

Cognitive Topology Learning for IoT: Interpretable and Fair Device Classification with Continuous Behavioral Insight

Addanki Raghavendra Chary¹, R. Rajashekar², Rekha Gangula^{3*}, Battu Siddu¹, Banka Akshay Kumar¹, Annarapu Rakshitha¹

¹UG Student, ²Assistant Professor, ³Associate Professor and Head, ^{1,2,3}Department of Computer Science and Engineering (AI&ML)

^{1,2,3}Vaagdevi Engineering College, Bollikunta, Warangal, 506005, Telangana, India

*Correspondence: Rekha Gangula (gangularekha@gmail.com)

Abstract

The rapid proliferation of Internet of Things (IoT) devices in smart environments has introduced significant challenges in identifying and securing heterogeneous network entities. Traditional rule-based classification systems are often inadequate due to the diversity and dynamic behavior of IoT devices. To address this issue, this study proposes a machine learning-driven IoT device classification platform that leverages network traffic fingerprints to accurately categorize devices. The system analyzes behavioral features such as packet frequency, byte count, and communication duration to map devices to their respective profiles. A key challenge in IoT classification is class imbalance, where frequently occurring devices dominate the dataset while rare devices are underrepresented. To mitigate this issue, the Adaptive Synthetic Sampling (ADASYN) technique is employed to generate synthetic samples for minority classes, ensuring balanced and unbiased model training. The proposed framework includes a comparative machine learning pipeline with classifiers such as Gaussian Naive Bayes (GNB), Multinomial Naive Bayes (MNB), Decision Tree Classifier (DTC), and a novel Greedy Tree Classifier (GTC). The GTC model, optimized using interpretable modeling techniques, achieves a balance between high classification accuracy and explainability through rule-based decision structures. Implemented using a modular Flask-based architecture, the platform provides both exploratory data analysis and real-time inference capabilities. Experimental results demonstrate that the GTC model outperforms baseline approaches in terms of F1-score, effectively distinguishing various IoT devices. The system offers a scalable and automated solution for enhancing network management and IoT security.

KeyWords: IoT devices, machine learning, network traffic, security, greedy tree classifier, adaptive synthetic sampling

1. Introduction

The rapid proliferation of Internet of Things (IoT) technologies across diverse sectors has significantly amplified the risk of large-scale cyberattacks. Reports indicate that the number of active IoT device connections worldwide reached approximately 10 billion in 2021 and is expected to surge to nearly 500 billion by 2030 [1]. IoT ecosystems comprise a wide range of interconnected devices capable of communicating and interacting remotely

through network infrastructures. Beyond conventional smart devices such as personal computers and smartphones, IoT extends to traditional non-smart objects that are enhanced with communication capabilities through technological integration.

Despite the substantial benefits offered by IoT systems, their expansion introduces critical security and privacy challenges. Many IoT devices are inherently constrained by limited computational power, low energy consumption,

and minimal storage capacity, making them incapable of supporting advanced security mechanisms [2]. Furthermore, the predominantly wireless nature of IoT networks, coupled with the absence of continuous human supervision, renders these devices highly vulnerable to cyber threats and unauthorized access [3]. As a result, IoT devices often become prime targets for attackers aiming to exploit system vulnerabilities and extract sensitive information.

In this context, identifying the types of devices connected within an IoT network has become a fundamental requirement for enhancing security and mitigating potential threats. Accurate device identification enables network administrators to detect anomalous behavior and isolate compromised devices efficiently. For instance, if a smart bulb exhibits unusual activities such as sending emails or streaming multimedia content instead of performing its intended function of controlling illumination, it can be flagged as suspicious and subjected to further analysis and filtering [5].

To address this challenge, extensive research has focused on leveraging hardware characteristics and device behavioral patterns for IoT device identification [6]. Depending on the application requirements, behavioral-based identification techniques can be broadly categorized into two levels of granularity: (1) individual device model identification and (2) device type identification. The first approach employs techniques such as radio frequency fingerprinting and low-level component analysis to distinguish devices at a granular level. Hardware performance metrics—including CPU, GPU, and memory utilization are analyzed to capture subtle variations arising from manufacturing differences, a method commonly referred to as hardware performance analysis [7].

However, distinguishing devices with identical hardware and software configurations requires

deeper inspection of manufacturing-level variations, which introduces additional challenges. Attackers can manipulate contextual or hardware-level parameters to alter device fingerprints, thereby undermining the reliability of identification systems and leading to potential misclassification. Alternatively, device type identification focuses on categorizing devices based on broader behavioral attributes, such as network traffic patterns and operational processes [8]. This approach enables classification of devices into functional categories (e.g., cameras, smart bulbs) by analyzing their activity profiles, offering a more scalable and robust solution for IoT network security [9].

2. Literature Survey

Yedilkhan et al. [10] proposed a machine learning-based framework for smart home device recognition using network traffic analysis. The method extracted statistical traffic features and applied supervised classification models to identify device types with improved accuracy. However, the approach showed limited robustness against encrypted traffic patterns. Trad et al. [11] proposed a Siamese Neural Network (SNN)-based architecture for IoT device identification by learning similarity metrics between traffic samples. The model compared paired inputs to distinguish device identities with reduced training data requirements. However, the method required complex pairwise training, which increased computational overhead.

Bkheet et al. [12] proposed a comparative framework using Gradient Boosting and Random Forest classifiers for smart home device classification. The study evaluated performance across multiple datasets and highlighted feature importance for improved classification accuracy. Sankar et al. [13] proposed an optimized Variational Autoencoder (VAE) integrated with a Wasserstein Generative Adversarial Network

(WGAN) for IoT device identification. The model generated synthetic data for feature enhancement and improved classification performance through deep generative learning.

Wang et al. [14] proposed a rapid IoT device identification method based on feature reduction and dynamic feature weight assignment. The approach selected optimal features and assigned adaptive weights to enhance classification efficiency and accuracy. Priyadarshini et al. [15] proposed a machine learning-based survivability framework for Industrial Internet of Things (IIoT) systems integrated with smart contracts. The approach ensured secure device communication and improved resilience through automated decision-making mechanisms.

Rekha Gangula et al. [16] proposed a stacked autoencoder-based intrusion detection framework for Internet of Things (IoT) environments. The model extracted hierarchical deep features from network traffic data. A weighted loss function improved class imbalance handling and enhanced detection accuracy. Das et al. [17] proposed a profiling and classification framework for IoT devices in smart home environments using machine learning techniques. The system analyzed behavioral patterns from network traffic to categorize devices effectively. Van Dan et al. [18] proposed an augmentation fusion-based classification method for detecting unauthorized IoT devices using network traffic data. The model combined data augmentation techniques with feature fusion to improve detection accuracy.

Llano-Miraval et al. [19] proposed an Artificial Intelligence (AI)-driven framework for IoT device fingerprinting using Transport Layer Security (TLS) handshake analysis. The approach extracted protocol-level features to classify devices and evaluate defense mechanisms against fingerprinting attacks. Bkheet et al. [20] proposed a deep learning-

based approach using Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) for smart home appliance identification. The model captured temporal dependencies in network traffic sequences to improve classification performance.

Davrazos et al. [21] proposed a meta-ensemble multi-class classifier for IoT device identification using combined outputs from multiple base learners. The framework enhanced classification reliability by aggregating predictions from diverse models. Rekha Gangula et al. [22] proposed an IoT intrusion detection system using an Enhanced Flower Pollination Algorithm with an ensemble classifier. The optimization technique selected optimal features and parameters. The ensemble model improved classification robustness and reduced false alarms. K. Shailaja et al. [23] proposed an intelligent deep feature-based intrusion detection system for network applications. The framework utilized deep learning for feature extraction and classification. The system improved detection accuracy across multiple attack categories.

3. Proposed System

The proposed system as shown in Figure 1 presents an end-to-end IoT device classification and analytics platform designed to process raw network traffic or sensor data and automatically identify IoT device types using machine learning techniques. The architecture follows a modular design integrating data intelligence, machine learning, and a web-based interface within a Flask framework. This unified structure enables scalable data handling, efficient preprocessing, robust model training, and real-time prediction. The system is organized into three major layers: the Data Intelligence Layer, the Machine Learning Pipeline, and the Web Interface Layer, each contributing to a specific phase of the analytical workflow to ensure accurate and interpretable IoT device classification.

Step 1: Data Acquisition and Input

Handling: The process begins with collecting raw IoT data, typically in CSV format, obtained from network traffic logs or sensor-generated datasets. These datasets contain features such as packet size, communication frequency, and transmission duration, which serve as the foundation for device identification.

Step 2: Data Intelligence Layer (Exploratory

Data Analysis): In this stage, the system performs comprehensive EDA to understand the dataset characteristics. Statistical profiling is conducted to compute key metrics such as mean, median, standard deviation, missing values, and memory usage. This helps in detecting anomalies, inconsistencies, or incomplete data. Additionally, correlation analysis is performed to identify relationships among features, enabling the selection of the most relevant attributes for classification. A notable feature is the dynamic labeling mechanism, where unlabeled datasets are handled using a quantile-based activity analyzer that generates synthetic labels based on feature distribution patterns.

Step 3: Data Preprocessing: Before model training, the system preprocesses the data to improve quality and performance. One major issue in IoT datasets is class imbalance, where certain device types dominate. To address this, ADASYN is applied to generate synthetic samples for minority classes. Further preprocessing steps include:

- Feature scaling and normalization to standardize input values
- Label encoding to convert categorical labels into numerical form

These transformations ensure compatibility with machine learning algorithms.

Step 4: Machine Learning Model Training:

The system employs multiple machine learning algorithms to perform classification:

- **GNB:** Suitable for continuous numerical features assuming Gaussian distribution
- **MNB:** Effective for frequency-based features like packet occurrences
- **DTC:** Uses interpretable rule-based structures
- **GTC:** An advanced interpretable model enhancing accuracy and transparency

Each model learns patterns from IoT traffic data to distinguish between device types.

Step 5: Model Evaluation: The trained models are evaluated using standard performance metrics. This comparative analysis helps identify the most effective model for IoT device classification tasks.

Step 6: Model Persistence: To improve efficiency, the system saves trained models and preprocessing components (scaler, encoder) as serialized .pkl files. This eliminates the need for retraining and enables quick loading for future predictions, reducing computational overhead and response time.

Step 7: Web Interface Layer (User

Interaction): The final layer provides an interactive web-based interface developed using Flask. It includes:

- **EDA Dashboard:** Visualizes dataset statistics, distributions, and correlations
- **Performance Suite:** Displays model comparison metrics in a leaderboard format
- **Prediction Center:** Enables real-time classification

Users can perform:

- **Batch Prediction:** Upload large datasets for bulk classification

- **Single Prediction:** Input feature values manually for instant results

Step 8: Real-Time IoT Device Classification:

Using the deployed models, the system performs real-time prediction of IoT device types, generating outputs either as annotated datasets or instant classification results, making the platform practical for real-world IoT monitoring and security applications.

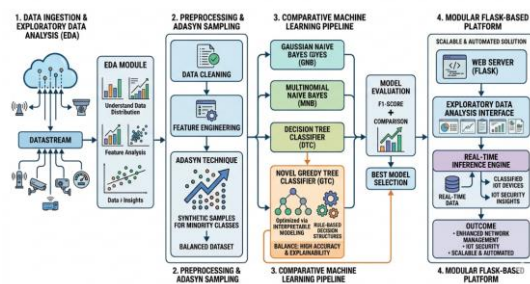


Figure. 1. Proposed system architecture of IoT devices classification.

3.1 Inference Engine

The inference engine as shown in Figure 2 is the specialized component of the system responsible for executing the pre-trained models to generate predictions on new, unseen IoT data. Unlike the training phase, which is computationally heavy, the inference engine is optimized for speed and reliability. It acts as a "Read-Only" environment that pulls the stored "intelligence" (saved .pkl files) and applies it to live inputs. This engine ensures consistency by using the exact same scaling and encoding parameters used during the original training phase, guaranteeing that a "Smart Camera" is identified correctly every time, as shown in figure 2. Initially, the GTC serves as a primary driver for the inference engine, especially when interpretability and speed are prioritized. Because GTC models (via imodels) are designed to be "shallow but sharp," they allow the inference engine to make decisions using a minimal set of logical rules, which is ideal for low-latency IoT environments. The inference

flow for the GTC model follows a "Path of Least Resistance" logic to reach a conclusion:

- **Request Reception:** The engine receives a feature set (e.g., packet frequency, byte counts) from the Flask UI.
- **State Restoration:** It loads the specific `greedytree_model.pkl` and the corresponding `scaler.pkl` to ensure the input data is normalized to the training environment's standards.
- **Greedy Rule Application:** The engine traverses the tree by evaluating the input against the most "informative" rules first (the greedy approach).
- **Threshold Validation:** At each node, the engine checks if the IoT feature exceeds the specific threshold optimized during the training phase.
- **Category Assignment:** Once a leaf node is reached, the engine assigns the final device category.
- **Confidence Assessment:** It calculates the class distribution at that specific leaf to provide a probability score for the prediction.

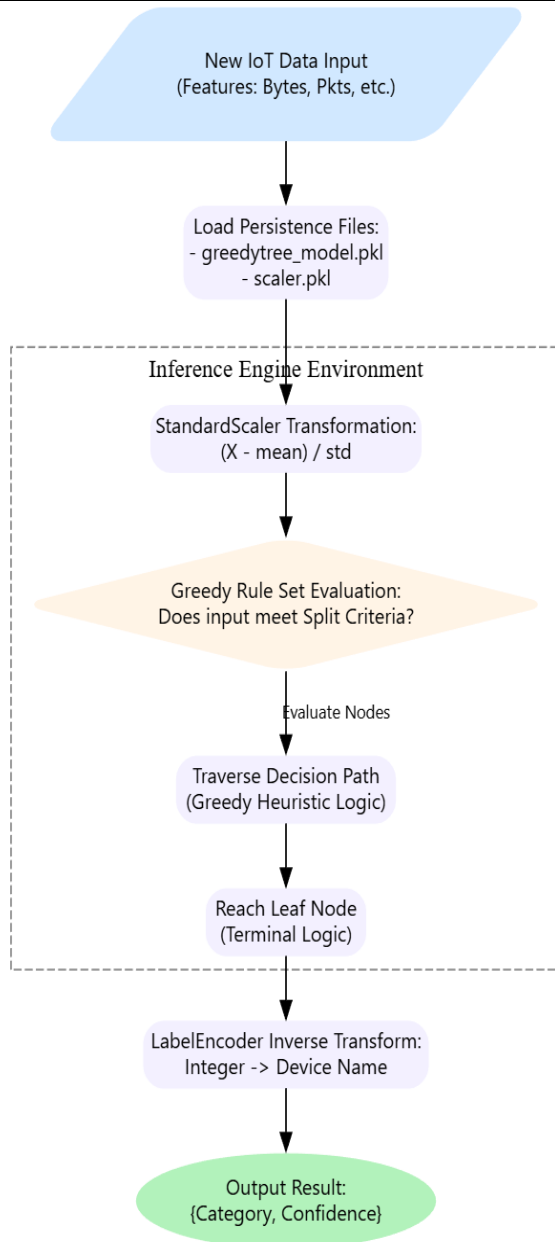


Figure. 2: Inference engine of proposed IoT device classification system.

4. RESULTS DESCRIPTION

Figure 3 presents the confusion matrix of the proposed GTC, illustrating the classification performance across different IoT device classes. The matrix shows a strong concentration of values along the diagonal, indicating that most instances are correctly classified. Misclassifications are minimal and sparsely distributed in off-diagonal elements, demonstrating the model's robustness in

distinguishing between device categories. This aligns with the high-performance metrics achieved by the GTC model, including an accuracy of 97.08%, precision of 97.17%, recall of 97.08%, and F-score of 97.07%, confirming that the model produces highly reliable predictions with very low error rates.

Figure 4 illustrates the Receiver Operating Characteristic (ROC) curve of the proposed GTC model, which evaluates its ability to discriminate between classes. The curve is positioned very close to the top-left corner, indicating a high True Positive Rate (TPR) and a low False Positive Rate (FPR). This suggests excellent classification capability across all thresholds. The Area Under the Curve (AUC) is expected to be close to 1.0, reflecting near-perfect separability. These observations are consistent with the reported performance values, where the GTC model achieves 97.08% accuracy, 97.17% precision, and 97.08% recall, confirming superior detection capability compared to other models.

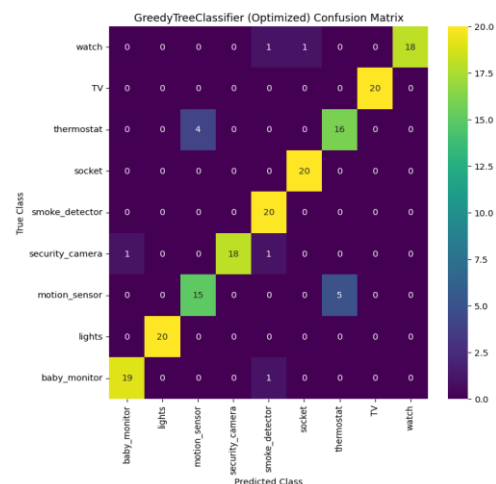


Figure. 3. Confusion matrix obtained using the proposed GTC model.

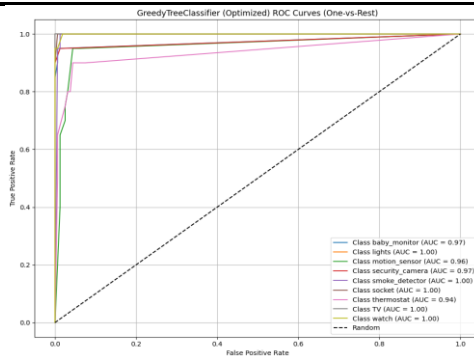


Figure. 4. ROC curve obtained for the proposed GTC model.

Figure 5 provides a comparative performance analysis of the existing models (DTC, MNB, GNB) and the proposed GTC model using key evaluation metrics. The DTC model shows very poor performance with 18.89% accuracy, 10.32% precision, 18.89% recall, and 11.14% F-score, indicating ineffective classification. The MNB classifier demonstrates moderate performance with 76.11% accuracy, 79.88% precision, 76.11% recall, and 76.38% F-score. The GNB classifier performs better, achieving 80.14% accuracy, 87.36% precision, 80.14% recall, and 78.72% F-score, reflecting improved predictive capability. However, the proposed GTC model significantly outperforms all existing models, achieving 97.08% accuracy, 97.17% precision, 97.08% recall, and 97.07% F-score, clearly demonstrating its effectiveness and superiority for IoT device classification.

Algorithm	Status	Accuracy	Precision	Recall	F1 Score	Performance Grade
GaussianNB	Link	0.8014	0.8736	0.8014	0.7872	Good
MultiModelNB	Link	0.7611	0.7988	0.7611	0.7638	Fair
DecisionTree	Link	0.1889	0.1032	0.1889	0.1714	Poor
GreedyGre	Link	0.9778	0.9781	0.9778	0.9777	Excellent

Figure. 5: Detailed performance evaluation of existing and proposed models.

Figure 6 demonstrates the functional output of the system's inference engine, showing how the trained GTC model handles a live classification request.

- **Input-to-Output Mapping:** It displays a single prediction event where raw numeric features (the "fingerprint" of the device) are transformed into a human-readable result.
- **Detailed Inference Results:**
 - **Predicted Category:** The final label assigned to the device (e.g., "Smart Camera," "Thermostat"), which is the result of the LabelEncoder decoding the model's internal numeric prediction.
 - **Confidence Score:** Often expressed as a percentage or a decimal (e.g., 98% or 0.985), this indicates the mathematical certainty of the GTC model. A high score here validates that the device's traffic closely matches the specific greedy rules generated during training.
- **System Validation:** In the web interface, this output usually appears in a success notification, confirming that the engine has successfully identified the hardware profile from the network stream.

<code>domain_in</code>	<code>domain_in_suffix_in_1</code>	<code>suffix_in_1</code>	<code>domain_in_suffix_in_2</code>	<code>suffix_in_2</code>	<code>domain_in_suffix_in_3</code>	<code>suffix_in_3</code>	<code>domain_in_suffix_in_4</code>	<code>suffix_in_4</code>	<code>predicted_category</code>
0	0	0	0	0	0	1	0	0	@ baby_monitor
0	0	0	0	0	0	1	0	0	@ baby_monitor
0	0	0	0	0	1	0	0	0	@ baby_monitor
0	0	0	0	0	0	0	1	0	@ TV
0	0	0	0	0	0	0	1	0	@ TV
0	0	0	0	0	0	1	1	0	@ TV
0	0	0	0	0	0	0	1	0	@ TV
0	0	0	0	0	0	0	1	0	@ TV
0	0	0	0	0	0	0	1	0	@ TV
0	0	0	0	0	0	0	1	0	@ TV
0	0	0	0	0	0	0	1	0	@ socket
0	0	0	0	0	0	0	1	0	@ socket
0	0	0	0	0	0	0	1	0	@ socket
0	0	0	0	0	0	0	1	0	@ lights
0	0	0	0	0	0	0	1	0	@ thermostat
0	0	0	0	0	0	0	1	0	@ thermostat
0	0	0	0	0	0	0	1	0	@ thermostat
0	0	0	0	0	0	0	1	0	@ TV
0	0	0	0	0	0	0	1	0	@ TV
0	0	0	0	0	0	0	1	0	@ TV
0	0	0	0	0	0	0	1	0	@ watch
0	0	0	0	0	0	0	1	0	@ watch
0	0	0	0	0	0	0	1	0	@ watch
0	0	0	0	0	0	0	1	0	@ lights
0	0	0	0	0	0	0	1	0	@ lights
0	0	0	0	0	0	0	1	0	@ lights
0	0	0	0	0	0	0	1	0	@ motion_sensor

Figure. 6: Prediction obtained using proposed GTC model.

5. Conclusion

The research effectively demonstrates the potential of machine learning in enabling automated IoT device classification by moving

beyond traditional static identifiers such as MAC addresses toward dynamic behavioral fingerprinting. This transition enhances the robustness of network security by capturing real-time communication patterns rather than relying on easily spoofed identifiers. A key contribution of the system is the integration of ADASYN, which addresses the challenge of class imbalance commonly present in IoT datasets. By generating synthetic samples for underrepresented device classes, the approach ensures that the models achieve consistently high precision and balanced performance across all categories. Among the evaluated models, the GTC proves to be the most effective, offering an optimal balance between high predictive accuracy and interpretability. Its capability to produce concise yet highly discriminative decision rules allow network administrators to clearly understand the reasoning behind each classification, which is essential for security auditing and trust. Furthermore, the inclusion of a modular Flask-based web interface significantly enhances usability by translating complex backend computations into an intuitive and interactive dashboard. This enables users to perform exploratory data analysis, compare model performance, and execute real-time as well as batch predictions with ease. So, the proposed platform delivers a scalable, efficient, and interpretable solution for IoT device management. It minimizes manual intervention, improves classification reliability, and supports the development of autonomous and secure smart environments.

References

- [1]. M. L. Hoang, "A Review of Developments and Metrology in Machine Learning and Deep Learning for Wearable IoT Devices," in *IEEE Access*, vol. 13, pp. 106035-106054, 2025, doi: 10.1109/ACCESS.2025.3573937.
- [2]. Ntayagabiri, Jean Pierre, Youssef Bentaleb, Jeremie Ndikumagenge, and Hind El Makhtout. "A Comparative Analysis of Supervised Machine Learning Algorithms for IoT Attack Detection and Classification." *Journal of Computing Theories and Applications* 2, no. 3 (2025): 395-409.
- [3]. Luqman, Muhammad, Muhammad Zeeshan, Qaiser Riaz, Mehdi Hussain, Hasan Tahir, Noman Mazhar, and Muhammad Saffeer Khan. "Intelligent parameter-based in-network IDS for IoT using UNSW-NB15 and BoT-IoT datasets." *Journal of the Franklin Institute* 362, no. 1 (2025): 107440.
- [4]. Cao, Zhong, Zhicai Zhao, Wenli Shang, Shan Ai, and Shen Shen. "Using the ToN-IoT dataset to develop a new intrusion detection system for industrial IoT devices." *Multimedia Tools and Applications* 84, no. 16 (2025): 16425-16453.
- [5]. Abbasi, Fereshteh, Marjan Naderan, and Seyed Enayatallah Alavi. "Dimensionality reduction with deep learning classification for botnet detection in the Internet of Things." *Expert Systems with Applications* 267 (2025): 126149.
- [6]. Çetin, Aziz, and Sıtkı Öztürk. "Comprehensive exploration of ensemble machine learning techniques for IoT cybersecurity across multi-class and binary classification tasks." *Journal of Future Artificial Intelligence and Technologies* 1, no. 4 (2025): 371-384.
- [7]. Sheelavant, Kumares, Charan KV, B. Yamini Supriya, Purshottam J. Assudani, Chandra Bhushan Mahato, and Sanjay Kumar Suman. "Ensemble Learning-Based Intrusion Detection and Classification for Securing IoT

- Networks: An Optimized Strategy for Threat Detection and Prevention." *Journal of Intelligent Systems & Internet of Things* 17, no. 2 (2025).
- [8]. Rabbani, Mahdi, Jinkun Gui, Zeming Zhou, Fatemeh Nejati, Mansur Mirani, Gunjan Piya, Igor Opushnyev, Rongxing Lu, and Ali Ghorbani. "A lightweight IoT device identification using enhanced behavioral-based features." *Peer-to-Peer Networking and Applications* 18, no. 2 (2025): 2.
- [9]. Rahman, Md Mizanur, Faycal Bouhafs, Sayed Amir Hoseini, and Frank den Hartog. "UNSW HomeNet: A network traffic flow dataset for AI-based smart home device classification." *Computers & Industrial Engineering* 204 (2025): 111041.
- [10]. Yedilkhan, Didar, and Saida Smakova. "Machine Learning Approaches for Smart Home Device Recognition from Network Traffic." *Procedia Computer Science* 231 (2024): 709-714.
- [11]. Trad, Fouad, Ali Hussein, and Ali Chehab. "Using siamese neural networks for efficient and accurate IoT device identification." In *2022 Seventh International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 1-7. IEEE, 2022.
- [12]. Bkheet, Sana Abdelaziz, Gamal Saad Mohamed Khamis, Abdulaziz Alenazi, Wiam Abdelrahman Almalih, Mnahil M. Bashier, and Zakariya MS Mohammed. "Comparative Performance of Gradient Boosting and Random Forest for Smart Home Device Classification." (2025).
- [13]. Sankar, Jothi Shri, Saravanan Dhatchnamurthy, and Keerat Kumar Gupta. "Bolstering IoT security with IoT device type identification using optimized variational autoencoder Wasserstein generative adversarial network." *Network: Computation in Neural Systems* 35, no. 3 (2024): 278-299.
- [14]. Wang, Xiuting, Xiangyang Luo, Shaoyong Du, Lingling Li, Yang Yang, and Fenlin Liu. "A rapid device type identification method based on feature reduction and dynamical feature weights assignment." In *International Conference on Artificial Intelligence and Security*, pp. 663-677. Cham: Springer International Publishing, 2022.
- [15]. Priyadarshini, Ishaani, Raghvendra Kumar, Ahmed Alkhayyat, Rohit Sharma, Kusum Yadav, Lulwah M. Alkwai, and Sachin Kumar. "Survivability of industrial internet of things using machine learning and smart contracts." *Computers and Electrical Engineering* 107 (2023): 108617.
- [16]. Rekha Gangula, V. Murali Mohan, Ranjeeth Kumar. M, "Stacked autoencoder with weighted loss function for intrusion detection in IoT application". *Multimed Tools Appl* (2024).
<https://doi.org/10.1007/s11042-024-19962-7>.
- [17]. Das, Sudhir Kumar, Sujit Bebortha, Bibudhendu Pati, Chhabi Rani Panigrahi, and Dilip Senapati. "Profiling and classification of IoT devices for smart home environments." In *Machine Learning for Cyber Physical System: Advances and Challenges*, pp. 85-121. Cham: Springer Nature Switzerland, 2024.
- [18]. Van Dan, Duong, and Ta Minh Thanh. "Unauthorized iot devices detection based on network traffic using

- augmentation fusion classification method: DV Dan, TM Thanh." International Journal of Information Security 24, no. 6 (2025): 225.
- [19]. J. D. Llano-Miraval, C. Campo, C. Garcia-Rubio and M. Moure-Garrido, "AI Versus IoT Security: Fingerprinting and Defenses Against TLS Handshake-Based IoT Device Classification," in IEEE Access, vol. 13, pp. 165607-165622, 2025, doi: 10.1109/ACCESS.2025.3611160.
- [20]. Bkheet, Sana Abdelaziz, Johnson I. Agbinya, and Gamal Saad Mohamed Khamis. "Advanced Deep Learning Approach for Smart Home Appliance Identification Using Recurrent Neural Networks with LSTM." IoT 5, no. 4 (2024): 835-851.
- [21]. Davrazos, Gregory, Theodor Panagiotakopoulos, Sotiris Kotsiantis, and Achilles Kameas. "IoT device identification using a meta-ensemble multi-class classifier." In 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), pp. 1-4. IEEE, 2023.
- [22]. Rekha Gangula, V. Murali Mohan, Ranjeeth Kumar. M, "Network Intrusion Detection System for Internet of Things based on Enhanced Flower Pollination Algorithm and Ensemble Classifier", Concurrency and Computation: Practice and Experience – Wiley Online Library, Accepted on 05 April 2022, ISSN: 1532-0634 (online), ISSN: 1532-0626(print), Impact Factor: 1.536. Standard age of Journal - 21 Yrs. (since 2001).
- [23]. K Shailaja, B Srinivasulu, Lingala Thirupathi, Rekha Gangula, Thejoramnaresh Reddy Boya, Vemulamma Polem , "An Intelligent Deep Feature Based Intrusion Detection System for Network Applications". Wireless Pers Commun 129, 345–370 (2023). <https://doi.org/10.1007/s11277-022-10100-w>