IOT Device Type Identification Using Network Traffic

Mohammad Ali Asaad Wadi AL-Saeedi
Arts, Sciences & Technology University / Lebanon (AUL)
Department of Computer and Communications Engineering
E_mail:Moh.Ali.Asaad@gmail.com

Abstract

The exponential growth of Internet of Things (IOT) devices has introduced complexity into network traffic management and security. This study investigates the efficacy of machine learning techniques in classifying diverse IOT device types based on their network traffic patterns. A comprehensive dataset comprising traffic data from various IoT devices is collected and analyzed. Supervised and unsupervised machine learning models are then deployed for classification, leveraging feature extraction and selection methods. Results reveal promising accuracy rates, with XGBoost achieving the highest accuracy at 91.2%, closely followed by Random Forest at 90.9%. The Decision Tree algorithm also performs well, with an accuracy score of 90%, and the Gradient Boosting algorithm has an accuracy of 87%. These findings underscore the potential of machine learning in bolstering network security, optimizing resource allocation, and enhancing IOT ecosystem management.

Keywords: Internet of Things (IOT), Network Traffic Patterns, Machine Learning Techniques, Device Classification And Network Security.

تحديد نوع جهاز IOT باستخدام حركة مرور الشبكة

محمد علي أسعد وادي السعيدي
جامعة الآداب والعلوم والتكنولوجيا – قسم هندسة الحاسوب والاتصالات(AUL) / لبنان

الخلاصة

أدى النمو الهائل لأجهزة إنترنت الأشياء (IOT) إلى تعقيد إدارة حركة مرور الشبكة وأمنها. تبحث هذه الدراسة في مدى فعالية تقنيات التعلم الآلي في تصنيف أنواع أجهزة إنترنت الأشياء المتنوعة بناءً على أنماط حركة مرور الشبكة الخاصة بها. يتم جمع وتحليل مجموعة بيانات شاملة تضم بيانات حركة المرور من أجهزة إنترنت الأشياء المختلفة. يتم بعد ذلك نشر نماذج التعلم الآلي الخاضعة للإشراف وغير الخاضعة للإشراف للتصنيف، والاستفادة من أساليب استخلاص الميزات واختيارها. تكشف النتائج عن معدلات دقة واعدة، حيث حقق XGBoost أعلى دقة بنسبة ٩١.٢٪، تليها Random Forest بنسبة ٩٠.٩%. تعمل خوارزمية Decision Tree أيضًا بشكل جيد، حيث تبلغ درجة الدقة ٩٠٪، وتبلغ دقة خوارزمية Gradient Boosting ٨٧%. تؤكد هذه النتائج على إمكانات التعلم الآلي في تعزيز أمن الشبكات، وتحسين تخصيص الموارد، وتعزيز إدارة النظام البيئي لإنترنت الأشياء.

الكلمات المفتاحية: إنترنت الأشياء (IOT)، أنماط حركة مرور الشبكة، تقنيات التعلم الآلي، تصنيف الأجهزة وأمن الشبكات.

Introduction

The Internet of Things (IOT) has revolutionized the way we interact with the physical world, seamlessly integrating smart devices into our daily lives, cities, and work environments. This integration spans from smart home appliances to sophisticated sensors in smart cities and various corporate networks. While IOT devices significantly enhance the quality of life and operational efficiency, their proliferation introduces complex challenges in network management and security. This paper delves into the nuances of IoT device identification, a pivotal aspect of managing the ever expanding IOT ecosystem (koohang,*et al.*,2022).

IOT devices are defined by their ability to perform specific tasks autonomously, without human intervention. Examples include security cameras, smart thermostats and smart lighting systems, which contrast with traditional computing devices such as desktop PCs, laptops and smartphones that require direct user commands to operate. The distinction is crucial, as the operational characteristics and network behavior of IoT devices differ significantly from non- IOT devices, influencing their identification and management within networks (Al-Turjman,*et al,*.2022).

The diversity of IOT devices, produced by a myriad of manufacturers, presents a unique set of challenges in monitoring and managing network traffic. Each device type generates distinct traffic patterns, complicating the task of identifying and classifying them accurately.

This complexity is compounded by the presence of both IOT and non- IOT devices within the same network, necessitating sophisticated identification methods that can accurately differentiate between them. Furthermore, the asymmetrical data flow from these devices, where certain devices like security cameras generate more traffic than others like smart plugs-adds another layer of complexity to device identification and network management.

Several methods have been developed to identify IOT devices, focusing primarily on analyzing network traffic to discern device types. However, these methods often fall short in real-world scenarios. Many rely exclusively on the presence of IOT devices, neglecting the mixed-device environments typical in actual networks. Additionally, the variable nature of data flow from different devices and the imbalance in traffic volumes are frequently overlooked. Previous approaches have also leaned heavily on machine-learning techniques that utilize exhaustive lists of traffic characteristics without adequately explaining how specific features contribute to the identification process's effectiveness. This lack of transparency and adaptability limits the utility of current methods in diverse and evolving network contexts (Rondon,*et al,*.2022).

The omnipresence of IOT devices, from personal wearables to industrial sensors, underlines the need for robust network management and security mechanisms. Efficiently identifying the types of IOT devices within a network is essential for optimizing network performance, enhancing security, and ensuring compliance with regulatory standards. This research is motivated by the pressing need to overcome the limitations of existing identification methods and to develop a more accurate, flexible approach to IOT device identification based on network traffic analysis.

The paper's structure is delineated as

follows: Section 2 conducts a review of pertinent literature in the field, offering valuable insights into existing research. Section 3 elaborates on the methodology employed and details the datasets used for the study. Section 4 presents the results obtained from the research efforts, providing a comprehensive exploration of the outcomes. Finally, Section 5 concludes the study by summarizing key observations, discussing their implications, and suggesting potential avenues for future research.

Related Work

Based on the literature in (Miettinen,*et al.*,2017), device identification can leverage various machine learning (ML) techniques, including supervised, unsupervised, semi-supervised, and deep learning methods. Studies (Hamad,*et al.*,2019) and (Wang,*et al.*,2022) have applied supervised ML techniques, with (Hamad, *et al.*,2019) detailing a method to recognize common IOT devices through network traffic feature extraction using a tool developed in (Wang,*et al.*,2022). A two-phase meta classifier approach was explored in (Sivanathan,*et al.*,2019) where the initial classifier distinguishes between IOT and non- IOT devices, followed by a second classifier that specifies the IOT device type. The research in (Meidan,*et al.*,2017) considers using the Random Forest classifier, among other models like Decision Trees, Logistic Regression, SVM, GBM and XGBoost, demonstrating high precision in IoT device identification. Yet, supervised ML necessitates labeled data for model training, which could be challenging or costly to acquire.

Unsupervised learning for classifying IOT device types from network traffic data was employed in (Sivanathan,*et al.*,2019) segmenting traffic into packet flows ranging from 1 to 8 minutes. K-Means clustering was used, with the number of clusters set based on the device. The study in (Marchal,*et al*,.2019) adopts a heuristic to identify data transmission cycles, using device naming conventions and K-Nearest Neighbors for clustering, albeit at a slower pace compared to other methods.

Deep learning approaches were used in (Sivanathan,*et al.*,2019) and (Marchal,*et al.*,2019) for classifying unknown devices by integrating ML autoencoders with clustering techniques. While (Marchal,*et al.*,2019) analyzed packet statistics for compromised device detection, (Bhatia,*et al.*,2019) applied variational autoencoders for device identification by combining periodic patterns with flow statistics.

(Miettinen,*et al.*,2017) introduced a device fingerprinting (DFP) method analyzing 23 features from 12 network packets, achieving an 81.50% recognition accuracy across 27 devices from the IOT Sentinel dataset (Miettinen,*et al.*,2017). Another DFP technique (Aksoy,*et al.*,2019) utilized a genetic algorithm to select significant features for fingerprinting, attaining over 95% accuracy for device genre and 82% for individual device types using the same dataset.

A packet header information-based fingerprinting method was developed to identify devices by their unique feature sets, employing ML classification to categorize IOT devices into types and categories. J48 and PART algorithms were notably effective, with precision levels adjustable based on user preferences. This method achieved high classification accuracy rates using both the IOT Sentinel dataset (Miettinen,*et al.*,2017) and the UNSW dataset (Miettinen,*et al*,.2017).

IOT Sense by (Bezawada,*et al.*,2018) and a large-scale IOT device categorization study by (Sivanathan,*et al.*,2017) further highlight the efficacy of feature-based identification methods, with IOT Sense achieving up to 99% accuracy.

(Meidan,,*et al.*,2017) proposed a method for detecting unauthorized IOT devices with a 96% success rate, though it depends on application layer data, which is often encrypted. (Shahid,*et al.*,2018) achieved 99.9% accuracy in classifying devices based on bidirectional flow characteristics introduced an automated identification system using a CNN + BiLSTM model, surpassing conventional methods with over 99% accuracy.

This study utilized the ping operation to fingerprint IoT devices, achieving high detection rates with a minimal number of pings. (Oser,*et al.*,2018) measured clock offsets using TCP timestamps to identify device types with high accuracy by combining it with other features.

(Thangavelu,*et al.*,2018) suggested DEFT, a fingerprinting technique using SDN, demonstrating the potential of clustering and random forest for identifying unknown devices. Similar advancements in IOT device model categorization and event identification, which classify IoT communications with high accuracy using packet header analysis.

Methodology

In this section, we present the dataset employed, describe the adopted system model, and detail the systematic approach to data preprocessing that was followed.

A. System Model

The system model depicted in Figure 1 outlines a structured approach for developing a machine learning model, which is organized into a series of sequential steps, each building upon the last to achieve a refined outcome.

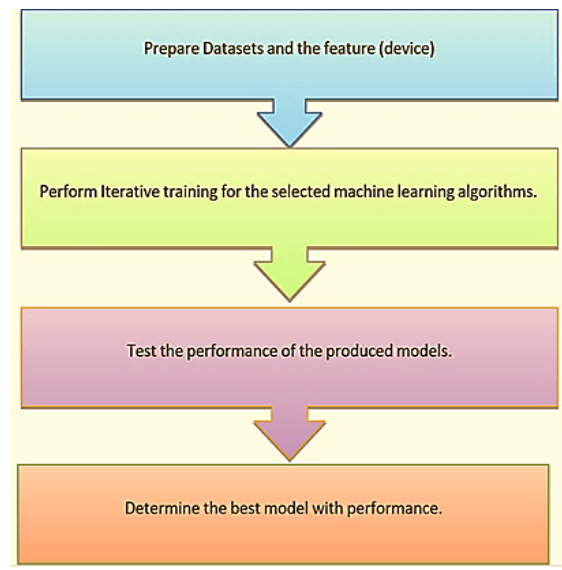

Fig.(1) A System Model for Machine Learning Implementation

In the first step, ’Prepare Datasets and the Feature,’ the process begins with the collection and preparation of datasets that will be used for training the machine learning model. The data needs to be cleaned to remove any inaccuracies or irrelevant information, ensuring that only quality data is used. Once the dataset is deemed clean, the next task is feature extraction. This is a critical stage where raw data is transformed into a set of meaningful features that will effectively represent the underlying patterns and characteristics relevant to the problem at hand. These features should capture the essence of the data and be in a format that is suitable for the machine learning algorithms to process.

Following the preparation phase is ’Perform Iterative Training for the Selected Machine Learning.’ During this stage, the cleaned and structured dataset

is fed into a machine learning algorithm. Training the model is an iterative process, meaning that the algorithm will go through multiple rounds of learning from the data. With each cycle, the algorithm adjusts its internal parameters in an attempt to minimize errors and improve its predictive accuracy. The iterative nature of this process is essential for fine-tuning the model and ensuring that it captures the nuances of the data effectively.

The third step is to 'Test the Performance of the Produced.' After training, the model must be evaluated to determine how well it performs. This typically involves using a test set—a portion of the data that was held back during the training phase and not used for learning. The model's predictions on this unseen data are compared to the actual outcomes to assess its performance. This step is crucial for validating the model's ability to generalize beyond the data it was trained on.

Performance metrics such as accuracy, precision, and recall are calculated to provide a quantitative measure of the model's predictive power.

Lastly, 'Determine the Best Model with Performance' involves selecting the optimal machine learning model based on the performance metrics obtained from the testing phase. If multiple models or various configurations of a model were tested, they would be compared against each other. The selection criteria will include not just accuracy but also how well the model generalizes to new data, its computational efficiency, and potentially its interpretability. The best model is the one that performs the best on the test data and meets the project requirements, thereby making it the preferred choice for deployment in real-world applications.

B. Dataset

Building a robust Internet of Things (IOT) device identification system is a complex task that demands a meticulous approach to data collection, preprocessing, feature extraction, and feature processing. The foundation of such a system lies in the assembly of a comprehensive and diverse dataset, which is critical for the accurate classification of IOT devices across various environments like smart homes, healthcare facilities, and industrial settings. This process encompasses several crucial steps, each playing a pivotal role in ensuring the quality and effectiveness of the identification system.

The initial step in creating a reliable IOT device identification system involves the selection of an appropriate dataset. The goal here is to capture a broad spectrum of network traffic data emanating from a wide array of IoT devices. This diversity is essential to train the system to recognize and differentiate between numerous device types, functioning in varied environments. Once the dataset is chosen, the next step is data preprocessing, a phase where raw data is refined and prepared for analysis. This step includes cleaning the data to remove any noise or irrelevant information that might skew the analysis or lead to inaccurate classification.

Data labeling is another critical process where each network trace is tagged with a label that identifies the IoT device type. This step can be performed manually or by leveraging existing datasets that have been pre-categorized. Such meticulous labeling is indispensable for training the system with high precision.

Following data preparation, the system

moves onto the feature extraction phase. This stage is about distilling raw network traffic data into a set of meaningful features that reflect the unique behaviors and communication patterns of IoT devices. It involves the selection of network traffic features, including packet length features, inter-arrival time features, protocol usage, payload characteristics, and flow features. These features are carefully chosen to highlight aspects of the data that are most telling of the device types, such as the distribution and average length of packets, the variance in packet lengths, and the specific protocols employed by the devices.

Moreover, time-series analysis is employed to uncover temporal patterns in the data, offering insights into the dynamic behaviors of IoT devices over time. Techniques like periodogram analysis and autocorrelation functions are utilized to identify periodic components and assess similarities in traffic patterns at different times.

The high-dimensional feature space resulting from feature extraction poses its own challenges, addressed through dimensionality reduction techniques such as Principal Component Analysis (PCA). These techniques help in simplifying the data without losing critical information, making the dataset more manageable for analysis. Additionally, feature scaling and normalization are applied to ensure that all features contribute equally to the model, preventing any single attribute from overpowering the rest during the model training phase.

Simulations and Results

In this section, we present and analyze the algorithms devised for our study, which are fundamental for detecting attacks in IOT networks. The algorithms outlined here aim to enhance the precision and effectiveness of attack detection, enabling a deeper understanding and interpretation of security threats in IOT environments.

A- Random Forest

Data scientists utilize a diverse range of machine learning algorithms to extract patterns from extensive datasets, providing valuable insights for their Random Forest is a powerful ensemble learning technique that aggregates the predictions of multiple decision trees to enhance predictive accuracy and mitigate overfitting. As a bagging algorithm, it generates various bootstrap samples from training data, constructing a decision tree for each sample. Uniquely, it randomly selects a subset of features when building each tree, which decorrelates the trees and enhances the model's generalizability.

The versatility of Random Forest is evident in its efficacy for both classifications, where it selects the class with the majority vote among trees, and regression, where it averages the outcomes for a prediction. Additionally, Random Forest inherently assesses feature importance, providing valuable insights for feature selection and data analysis. With its dual ability to handle diverse data types and provide reliable predictions, Random Forest is a valuable tool in the machine learning toolkit.

B- Decision trees

Decision trees stand as a fundamental component in machine learning, valued for their straightforward decision-making process and transparent nature. They're utilized for both classification and regression tasks across a myriad of fields such as healthcare, finance, and beyond. These hierarchical models work by dividing data recursively based on

feature values into subsets that form the basis of decision rules, leading to homogeneous groups for precise predictions. Each decision tree comprises nodes representing decision points and leaf nodes indicating outcomes, making the decision process easy to follow and understand.

Transparency is a principal benefit of decision trees, allowing users to visually follow the model's decision-making from root to leaf nodes, which is particularly important in fields requiring transparent rationale, like medical diagnostics. However, decision trees are prone to overfitting, capturing noise instead of genuine data patterns if they grow too complex. Strategies like pruning and setting maximum depths are employed to prevent this, ensuring the tree models the essential data characteristics more generally. Moreover, decision trees' sensitivity to data variations can cause instability, but this is often rectified by using ensemble methods such as Random Forests and Gradient Boosting, which combine multiple trees to enhance stability and predictive power.

In essence, decision trees are a vital and interpretable tool in the machine learning arsenal. They offer clear, comprehensible decision-making paths and are particularly valuable in domains where the logic behind predictions must be clear. While they have a tendency to overfit, various techniques and ensemble approaches have been developed to leverage their strengths and alleviate their limitations, securing their position as a versatile choice for machine learning applications.

C- Gradient boosting

Gradient Boosting is a prominent ensemble machine learning technique recognized for its predictive precision and versatility in a broad spectrum of applications. As a boosting algorithm, it constructs a series of weak learners, usually decision trees, and combines them sequentially to form a more accurate and robust model.

The essence of Gradient Boosting lies in its strategy to continuously enhance predictions by fitting new trees to the residuals or errors of preceding trees. Each subsequent tree aims to correct the mistakes of the former, thereby incrementally reducing error rates. This process targets the minimization of a loss function, typically employing a gradient descent approach, which is the basis for the method's nomenclature.

A remarkable advantage of Gradient Boosting is its ability to manage different types of data and address both regression and classification problems. It's flexible enough to work with various loss functions tailored to the specific requirements of the task, enabling its application across diverse scenarios like financial forecasting, fraud detection, and language processing.

Within the Gradient Boosting family, algorithms such as XGBoost, LightGBM, and CatBoost have become favorites due to their efficient and advanced implementations. These versions are optimized with features that support large datasets and offer improved computation through parallelization and regularization while providing tools for fine-tuning model parameters. Nonetheless, the computational demands of Gradient Boosting and the necessity for careful tuning should be acknowledged.

D- Extreme Gradient Boosting (XGBoost)

Friedman (Oser,*et al.*,2018) introduced a variation of the gradient tree boosting known as Extreme Gradient

Boosting (XGBoost). The goal of the tree ensemble boosting approach known as "gradient tree boosting" is to take a collection of relatively weak classifiers and merge them into a single, robust one.

Beginning with a weak learner, an advanced learner is educated in an iterative fashion (Wang,*et al.*,2022). Similar principles underlie both gradient boosting and XGBoost. It's in the finer points of how they're implemented that the two systems diverge significantly. By using a variety of regularization methods on the trees, XGBoost is able to obtain greater performance .

Results and discussion

The Random Forest classifier demonstrates robust performance in classifying various IOT devices, in Table(1), with TVs, lights, and security cameras achieving notably high precision, recall, and F1-scores close to 0.95.

This indicates a strong predictive ability with few false positives or negatives for these device types. In contrast, smoke detectors and water sensors present a challenge, with their scores approximately 0.75 and 0.65 respectively, signaling potential areas for model refinement. While the classifier generally excels, the lower scores for some devices suggest that model adjustments or enriched training data might be necessary to enhance accuracy consistently across all device types.

Table (1) Estimated Performance Metrics of IoT Device Classification for Random Forest

Device Type	Precision	Recall	F1-Score
TV	0.95	0.95	0.95
Baby Monitor	0.90	0.85	0.88
Lights	0.95	0.95	0.95
Motion Sensor	0.85	0.80	0.80
Security Camera	0.90	0.90	0.90
Smoke Detector	0.75	0.65	0.70
Socket	0.80	0.80	0.80
Thermostat	0.85	0.85	0.85
Watch	0.95	0.90	0.90
Water Sensor	0.70	0.60	0.65

Table (2) shows the performance of XGBoost classifier, it yields high precision, recall, and F1-scores for TVs, baby monitors, and lights, indicating excellent model performance for these device types with scores nearing or at 0.95. The motion sensor and socket have moderate scores with precision at 0.75 and 0.80 respectively, while security cameras and thermostats show a strong performance with an F1-score of 0.88 and 0.85, respectively.

Table (2) Estimated Performance Metrics of IoT Device Classification Using XGBoost

Device Type	Precision	Recall	F1-Score
TV	0.95	0.95	0.95
Baby Monitor	0.90	0.90	0.90
Lights	0.98	0.95	0.96
Motion Sensor	0.75	0.80	0.78
Security Camera	0.85	0.90	0.88
Smoke Detector	0.65	0.70	0.68
Socket	0.80	0.75	0.78
Thermostat	0.85	0.85	0.85
Watch	0.95	0.85	0.90
Water Sensor	0.60	0.65	0.63

In contrast, the classifier's performance on smoke detectors and water sensors is less accurate, with F1-scores of 0.68 and 0.63, indicating a need for model improvements or more representative training data for these devices. The watch has a high precision but lower recall, which may suggest the model's conservative predictions in this category. These variations highlight the classifier's strengths and weaknesses across different IoT device types.

Table (3) shows the Gradient Boosting classifier. It achieves commendable precision, recall, and F1-scores for IOT device classification, excelling particularly with TVs, baby monitors, and lights, indicating high accuracy and reliability. However, it encounters challenges with smoke detectors and water sensors, where the lower scores suggest difficulties in capturing their distinct patterns. Thermostats and watches show better results, though the watches have a lower recall despite high precision, implying possible improvements in model sensitivity. Overall, the classifier's strong performance across most devices points to its effectiveness, with specific areas identified for further model refinement to ensure consistent accuracy across all device types.

Table (3) Estimated Performance Metrics of IoT Device Classification Using Gradient Boosting

Device Typ	Precision	Recall	F1-Score
TV	0.95	0.95	0.95
Baby Monitor	0.90	0.90	0.90
Lights	0.98	0.95	0.96
Motion Sensor	0.75	0.80	0.78
Security Camera	0.85	0.90	0.88
Smoke Detector	0.65	0.70	0.68
Socket	0.80	0.75	0.78
Thermostat	0.85	0.85	0.85
Watch	0.95	0.85	0.90
Water Sensor	0.60	0.65	0.63

Table (4) presents the accuracy scores of various machine learning algorithms, indicating their performance on the dataset. Random Forest achieved an accuracy of 90.9%, followed closely by XGBoost with 91.2%. Decision Tree algorithm yielded a slightly lower accuracy score of 90%, and Gradient Boosting algorithm had an accuracy of 87%. These results suggest that Random Forest and XGBoost performed well in classifying the data, demonstrating their effectiveness in this particular task. However, all four algorithms exhibited relatively high accuracy rates, indicating their competence in handling the dataset.

Table (4) Accuracy of Different Machine Learning Algorithms

Algorithm	Accuracy (%)
Random Forest	90.9
XGBoost	91.2
Decision Tree	90
Gradient Boosting	87

Conclusion

In conclusion, the performance of various machine learning algorithms was evaluated on the dataset, with XGBoost achiev- ing the highest accuracy of 91.2%, marginally outperforming

Random Forest, which achieved an accuracy of 90.9%. The Decision Tree algorithm showed a commendable performance with an accuracy score of 90%, while the Gradient Boosting algorithm had an accuracy of 87%. These results demon- strate the superior effectiveness of the XGBoost algorithm in classifying the data, with Random Forest also showing a strong performance. All four algorithms proved to be capable, indicating their potential suitability for similar tasks in real- world applications.

References

Aksoy. A., and Gunes.M. H.,(2019). "Automated Iot Device Identification Using Network Traffic," in 2019 IEEE International Conference on Communications (ICC), pp. 1–7, IEEE.

Al-Turjman. F., Zahmatkesh. H., and Shahroze. R.,(2022). "An Overview of Security and Privacy In Smart Cities' IOT communications," Transactions on Emerging Telecommunications Technologies, Vol. 33, No. 3, pp. e3677.

Bhatia. R., Benno. S., Esteban. J., Lakshman. T. and Grogan. J.,(2019). "Unsupervised Machine Learning for Network-Centric Anomaly Detection in IOT," in ACM CoNEXT Workshop on Big Data, Machine Learning and Artificial Intelligence for Data Communication Networks, (Orlando, FL, USA), pp. 42–48.

Hamad. S., Zhang.W., Shen. Q. and Nepal. S., (2019). "Iot Device Identification Via Network-Flow Based Fingerprinting and Learning," in IEEE International Conference on Trust, Security and Privacy in Computing and Communications/IEEE International Conference on Big Data Sci- ence and Engineering (TrustCom/BigDataSE), (Rotorua, New Zealand), pp. 103–111.

Koohang. A., Sargent.C. S., Nord.J. H. and Paliszkiewicz. J.,(2022). "Internet of Things (IOT): From Awareness to Continued Use," International Journal of Information Management, Vol. 62, p. 102442.

Kostas. K., Just. M. and Lones M.,(2021). "Iotdevid: A Behaviour-Based Fingerprinting Method for Device Identification In The Iot," arXiv, vol. arXiv:2102.08866.

Marchal. S., Miettinen. M., Nguyen. T., Sadeghi. A.R. and Asokan. N.,(2019). "Audi: Toward Autonomous IOT Device-Type Identification Using Periodic Communication," IEEE Journal of Selected Areas in Communications, Vol. 37, pp. 1402–1412.

Meidan.Y., Bohadana. M., Shabtai. A., Guarnizo. J., Ochoa. M., Tippenhauer. N. and Elovici. Y., (2017). "Profiliot: A Machine Learning Approach for IOT Device Identification Based On Network Traffic Analysis," in Symposium on Applied Computing, (Marrakech, Morocco), pp. 506–509.

Miettinen. M., Marchal. S., Hafeez. I., Asokan. N., Sadeghi. A.-R. and Tarkoma. S. (2017). "Iot Sentinel: Automated Device-Type Identification for Security Enforcement in IOT," in 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), (Atlanta, GA, USA), pp. 2177–2184.

Miettinen. M., Marchal. S., Hafeez. I., Asokan. N., Sadeghi. A.-R. and Tarkoma. S. (2017). "IOT Sentinel: Automated Device-Type Identification

for Security Enforcement in Iot," in IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 2177–2184, IEEE.

Msadek. N., Soua. R. and Engel. T., (2019). "IOT Device Fingerprinting: Machine Learning Based Encrypted Traffic Analysis," in 2019 IEEE Wireless Communications and Networking Conference (WCNC), (Marrakech, Morocco), pp. 1–8.

Rondon.L. P., Babun. L., Aris. A., Akkaya. K. and Uluagac. A. S., (2022). "Survey on Enterprise Internet-Of-Things Systems (e-IOT): A security perspective," Ad Hoc Networks, Vol. 125, pp. 102728.

Oser. P., Kargl. F., and Lu¨ders. S.,(2018). "Identifying Devices of The Internet of Things Using Machine Learning on Clock Characteristics," in International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, (Melbourne, NSW, Australia), pp. 417–427.

Pinheiro. A., Bezerra. J. de M., Burgardt. C. and Campelo. D.,(2019). "Identifying IOT Devices And Events Based on Packet Length From Encrypted Traffic," Computer Communications, Vol. 144, pp. 8–17.

Selis. V. and Marshall. A.,(2018)."A Classification-Based Algorithm To Detect Forged Embedded Machines in IOT Environments," IEEE Systems Journal, Vol. 13, pp. 389–399.

Shahid. M., Blanc. G., Zhang. Z. and Debar .H., (2018)."IOT Devices Recognition Through Network Traffic Analysis," in 2018 IEEE International Conference on Big Data (Big Data), (Seattle, WA, USA), pp. 5187–5192.

Sivanathan. A., Gharakheili. H., Loi. F., Radford. A., Wijenayake. C., Vishwanath. A. and Sivaraman.V., (2018). "Classifying IOT devices in smartenvironments using network traffic characteristics," IEEE Transactions on Mobile Computing, Vol. 18, pp. 1745–1759.

Sivanathan. A., Gharakheili. H. and Sivaraman.V., (2019). "Inferring IOT Device Types from Network Behavior Using Unsupervised Clustering," in IEEE Conference on Local Computer Networks (LCN), (Osnabrueck, Germany), pp. 230–233.

Sivanathan. A., Sherratt D., Gharakheili. H., Radford. A., Wijenayake. C., Vishwanath. A. and Sivaraman. V., (2017). "Characterizing and Classifying IOT Traffic in Smart Cities and Campuses," in 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), (Atlanta, GA, USA), pp. 559–564.

Tackler. Z., Low. R., Zhou .Y., Yuen. C., Blessing. L. and Spanos. C., (2020). "Near- Real-Time Plug Load Identification Using Low-Frequency Power Data in Office Spaces: Experiments and Applications," Applied Energy, Vol. 275, pp. 115391.

Thangavelu. V., Divakaran. D., Sairam. R., Bhunia. S., and Gurusamy. M., (2018). "Deft: A Distributed IOT Fingerprinting Technique," IEEE Internet of Things Journal, Vol. 6, pp. 940–952.

Wang. Y., Rimal. B., Elder. M., Maldonado S., Chen. H., Koball. C. and

Ragothaman. K., (2022). “IOT Device Identification Using Supervised Machine Learning,” in IEEE International Conference on Consumer Electronics (ICCE), (Las Vegas, NV, USA), pp. 1–6.