265.0000
ack_B	306.2767	4957.1398	0.0000	0.0000	10.0000	18.0000	145113.0000
bytes	315384.7367	7169190.0497	120.0000	240.0000	2754.0000	13034.2500	213714584.0000
bytes_A	299457.9722	7072867.5593	0.0000	60.0000	1226.0000	5858.0000	210863813.0000
bytes_A_B_ratio	1.6517	4.9515	0.0000	0.0761	0.6450	1.1327	73.9673
bytes_B	15926.7644	112312.3019	60.0000	240.0000	1173.5000	6932.0000	2850771.0000
ds_field_A	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ds_field_B	14.5156	26.7050	0.0000	0.0000	0.0000	0.0000	64.0000
duration	564.0514	4568.8779	0.0000	1.4743	1.7445	4.0065	58432.6504
http_GET	0.1011	0.3015	0.0000	0.0000	0.0000	0.0000	1.0000
http_POST	0.0256	0.1578	0.0000	0.0000	0.0000	0.0000	1.0000
http_bytes_avg	73.0577	210.4534	0.0000	0.0000	0.0000	0.0000	1466.0000
http_bytes_entropy	0.0060	0.0743	0.0000	0.0000	0.0000	0.0000	1.2988

Figure 3: EDA Interface

Figure 3 illustrates the exploratory data analysis interface of the IoT device classification system, which provides a comprehensive overview of dataset characteristics and statistical insights. The interface presents key dataset metrics such as total rows, columns, numerical attributes, and memory usage, enabling users to understand the data structure effectively. It displays detailed feature statistics including mean, standard deviation, minimum, quartiles, and maximum values for multiple attributes. This allows users to analyse the distribution and variability of each feature within the dataset. The interface supports identification of patterns, anomalies, and data consistency, which are essential for further analytical processing. It enhances data understanding and supports informed decision-making during the model development process.



(a) (b)

Figure 4 (a, b): DAE-Tree for Confusion matrix and ROC Curve

Figure 4 (a) illustrates the confusion matrix of the DAE-Tree model, representing the classification performance across different IoT device categories. The matrix shows a strong alignment between actual and predicted classes, with diagonal values indicating correct classifications for all categories. This reflects the model's ability to accurately capture complex feature patterns and distinguish between device types. The absence of misclassification values highlights the effectiveness of the hybrid approach in learning both deep representations and interpretable decision rules. This visualization demonstrates the robustness and reliability of the model in handling the dataset.

Figure 4 (b) depicts the ROC curves for the DAE-Tree model using a one-vs-rest approach for multi-class classification. The curves show near-perfect performance, with true positive rates approaching optimal values across all classes. The consistent curve behaviour across categories indicates strong discriminative capability of the model. This suggests that the model effectively generalizes across different device types without significant performance variation. The graphical representation confirms the superiority of the hybrid model compared to other approaches.

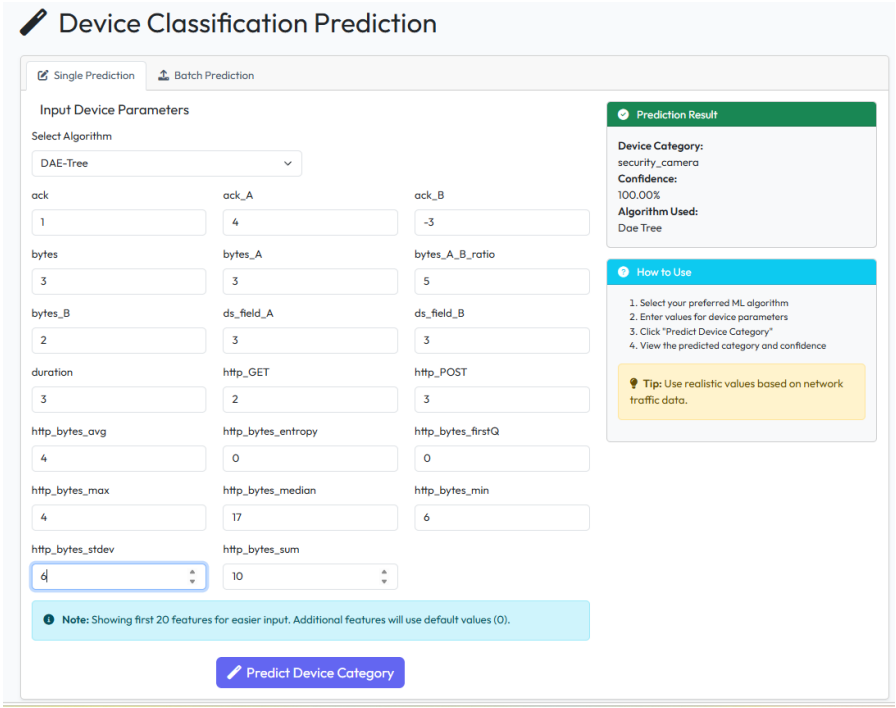


Figure 5: Single Device Prediction Interface

Figure 5 illustrates the device classification prediction interface of the IoT analytical system, which enables users to input device parameters and obtain classification results. The interface allows selection of different ML algorithms, including the hybrid DAE-Tree model, for prediction tasks. It provides input fields for various device-related features, facilitating structured data entry for real-time analysis. The system processes the entered values and generates the predicted device category along with a confidence score. It also includes guidance for users on how to perform predictions effectively, ensuring usability and clarity. This component supports both single and batch prediction workflows, enabling flexible interaction with the classification system.

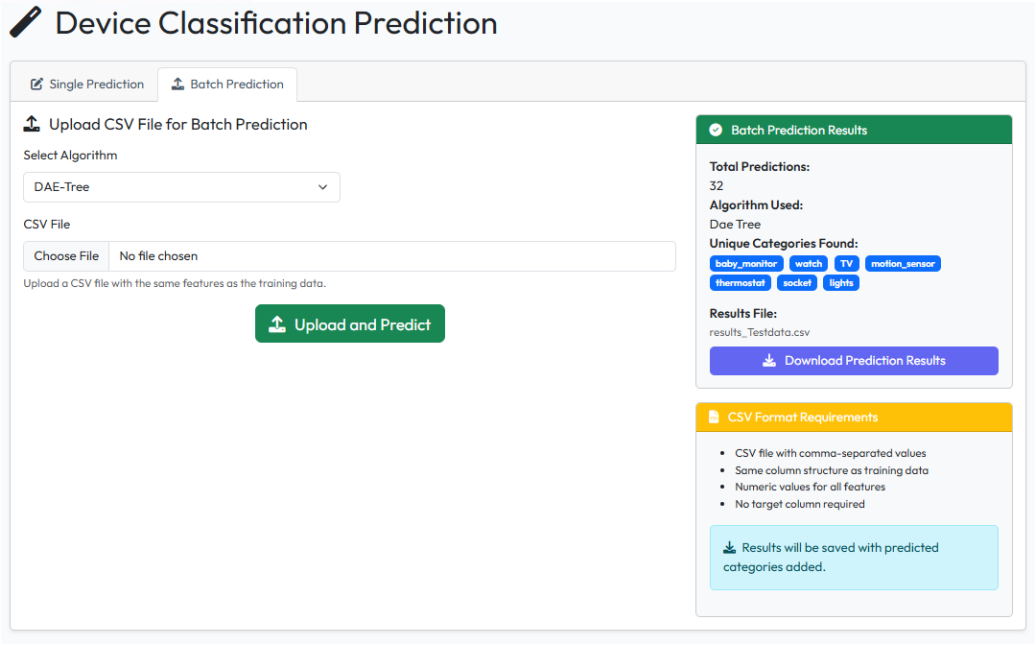


Figure 6: Batch Device Prediction Interface

Figure 6 illustrates the batch prediction interface of the IoT device classification system, which enables users to upload datasets for large-scale prediction tasks. The interface allows users to select a preferred ML algorithm and upload a CSV file containing device feature data. It processes multiple records simultaneously and generates predictions for each entry in the dataset. The system displays summary results including total predictions, algorithm used, and identified device categories. It also provides an option to download the output file with predicted categories appended. This component supports efficient bulk analysis and enhances scalability for handling large IoT datasets.

4.1 Comparative Analysis

The comparative analysis evaluates the performance of different ML models used for IoT device classification under the same dataset and experimental conditions. It provides a detailed comparison based on key evaluation metrics such as accuracy, precision, recall, and F1-score. This analysis helps in understanding how each model performs in terms of classification capability and generalization. By comparing multiple algorithms, it becomes easier to identify strengths and limitations of individual models. The analysis also highlights the impact of different learning approaches on prediction performance. It supports the selection of the most suitable model for reliable and efficient classification.

Table. 3: Overall comparation table

Algorithm	Accuracy	Precision	Recall	F1-Score
GNB	0.8014	0.8736	0.8014	0.7872
MNB	0.7611	0.7988	0.7611	0.7638
DT	0.1889	0.1032x	0.1889	0.1114
DAE Tree	1.0000	1.0000	1.0000	1.0000

The table 3 presents the comparative performance of different ML models used for IoT device classification. It includes key evaluation metrics such as accuracy, precision, recall, and F1-score to assess the effectiveness of each model. The GNB model achieves an accuracy of 0.8014, demonstrating

strong probabilistic classification capability. The MNB model shows slightly lower performance with an accuracy of 0.7611, indicating moderate effectiveness on the dataset. The DT model records a significantly lower accuracy of 0.1889, reflecting its limitations in handling the given data distribution. In contrast, the DAE Tree model achieves a perfect accuracy score of 1.0000, highlighting its superior ability to capture complex patterns and deliver highly accurate predictions.

5. CONCLUSION

The research presents an efficient and intelligent framework for IoT device classification using multiple ML models integrated with a hybrid analytical approach. The system successfully processes structured IoT data through preprocessing, exploratory analysis, and model training to achieve accurate classification results. Comparative evaluation demonstrates that the hybrid DAE-Tree model significantly outperforms traditional models such as GNB, MNB, and DT in terms of accuracy, precision, recall, and F1-score. The incorporation of advanced feature representation through the autoencoder enhances the model's ability to capture complex patterns in the data. The use of ADASYN further improves performance by addressing class imbalance, leading to more reliable predictions. The system also supports real-time and batch prediction, ensuring scalability and practical applicability in dynamic IoT environments. The integration of visualization and performance analysis tools improves interpretability and user understanding. The framework achieves high classification accuracy and demonstrates improved efficiency compared to conventional approaches. This research validates the effectiveness of combining deep learning with interpretable models for enhanced IoT analytics.

REFERENCES

[1]. Ahmad N, Laplante P, DeFranco JF. 2020. Life, IoT, and the pursuit of happiness. IT Professional 22(6):4-7
[2]. Hao Q, Rong Z. 2023. IoTTFID: an incremental IoT device identification model based on traffic fingerprint. IEEE Access 11:58679-58691
[3]. Abomhara M, Køien GM. 2015. Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. Journal of Cyber Security and Mobility 4(1):65-88
[4]. Chakraborty B, Divakaran DM, Nevat I, Peters GW, Gurusamy M. 2021. Cost-aware feature selection for IoT device classification. IEEE Internet of Things Journal 8(14):11052-11064
[5]. Divakaran DM, Singh RP, Sudheera KLK, Gurusamy M, Sachidananda V. 2020. ADROIT: detecting spatio-temporal correlated attack-stages in IoT networks.
[6]. Jmila H, Blanc G, Shahid MR, Lazrag M. 2022. A survey of smart home IoT device classification using ML-based network traffic analysis. IEEE Access 10:97117-97141
[7]. Sanchez PMS, Valero JMJ, Celdran AH, Bovet G, Perez MG, Perez GM. 2021. A survey on device behavior fingerprinting: data sources, techniques, application scenarios, and datasets. IEEE Communications Surveys & Tutorials 23(2):1048-1077
[8]. Sánchez Sánchez PM, Jorquera Valero JM, Huertas Celdrán A, Bovet G, Gil Pérez M, Pérez GM. 2023. A methodology to identify identical single-board computers based on hardware behavior fingerprinting. Journal of Network and Computer Applications 212(2):103579
[9]. Sánchez Sánchez PM, Huertas Celdrán A, Bovet G, Martínez Pérez G. 2024. Adversarial attacks and defenses on ML- and hardware-based IoT device fingerprinting and identification. Future Generation Computer Systems 152(12):30-42
[10]. Meidan Y, Bohadana M, Shabtai A, Guarnizo JD, Ochoa M, Tippenhauer NO, Elovici Y. 201 7. ProfilIoT: a ML approach for IoT device identification based on network traffic analysis.
[11]. Tahaei H, Afifi F, Asemi A, Zaki F, Anuar NB. 2020. The rise of traffic classification in IoT networks: a survey. Journal of Network and Computer Applications 154:102538

[12]. Purmani, S. S. R. (2025). Streamlining IT operations and service management with agile frameworks. European Journal of Advances in Engineering and Technology, 12(4), 76–81.
[13]. Kawai H, Ata S, Nakamura N, Oka I. 2017. Identification of communication devices from analysis of traffic patterns.
[14]. Vasagam, M., Kumar, A., & Garg, A. (2026). Learning Execution Plan Embeddings for Multi-Dimensional Query Resource Prediction. IEEE Access.
[15]. Poojari, R. Frameworks for Data Management and Lineage in Large-Scale Healthcare Data Systems.
[16]. Miettinen M, Marchal S, Hafeez I, Asokan N, Sadeghi A-R, Tarkoma S. 2017. IoT SENTINEL: automated device-type identification for security enforcement in IoT.
[17]. Sun Y, Fu S, Zhang S, Zhu H, Li Y. 2020. Accurate IoT device identification from merely packet length.
[18]. Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P. 2019. Network intrusion detection for IoT security based on learning techniques. IEEE Communications Surveys and Tutorials 21(3):2671-2701
[19]. Yousefnezhad N, Malhi A, Främling K. 2021. Automated IoT device identification based on full packet information using real-time network traffic. Sensors (Basel) 21(8):2660
[20]. Tien CW, Huang TY, Chen PC, Wang JH. 2020. Automatic device identification and anomaly detection with ML techniques in smart factories.
[21]. Zhang L, Gong L, Qian H. 2020. An effective IoT device identification using ML algorithm.