

Towards a Better Understanding of IoT Domain Names: A Study of IoT Backend

Ibrahim Ayoub, Martine S. Lenders, Benoît Ampeau, Sandoche Balakrichenan, Kinda Khawam, Thomas C. Schmidt, Matthias Wählisch

▶ **To cite this version:**

Ibrahim Ayoub, Martine S. Lenders, Benoît Ampeau, Sandoche Balakrichenan, Kinda Khawam, et al.. Towards a Better Understanding of IoT Domain Names: A Study of IoT Backend. IEEE Access, 2025, pp.1-1. ⟨10.1109/access.2025.3561521⟩. ⟨hal-05059780⟩

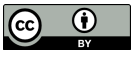
HAL Id: hal-05059780

https://hal.science/hal-05059780v1

Submitted on 26 May 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 March 2025, accepted 8 April 2025, date of publication 16 April 2025, date of current version 25 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3561521

RESEARCH ARTICLE

Toward a Better Understanding of IoT Domain Names: A Study of IoT Backend

IBRAHIM AYOUB[1,2], MARTINE S. LENDERS[3], BENOÎT AMPEAU[1], SANDOCHE BALAKRICHENAN[1], KINDA KHAWAM[4], THOMAS C. SCHMIDT[5], (Member, IEEE), AND MATTHIAS WÄHLISCH[3,6], (Member, IEEE)

[1]Association Française pour le Nommage Internet en Coopération (Afnic), 78280 Guyancourt, France
[2]École Doctorale Sciences et Technologies de l'Information et de la Communication (ED STIC), Université Paris-Saclay, 91190 Gif-sur-Yvette, France
[3]Faculty of Computer Science, Technische Universität Dresden (TU Dresden), 01069 Dresden, Germany
[4]Laboratoire DAVID, Université de Versailles Saint-Quentin-en-Yvelines, 78035 Versailles, France
[5]Department Informatik, Hamburg University of Applied Sciences (HAW Hamburg), 20099 Hamburg, Germany
[6]Barkhausen Institut, 01067 Dresden, Germany

Corresponding author: Ibrahim Ayoub (ibrahim.ayoub@afnic.fr)

This work was supported by the Agence Nationale de la Recherche (ANR, French National Research Agency) under grant number ANR-20-CYAL-0002, and by the Bundesministerium für Bildung und Forschung (BMBF, German Federal Ministry of Education and Research) under grant numbers 16KIS1386K (TU Dresden) and 16KIS1387 (HAW Hamburg), as part of the PIVOT Project (https://pivot-project.info/).

ABSTRACT In this paper, we study Internet of Things (IoT) domain names, the domain names of backend servers on the Internet that are accessed by IoT devices. We investigate how they compare to non-IoT domain names based on their statistical and DNS properties and the feasibility of classifying these two classes of domain names using machine learning (ML). We construct a dataset of IoT domain names by surveying past studies that used testbeds with real IoT devices. For the non-IoT dataset, we use two lists of top-visited websites. We study the statistical and DNS properties of the domain names. We also leverage machine learning and train six models to perform the classification between the two classes of domain names. The word embedding technique we use to get the real-valued vector representation of the domain names is Word2vec. Our statistical analysis highlights significant differences in domain name length, label frequency, and compliance with typical domain name construction guidelines, while our DNS analysis reveals notable variations in resource record availability and configuration between IoT and non-IoT DNS zones. As for classifying IoT and non-IoT domain names using machine learning, Random Forest achieves the highest performance among the models we train, yielding the highest accuracy, precision, recall, and F_1 score. Our work offers novel insights to IoT, potentially informing protocol design and aiding in network security and performance monitoring.

INDEX TERMS Domain names, IoT, machine learning, security.

I. INTRODUCTION

IoT devices are rarely standalone and often contact dedicated backend servers [1]. Interacting with these backend servers is essential for the functioning and maintenance of these devices. To reach one of its backend servers, an IoT device is usually equipped with the domain name of this server which it resolves using the Domain Name System (DNS). Afterwards, a connection can be established, and the device could, for example, relay data to the server or receive commands and firmware updates. To our knowledge, the domain names of the servers that constitute the IoT backend have not been studied in isolation despite the many conclusions that could be drawn from such a study. We aim to address this gap in this paper.

We aim to gain a better understanding of IoT backend servers using their domain names. For simplicity, we refer to these domain names as IoT domain names. We conduct a three-phase study over IoT domain names and compare them to non-IoT domain names, *i.e.*, domain names of servers catering to non-IoT devices and human users.

The associate editor coordinating the review of this manuscript and approving it for publication was Junho Hong.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

The study will allow us to investigate the structure of IoT domain names and if they differ from their non-IoT counterparts, allowing us to determine any patterns or identifiers that may appear in one class of domain names but not in the other. We will also analyze the DNS zones of IoT and non-IoT domain names. A DNS analysis gives an insight into the DNS practices in each class of domain names. It reveals information regarding, for example, the security and dynamism of the average DNS zone from each class of domain names. The final phase of this study involves using machine learning to classify IoT and non-IoT domain names. This will allow detection of the subtle differences between IoT and non-IoT domain names, differences that may not be traceable visually or statistically.

In the context of IoT, machine learning has been widely used, particularly in IoT security. It has been applied to intrusion detection [2], [3], [4], [5], [6], [7], [8] in IoT networks, helping to identify malicious activity. In addition, machine learning models have proven useful for optimizing IoT networks, especially in resource and energy management [9], [10], [11], [12]. Furthermore, different machine learning models have been leveraged to classify and identify IoT devices based on their network traffic patterns [13]. In the broader context, machine learning has also been employed to classify domain names, aiding in the detection of malicious and phishing domain names using classical, deep-learning-based, and transformer-based approaches [14], [15], [16], [17], [18], [19], [20], [21].

In this work, we compile two datasets of domain names. First, using packet captures of real IoT devices from 12 past studies, we construct a dataset of IoT domain names. We call this dataset the IoT dataset. Second, we synthesize two datasets of non-IoT domain names. Previous works [14], [16], [19], [20], [21], [22], [23], [24] use lists of top-visited websites as a negative class in domain name classification problems. We also use two top lists, namely Cisco Umbrella 1 Million [25] and Tranco Top 1M [26], [27] to evaluate the performance of the machine learning models with such lists and the validity of using them in IoT domain name classification.

The remainder of the paper is organized as follows. Section II presents the motivation behind this work. Section III presents the related work, and Section IV provides background information, while Sections V, VI, VII, and VIII outline data collection and the three phases of the study, respectively. Section IX presents the results obtained while Section X discusses the key takeaways, and Section XI presents the limitations of the work. Finally, Section XII concludes the work.

II. MOTIVATION

In this work, we do not study specific IoT technologies or certain properties of these technologies. Instead, we study IoT from a new perspective by studying the domain names of the servers contacted by IoT devices. We aim to investigate the differences between the domain names of these servers and the domain names of servers contacted by non-IoT devices and human users.

We first study the structure of IoT and non-IoT domain names. This includes studying the composition and the statistical properties of these domain names and drawing conclusions about the patterns and particularities of the average domain name of each class. This phase will clarify if the choice of domain names depends on the types of devices the underlying server caters to. In addition, we will be able to extrapolate any possible current conventions in naming IoT domain names and if any special attention is given when assigning domain names to such servers. This could be helpful, for example, to model and generate IoT domain names that align with the average domain name of that type—for instance, aiding protocol design such as name compression for constrained devices [28] and in data augmentation in the context of machine learning.

Second, we study the DNS properties of the IoT and non-IoT domain names. This includes resolving several DNS resource records (RRs) for each domain name of each dataset and calculating for each RR the percentage of domain names that have it, the average resolution time, the average Time To Live (TTL), which is the duration a resolver retains a cached response before initiating a new query, and the average response size. This study will help identify how the DNS zones of each class of domain names are set up. We will be able, for example, to identify the dominant RRs used in each class, whether security-related RRs such as Domain Name System Security Extensions (DNSSEC)-related and Certification Authority Authorization (CAA) RRs appear in IoT DNS zones, how static IoT DNS zones are by comparing their TTL values, and the average size of DNS responses from these zones to indicate if IoT zones, belonging generally to constrained devices, have smaller DNS responses to enhance performance.

The third phase of our study evaluates the feasibility of classifying between the two classes of domain names based solely on the domain name. We investigate if popular machine learning models often used in domain name classification problems can successfully classify IoT and non-IoT domain names based on their inherent differences. This phase of our study provides a deeper insight into the differences between the two classes of domain names. This includes differences that are not detectable by visually inspecting or statistically studying the domain names. We can also conclude through this study the parts of the domain name that are most indicative of class, *i.e.*, which section of the domain name holds the most information and therefore differentiates one class from the other. In addition to drawing conclusions about structural differences between the two classes, successfully classifying IoT and non-IoT domain names using machine learning enables the detection of potential IoT traffic in mixed traffic that contains both classes. This could aid in detecting outliers in IoT networks, for example, non-IoT traffic in networks consisting solely of IoT devices.

The three phases of this study will inform future projects utilizing IoT domain names—either for classification or for evaluating network setups. For example, future machine learning applications dealing with IoT domain names and limited by dataset size or balance could leverage the results of this study to create a model of an IoT domain name, using it to generate synthetic domain names with known statistical properties (such as average length, number of labels, recurring labels), known DNS properties, and insights into the proper setup of DNS zones. This would include understanding the security-related DNS characteristics and the performance profile that the average synthetic IoT domain name would exhibit in machine learning contexts. A recent study [29] demonstrated how the performance of machine learning models could be improved by incorporating synthetic data, emphasizing the importance of such modeling and highlighting potential data augmentation applications in machine learning settings. Protocol design, as seen in [28], is also a promising application of the results of this study, as protocols aiming, for example, to compress known network protocols—such as DNS—to better suit constrained IoT environments will need to test their designs and compare them either against baseline DNS or against other DNS compression protocols. Our results provide direct insight into the average IoT domain name, allowing these protocols to be evaluated on a wide range of domain names with well-known properties that reflect the real-world IoT domain names. Moreover, distinguishing and classifying IoT and non-IoT domain names via machine learning adds a new line of defense against the malicious activity of compromised IoT devices. Extensive work [14], [15], [16], [18], [19], [20], [21] has already been done to detect phishing and domain generation algorithm (DGA)-generated domain names—these are the domain names that compromised devices, a significant portion of which are typically IoT devices, contact to receive commands and victim information. Promptly flagging a domain name as non-IoT enables further scrutiny and may reveal malicious behavior if present.

III. RELATED WORK

A. MACHINE LEARNING-BASED CLASSIFICATION OF DOMAIN NAMES

Leveraging machine learning to classify domain names into two or more classes is a recurring theme in the literature. The role of the machine learning models often revolves around detecting malicious domain names and URLs or DGA-generated ones. Regardless of the preprocessing and word embedding applied, these models have efficiently classified the domain names in question. In [14], the authors combined a Convolutional Neural Network (CNN) and a Bidirectional Long Short-Term Memory (BiLSTM) to improve DGA detection—particularly for short and wordlist-based domain names—by capturing both local features and global sequence dependencies. Their model outperformed CNN and BiLSTM individually in binary and multi-class classification. The authors in [15] proposed CCBLA, integrating attention into CNN-BiLSTM to automatically extract and prioritize URL features, reducing dependency on prior knowledge and improving detection accuracy. A two-stage feature reinforcement was introduced in [16], combining a Slice Pyramid Network (SPN) for feature extraction with a transformer and a Squeeze-and-Excitation Network (SENet), which is a deep learning module designed to enhance feature representation. In [17], the authors proposed PhishBERT. BERT is a transformer-based language model that uses encoder-only transformer architecture. By training BERT on massive URL data (around 3 billion unlabeled URL data) and fine-tuning it with adversarial regularization, the authors were able to use it for phishing URL detection, outperforming traditional deep learning and Natural Language Processing (NLP)-based models in accuracy and efficiency. In [18], BERT was also used—this time to enhance the performance of classical machine learning models such as Random Forest (RF), Support Vector Machine (SVM), and Gradient Boosting in detecting DGAs. Their method integrates BERT embeddings with LSTM and recurrent neural networks (RNNs) to capture contextual and sequential patterns in domain names. In [29], Generative Adversarial Network (GAN)-based approaches were used to synthesize phishing and legitimate websites, which, combined with real data, were used to implement adversarial-attacks-resistant classifiers. Their Experiments showed that incorporating synthesized data increased F_1 scores and improved robustness. In [19], the authors proposed a DGA detection technique based on a novel subword segmentation technique combined with FastText embeddings and a CNN-BiLSTM model to enhance detection accuracy and minimize false positives in Chinese domains, whereas in [20], a lightweight full-convolutional model (Fast3DS) was proposed to enable real-time DGA detection, employing depthwise separable convolutions, global average pooling, and an attention mechanism to enhance efficiency while maintaining high accuracy. The advantages of ensemble methods were demonstrated in [21] where the authors used multi-layer ensemble classifiers, integrating classifiers such as K-Nearest Neighbor(KNN), Decision Tree (DT), RF, Extra Tree, and XGBoost. Their approach improves detection accuracy while maintaining a low response time, whereas in [30], a hybrid voting-based (LSD: Logistic Regression (LR), SVM, and DT) approach was proposed to reduce misclassification further.

B. MACHINE LEARNING FOR IoT

The prevalence of IoT has sparked significant research activity, exploring various aspects of this technology and the risks associated with it. Studies have focused on its coexistence with other technologies, such as DNS [31], and have investigated how modern tools like machine learning can be leveraged to enhance IoT security and, by extension, overall cybersecurity. Machine learning, for example, could help with Intrusion detection [2], [3], [4], [5], [6], [7], [8]. In [3], the authors proposed a data-driven intrusion detection system (IDS) for IoT networks, leveraging Automated

Machine Learning (AutoML) and Synthetic Minority Oversampling Technique (SMOTE) for improved classification. The study in [4] also proposed an anomaly-based IDS for IoT networks using a CNN and a binary multi-objective enhanced Capuchin Search Algorithm (BMECapSA) for feature selection, a hybrid approach that enhanced detection accuracy while reducing redundant features. Another IDS was proposed in [5], integrating CNN, LSTM, Deep Autoencoder, and Deep Neural Networks (DNN) to classify 13 different types of Distributed Denial of Service (DDoS) attacks. The work in [6] presented the IDSAI dataset, a balanced intrusion detection dataset for IoT environments. The survey in [7] is a review of eXplainable Artificial Intelligence (XAI) techniques for anomaly-based intrusion detection in IoT networks which emphasizes the need for interpretability in deep learning-based IDS. In [32], the authors developed a deep anomaly detection (DAD) model for IoT network traffic analysis using deep learning to detect anomalies in IoT networks. Other than IoT security, machine learning could also be employed for IoT resource management and optimization [9], [10], [11], [12], and for IoT device classification [13].

Table 1 provides a comparative overview of our work alongside the most notable recent machine learning approaches in IoT contexts. We highlight the primary focus of each study, the methodology used, the dataset(s) employed, and the evaluation metrics reported.

IV. BACKGROUND

A. IoT DOMAIN NAMES

IoT devices usually contact servers on the Internet to relay information about the physical world or to receive commands and firmware updates [1]. These devices rely on domain names as an indirection mechanism to connect to IP endpoints, simplifying maintenance. This allows, for example, a transparent change of server addresses, as only the mapping in the DNS would need to be changed. The IoT devices are typically pre-configured with the domain names of the servers they might need to contact, and they obtain the addresses of these servers by resolving the domain names via DNS. Beyond address resolution, future IoT devices might also use DNS to identify the service bindings of these servers, *e.g.,* whether they use the TCP-based HTTP/2, the QUIC-based HTTP/3, or other services such as CoAP, using SVCB resource records [33], [34].

The domain names of such servers may exhibit distinct construction characteristics influenced by various factors. An IoT backend server, for example, might have a name that correlates with its high-level function. For example, a collection of IP cameras might have a backend server whose domain name is `cam.example.com`. Moreover, large IoT backend service providers tend to adopt a naming convention for their servers, which follows the pattern below [1]:

`<subdomain>.<region>.<second-level-domain>`,

where `<subdomain>` could be the name of the IoT service or the protocol name, `<region>` refers to the location of the server, and `<second-level-domain>` could be the second-level domain of the service provider or a name related to the IoT service.

Last, being involved in machine-to-machine (M2M) communications, some IoT domain names may contain machine-friendly character sequences that do not prioritize legibility or memorability and are challenging for humans to comprehend. Meanwhile, non-IoT domain names do not follow the same patterns. Since servers with such domain names serve a wide range of devices and human users, the legibility and memorability of their domain names are prioritized.

B. DOMAIN NAME CLASSIFICATION

Domain name classification leverages machine learning techniques to classify two or more categories of domain names. It enables, for example, efficiently detecting phishing or domain names generated by DGAs [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [35], [36]. Phishing domain names are used by malicious servers that pose as legitimate ones and lure users into providing sensitive information and credentials. On the other hand, DGAs run on malware-infected devices and generate domain names to help the infected devices contact the Command & Control (C&C) servers. These domain name classification techniques could also be applied in IoT environments, aiding in classifying IoT and non-IoT domain names and detecting outliers in IoT traffic.

C. BRIEF OVERVIEW OF MACHINE LEARNING

Supervised machine learning is a subset of artificial intelligence in which models learn from labeled data to improve their performance. In this approach, the dataset contains input-output pairs, with the output (target) values serving as the correct answers for the model to learn from. The dataset is split into training and testing datasets. The models are trained using the training dataset, and their performance is evaluated using the testing dataset. In unsupervised machine learning, datasets are unlabeled, and models aim to identify patterns and relationships within the data. Reinforcement learning is a third type of machine learning that trains an agent that interacts with its environment and calculates a reward based on its actions. The agent aims to maximize the reward. Supervised machine learning techniques could be useful in the context of classification between IoT and non-IoT domain names.

Machine learning models have demonstrated their power in classification, be it binary (two classes) or multi-class. They have been used for domain name classification to detect phishing attacks [22], [23], [35], [36]. To achieve this, labeled datasets containing phishing and benign domain names are used to train the models.

The machine learning models usually expect numerical data which necessitates obtaining the real-valued vector

TABLE 1. Summary of related works and our contribution.

Work	*Focus Area*	*Methodology*	*Dataset(s) Used*	*Evaluation Metrics*
Our Work	Focuses on IoT domain names analysis	Statistical analysis + DNS analysis + classification of IoT and non-IoT domain names using machine learning	IoT domain names from 12 past studies that included testbeds of real IoT devices and non-IoT domain names from Cisco Umbrella and Tranco	Accuracy, Precision, Recall, F_1 score
Wang et al. (2023) [14]	Focuses on detecting malicious DGA-generated domain names	Combined CNN & BiLSTM	Benign domain names from the top one million data released by Qi'an Xin company, and DGA domain names from 360 Netlab	Precision, Recall, F_1 score
Yang et al. (2023) [16]	Focuses on detecting malicious DGA-generated domain names	Two-stage feature reinforcement (SPN + Transformer + SENet)	Benign domain names from The Majestic Million list of top-visited websites, and DGA domain names from 360 Netlab	Accuracy, Precison, Recall, F_1 score
Wang et al. (2023) [17]	Focuses on phishing URL detection	Large-scale pre-trained deep transformer model (PhishBERT)	3 billion URLs (Common Crawl, Open PageRank, PhishTank)	Accuracy, F_1 score, TPR@FPR=0.01%
Rao et al. (2024) [18]	Focuses on detecting malicious DGA-generated domain names	Hybrid approach (Machine Learning + Transformer-based Deep Learning)	Kaggle dataset that contains 160,000 domain names	Accuracy, Recall
Shirazi et al. (2023) [29]	Focuses on generating synthetic phishing datasets for robust detection and improved F_1 score	Adversarial Autoencoder + Wasserstein GAN for phishing data synthesis	10 publicly available phishing datasets + synthetic data	F_1 score improvement after adding synthetic data
Lee et al. (2024) [19]	Focuses on detecting malicious DGA-generated domain names	FastText-based embedding + CNN-BiLSTM deep learning model	Benign domain names from the Alexa Top 1M and the Majestic Million lists of top-visited websites, and DGA domain names from DGArchive and UMUDGA	Accuracy, Precision, Recall, F_1 score, false positive rate
Yang et al. (2021) [20]	Focuses on real-time detection of malicious DGA-generated domain names	Full-convolutional deep learning model (Fast3DS)	Benign domain names from the Alexa Top 1M list of top-visited websites, and DGA domain names from UMUDGA and SDGA	Accuracy, Precision, Recall, F_1 score
Ahmadi & Chen (2024) [21]	Uses ensemble learning for real-time phishing website detection with high accuracy and low response time	Multi-Layer Ensemble Model + Deep Learning	Benign domain names from the Alexa Top 1M list of top-visited websites, and DGA domain names from PhishTank	Accuracy, Precision, Recall, F_1 score
Karim et al. (2023) [30]	Uses a hybrid ensemble model to detect phishing URLs	Hybrid Machine Learning + hybrid LSD model (LR + SVM + DT)	Kaggle dataset that contains 11,054 URLs	Accuracy, Precision, Recall, F_1 score, Specificity
Xu et al. (2023) [3]	Uses Auto-ML for optimizing IoT network intrusion detection	Auto-ML + SMOTE	KDD Cup 99 dataset that contains 4.9 million records and various attack types	Accuracy, Kappa, F_1 score, Normalized Mutual Information (NMI)
Asgharzadeh et al. (2023) [4]	Uses CNN + Capuchin Search for intrusion detection in IoT networks	CNN-based IDS + BMECapSA	NSL-KDD & TON-IoT datasets	Accuracy, Precision, Recall, F_1 score
Ahmim et al. (2023) [5]	Detects and classifies multiple DDoS attack types in IoT networks	Hybrid Deep Learning Model (CNN + LSTM + Deep Autoencoder + DNN) with Transfer Learning	CIC-DDoS2019 dataset	Accuracy, Average Detection Rate, Average Accuracy, False Alarm Rate

representation of the domain names. Depending on the method used, the real-valued vector representation can capture the semantic information and context of words in the text.

There are several options to achieve this. One way is through NLP. NLP methods aim to obtain the real-valued vector representation of textual data. This includes, for example, character level embedding [20], [23], [37], which obtains a fixed-size real-valued vector representation of each character. Another NLP method is the Term Frequency-Inverse Document Frequency (TF-IDF), which assigns an importance value to each element of the text based on its frequency of appearance [24]. A different way of processing textual data is feature extraction. Feature extraction methods study the text and try to extract properties (*e.g.,* domain name length, number of hyphens, number of dots) and use those properties as a real-valued vector representation of the text [24], [38].

D. MACHINE LEARNING MODELS OVERVIEW

We use supervised machine learning techniques to perform the binary classification between IoT and non-IoT domain names. The following is a brief overview of the six models we use, which can be used for binary and multi-class classification, except Logistic Regression, which is used for binary classification but can be extended to perform multi-class classification.

1) NAÏVE BAYES (NB) [39]

NB is a parametric classifier based on Bayes's theorem. It is referred to as *naïve* since it assumes the mutual independence of features. This means that the value of one feature does not affect the value of any other feature in the input vector. While this assumption might not hold in real-world scenarios, it simplifies computation.

2) LOGISTIC REGRESSION (LR) [40]

LR is a parametric classifier. It is referred to as *logistic* since it uses the *logistic function* (sigmoid function) to map a linear combination of features to a probability score.

3) K-NEAREST NEIGHBORS (KNN) [41]

KNN is a non-parametric and instance-based classifier. It is a simple model that does not involve the traditional training

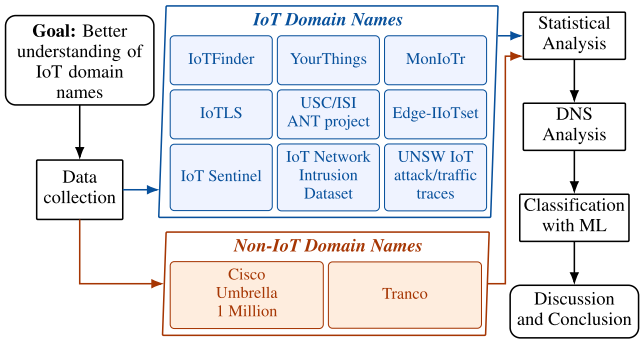


FIGURE 1. Overview of the three-phase study on IoT domain names.

and testing phases. Instead, it assumes that similar data points are placed closer to each other, so the class of a particular data point is similar to the class of its *K nearest neighbors*.

4) SUPPORT VECTOR MACHINE (SVM) [42]

SVM is a parametric classifier whose primary objective is to find a hyperplane that best separates the data points of different classes. The closest points to the hyperplane from the different classes are called *support vectors*.

5) DECISION TREE (DT) [43]

DT is a non-parametric classifier. A DT is a *tree-like structure* where every node represents a feature, every branch is a decision, and every leaf node is a class. DTs are trained through a process known as recursive splitting, and the goal is to have most features represented as nodes with their branches and leaf nodes. A downside of DTs is their tendency to experience overfitting as the size of the dataset grows.

6) RANDOM FOREST (RF) [44]

RF is a non-parametric classifier. It is referred to as an *ensemble method* because it leverages the power of a group (an ensemble) of individual Decision Trees to improve predictive accuracy and reduce overfitting.

E. Word2vec FOR WORD EMBEDDING

In this paper, we use a Word2vec model to obtain the real-valued vector representations—also known as word embeddings in NLP—of the domain names. Word2vec [45], [46] is one of several word embedding techniques, and it uses a shallow neural network to convert each word to a vector of real numbers. Word2vec captures semantic relationships between words by learning from large amounts of text data, and the resulting real-valued vectors depend on the context in which each word appears within the text. Two possible architectures for Word2vec exist: Continuous Bag-of-Words (CBOW) and Skip-gram. CBOW estimates the vector representation of a target word based on the context (*i.e.*, surrounding words) of this word. The number of surrounding words considered within the context is specified by a *window size*. Two words frequently appearing together in similar contexts within the text will have vector representations that are geometrically close to each other in the vector space. Consequently, words that do not appear together in the text are assigned vector representations that are distant geometrically. For Skip-gram, the target word is used to predict the surrounding context words within a specified window size. Between CBOW and Skip-gram, we chose CBOW as it is less expensive computationally and faster to train [46].

In the following sections, we demonstrate the different phases of our study: statistical analysis, DNS analysis, and the classification between IoT and non-IoT domain names using machine learning. Figure 1 summarizes our steps. We first discuss how the data were collected to construct the lists of IoT and non-IoT domain names.

V. DATA COLLECTION

In our analysis, we use two categories of domain name datasets. The first is the IoT dataset, which includes a list of domain names of servers contacted by IoT devices. The second category must contain domain names of servers that serve generic devices and human users, namely non-IoT domain names. For this purpose, we use two lists of top-visited websites, the Cisco Umbrella 1 Million [25] and Tranco [26], [27].

A. IoT DATASET

The IoT dataset should contain domain names of servers on the Internet contacted by IoT devices, *e.g.,* IoT backend servers that provide services to IoT devices, such as a server that saves the footage from IP cameras or servers from which IoT devices receive commands and software updates. This list may also include domain names of servers that are not IoT-specific but are nevertheless contacted by IoT devices. To construct the IoT dataset, we used 12 datasets from previous studies: IoTFinder [47], YourThings [48], MonIoTr [49], IoTLS [50], three datasets from the USC/ISI ANT project [51], [52], [53], Edge-IIoTset [54], IoT Sentinel [55], IoT Network Intrusion Dataset [56], UNSW IoT traffic traces [57], and UNSW IoT attack traces [58]. These datasets contain packet captures collected in testbeds that included real IoT devices. Each is available as a set of PCAP files, including DNS messages sent from and received by the devices. Figure 2 presents the categories of IoT devices included in our study along with examples from each category, and Table 2 lists the individual devices used in each testbed. Below is a brief overview of the studies:

- **IoTFinder** [47]: IoTFinder is a multi-label classifier for detecting IoT devices by studying passively collected DNS traffic. The data were collected between August 1, 2019, and September 30, 2019.
- **YourThings** [48]: A study of home-based IoT devices to assess their security properties. The data were collected between April 10 and April 19, 2018.
- **MonIoTr** [49]: A study of information exposed in the traffic of consumer IoT devices. The data were

collected between March 28 and May 8, 2019, and on September 1, 2019.

- **IoTLS** [50]: A study about the use of TLS in consumer IoT devices. The data were collected between January 2018 and March 2020.
- **USC/ISI ANT project** [51], [52], [53]: The ANT Lab is an Internet research group at the University of Southern California (USC) that has published several datasets related to various network topics, *e.g.,* traffic, outage, and DNS. We used three datasets from the USC/ISI ANT project. Two datasets contain the bootup traces of several IoT devices. The third dataset contains traffic observed in an IoT network of several devices over ten days.
- **Edge-IIoTset** [54]: An IoT traffic dataset that includes benign and attack traffic. The benign traffic we used in this paper was collected between November 21, 2021, and January 10, 2022.
- **IoT SENTINEL** [55]: IoT SENTINEL is a security system that identifies devices present in the network and monitors traffic from vulnerable ones. The traffic was collected during the setup of each device.
- **IoT Network Intrusion Dataset** [56]: An IoT traffic dataset that includes benign and attack traffic. We used the benign traffic in this paper.
- **UNSW IoT traffic traces** [57]: A study about the classification of IoT devices in Smart Home environments. The traffic was collected between October 2016 and April 2017.
- **UNSW IoT attack traces** [58]: A study about detecting volumetric attacks against IoT devices and the dataset includes benign and attack traffic. The traffic was collected for 16 days.

For each study, we obtained the corresponding set of packet captures, either publicly available for download or provided upon request. These packet captures recorded network traffic exchanged either between IoT devices or between the devices and external servers on the Internet. The captures varied in scope, with some containing the full network traffic and others offering a filtered version; however, all of them included the DNS traffic exchanged by the IoT devices, both within the local network and with external servers. Each capture, corresponding to an experiment, covered the whole duration of that experiment, which ranged from several days to a few months.

We first aimed to reduce the size of the packet capture set to a more manageable level while preserving the data useful to our study. Hence, we filtered the PCAP files, extracting only the unique packets pertaining to DNS responses received by each device. These packets should contain in their `'Answer'` section the domain names resolved by the DNS query each packet is associated with, *i.e.*, the domain names resolved by the IoT devices in each testbed. Some datasets also contained captures from generic non-IoT devices such as desktop PCs, smartphones, or gaming consoles. These devices are shaded in Table 2, and we removed their packet captures. Finally, we extracted the queried domain names from the resulting DNS responses. The number of unique IoT domain names obtained from each dataset is presented in Figure 3. The resulting IoT dataset contained 4145 unique domain names.

B. NON-IoT DATASETS

The non-IoT dataset should contain domain names of servers used by generic, non-IoT devices or those used by humans directly. For that purpose, we synthesize two non-IoT datasets using publicly available lists of top-visited websites. We use two of these lists:

- **Cisco Umbrella 1 Million** [25], which we refer to as the Cisco dataset, is a daily-published list of one million websites. Any domain name could be included in the list. The ranking of each domain name is based on the number of unique client IPs that visited it [25]. The list for our evaluation was gathered on July 15, 2024.
- **Tranco** [26], which we refer to as the Tranco dataset, is a research-oriented list of one million domain names. The ranking of each domain name is based on its average rank over the past 30 days from four other popular domain name lists [27]. The Tranco list for our evaluation was gathered on July 15, 2024, and thus covers the period from June 15 to July 14, 2024.

Such top-lists are usually used in research, especially in domain name classification problems [20], [22], [23], [24]. Some IoT domain names might appear in the top-visited websites lists, and we can account for that by removing these domain names from the Cisco and Tranco datasets.

VI. STATISTICAL STUDY

We perform a statistical analysis of the domain name lengths and number of labels in each dataset, for which the results can be seen in the violin plots in Figure 4. Violin plots are similar to box plots, showing key statistical properties. However, they also estimate the probability density function (PDF) as a trace that forms the ''body'' of the ''violins'' around the properties.

The violin plots allow us to easily spot a similarity between domain names in the IoT and Cisco datasets regarding domain name length and number of labels per domain. This is due to the way each dataset is constructed. The two datasets contain domain names as observed in the DNS requests and are, therefore, more representative. The Tranco dataset, on the other hand, is different. The average Tranco domain name has fewer characters and labels than the average domain name from the other datasets. The Tranco dataset mainly contains second-level domains in the form of *domain.tld*, while the IoT and Cisco datasets do not have that limitation.

Moreover, it is observed that IoT domain names do not adhere to the restrictions imposed by RFC 1035 [59], which limit the maximum domain name size to 253 bytes and the maximum size of the label to 63 bytes. In fact, among the 4145 domain names present in the IoT dataset, 73 domain names exceed 253 bytes with a maximum length of 652 characters. Meanwhile, the domain names from the Cisco

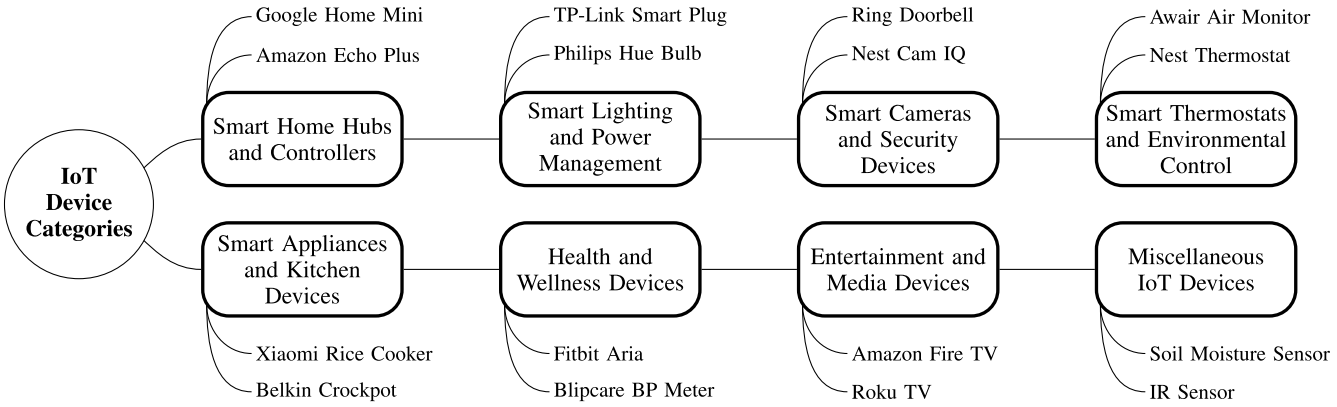


FIGURE 2. IoT devices by category.

TABLE 2. The IoT devices in the testbeds of the datasets used in this study.

IoTFinder & YourThings	Nest Camera	D-Link Mov Sensor	Xiaomi Cam	Smartlife Bulb	Soil Moisture Sensor	Amazon Echo
Echo Dot Gen3	Nest Guard	Echo Dot	Xiaomi Cleaner	Smartlife Remote	Sound Sensor	August Cam
Amazon Echo Gen1	Nest Protect	Echo Plus	Xiaomi Hub	Smartthings Hub	Stepper Motor	Awair air monitor
Amazon Fire TV	Nest Thermostat	Echo Spot	Xiaomi Rice Cooker	Switchbot Hub	Temperature Sensor	Belkin Camera
Android Tablet	Netgear Arlo Camera	Fire TV	Xiaomi Strip	TP-Link Bulb	Ultrasonic Sensor	Wemo Motion Sensor
Apple HomePod	Nintendo Switch	Flux Bulb	Yi Cam	TP-Link Plug	Water Level Sensor	Belkin Switch
Apple TV (4thGen)	Nvidia Shield	GE Microwave	ZModo Doorbell	Wemo Plug	**IoT SENTINEL**	Blipcare BP Meter
August Door Cam	Philips Hue	Google Home	**IoTLS**	Wink Hub 2	D-Link Home Hub	Canary Camera
AVTech IP Cam	Piper NV	Google Home Mini	Amazon Cloudcam	Yi Camera	D-link Day Camera	Dropcam
Belkin Netcam	Play Station 4	Honeywell T-stat	Amazon Echo Dot	Zmodo Doorbell	D-Link Door Sensor	Google Chromecast
Belkin Crockpot	Rachio 3	Insteon Hub	Amazon Echo Dot 3	**ISI**	D-link HD IP Camera	Hello Barbie
Belkin WeMo Link	Ring Doorbell	Invoke with Cortana	Amazon Echo Plus	Amazon Dash Button	D-link Motion Sensor	HP Printer
Wemo Motion Sensor	Roku 4	Lefun Cam	Amazon Echo Spot	Amazon Echo Dot	D-link Siren	iHome PowerPlug
Belkin WeMo Switch	Roku TV	LG TV	Amcrest Camera	Amazon Fire TV	D-link Smart Plug	LiFX Bulb
Bose SoundTouch 10	Roomba	Lightify Hub	Apple HomePod	Amcrest IP Cam	D-Link Water Sensor	NEST Smoke Sensor
Canary	Samsung Hub	Luohe Cam	Apple TV	Belkin SmartPlug	Edimax Cam	Netatmo Camera
Caseta Wireless Hub	Samsung Smart TV	Magichome Strip	Behmor Brewer	D-Link IP Cam	Edimax Smart Plug	Netatmo station
Chamberlain Opener	Securifi Almond	Microseven Cam	Blink Camera	Dyson Purifier	Edimax Smart Plug 2	Phillip Hue Lightbulb
Chinese Webcam	Sonos	Nest T-stat	Blink Hub	Foscam IP Cam	Ednet Cam	Pixstart photo frame
D-Link DCS-5009L	Sonos Beam	Netatmo Weather	D-Link Camera	Foscam IP Cam 2	Ednet Gateway	Ring Door Bell
Facebook Portal	TP-Link WiFi Bulb	Philips Bulb	Fire TV	Google Speaker	Fitbit Aria	Samsung Smart Cam
Google Home Hub	TP-Link WiFi Plug	Philips Hue Hub	GE Microwave	HP Envy 4500 Printer	Homematic switch	Smart Things
Google Home Mini	Ubuntu Desktop	Ring Doorbell	Google Home Mini	Philips Hue	Lightify Gateway	TP-Link Camera
Google OnHub	Wink Hub	Roku TV	Harman Invoke	Renpho Humidifier	MAX! Gateway	TP-Link Plug
Google Home	Wink Hub 2	Samsung Dryer	Insteon Hub	Samsung IP Cam	Philips Hue Bridge	Triby Speaker
Harman Invoke	Withings Home	Samsung Fridge	LG Dishwasher	Tenvis IP Cam	Philips Hue Switch	Withings Monitor
Insteon Hub	Xbox One X	Samsung TV	LG TV	TPLink Smart Bulb	Smarter iKettle	Withings Scale
iPad	**MonIoTr**	Samsung Washer	Meross Dooropener	TPLink Smart Plug	Smarter Coffee	Withings sleep sensor
iPhone	Allure with Alexa	Sengled Hub	Nest Thermostat	Wansview IP Cam	TP-Link Smart Plug	Amazon Echo
Koogeek Lightbulb	Amazon Cloud Cam	Smarter Brewer	Philips Hub	Wyze IP Cam	TP-Link Smart Plug 2	Chromecast Ultra
LG WebOS TV	Amcrest Cam	Smarter iKettle	Ring Doorbell	**Edge-IIoTset**	WeMo Insight Switch	iHome Smart plug
LIFX Virtual Bulb	Anova Sousvide	Smart Things Hub	Roku TV	DC Motor	WeMo Bridge	LiFX bulb
Harmony Hub	Apple TV	TP-Link Bulb	Samsung Dryer	Flame Sensor	Wemo Switch	Netatmo camera
Logitech Logi Circle	Behmor Brewer	TP-Link Plug	Samsung Fridge	Heart Rate Sensor	Withings Scale	Phillips Hue bulb
VeraLite controller	Blink Cam	Wansview Cam	Samsung TV	Humidity Sensor	**Kang**	Samsung smartcam
My Cloud EX2 Ultra	Blink Hub	WeMo Plug	Samsung Washer	IR Receiver Sensor	EZVIZ WiFi Camera	TP-Link smart plug
Nest Bell	Bosiwo Cam	WiMaker Camera	Sengled Hub	pH Sensor	SKT NUGU speaker	WeMo motion
Nest Cam IQ	D-Link Cam	Wink 2 Hub	Smarter iKettle	Servo Motor	**UNSW\Attack**	WeMo Switch

and Tranco datasets are at most 253 bytes, with maximum lengths of 253 and 75 characters for the Cisco and Tranco datasets, respectively. This suggests that size restrictions are less stringent when choosing an IoT domain name. However, considering that a significant portion of IoT devices are constrained and have limited resources, reducing the length of the domain names these devices resolve could reduce the overall size of the DNS messages and potentially lower power consumption.

In addition to studying the length and number of labels, we plot the relative frequencies of the top labels in each dataset in Figure 5. The goal is to assess how common—or uncommon—the labels between IoT and non-IoT domain names are. Hence, we plot the top 20 labels for each dataset and aggregate the remaining labels under *others*.

The label "com" comes first in the three datasets, accounting for approximately 20% of the labels in the Cisco and Tranco datasets and 8% of the labels in the IoT dataset. Beyond "com", the IoT and non-IoT domain names diverge in their subsequent top labels. The most frequent labels in the two non-IoT datasets, Cisco and Tranco, are mostly Top-Level Domains (TLDs) such as "net", "org", and "co". See Figures 5b and 5c. Meanwhile, the most frequent labels in the IoT dataset do not follow the same trend. We observe in Figure 5a that, contrary to the non-IoT datasets, the top 20 labels in the IoT dataset are not mostly TLDs. Instead, we observe protocol-related labels such as "tcp" and "udp" and IoT-technology-specific and brand-specific labels such as "_homekit" and "nest". This indicates that the domain names in IoT environments are often tailored specifically for

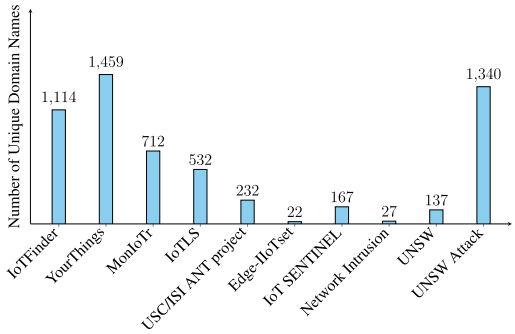


FIGURE 3. The number of unique IoT domain names extracted from each dataset.

these environments. Furthermore, the presence of ''local'' indicates a prevalence of local domains, highlighting the focus on local communications within IoT environments. Finally, the appearance of ''in-addr'' and ''arpa'' in the top labels indicates significant reverse DNS activity.

The statistical study demonstrates the characteristics of both classes of domain names and allows us to draw a few conclusions about the differences between IoT and non-IoT domain names. First, we noticed a deviation from RFC 1035 [59] guidelines concerning domain name size in the IoT dataset, a phenomenon not present in the non-IoT datasets, namely the Cisco and Tranco datasets. The DNS activity in IoT environments mainly involves machine-to-machine communications, with many of these domain names configured using auto-configuration schemes. When combined with the significant local DNS activity in IoT environments—evidenced by the strong presence of labels like ''local'' in the IoT dataset—it becomes apparent why longer domain names might appear in such networks. Second, when studying the most frequent labels in each dataset, several protocol-related or technology-related labels such as ''tcp'' and ''nest'' appeared among the top labels in the IoT dataset, signaling an inclination to give explicit, self-explanatory domain names to IoT-related backend servers. On the other hand, the top 20 most frequent labels in the non-IoT datasets were predominantly TLDs. This demonstrates the stark diversity of these domain names in labels other than TLDs, which, unlike IoT domain names, do not indicate specific protocols, technologies, or providers.

VII. DNS ANALYSIS

In this section, we perform a DNS analysis of the different datasets. The goal is to compare the IoT, Cisco, and Tranco datasets by resolving several RRs and recording their availability, the query duration, the TTL, and the response size for each RR. This study gives insights into any possible differences between the DNS zones of the average domain name from each dataset. If found, we highlight these differences. The RRs we resolve are:

- ***A***: IPv4 address record. It maps a domain name to an IPv4 address.
- ***AAAA***: IPv6 address record. It maps a domain name to an IPv6 address.
- ***MX***: Mail exchange record. It routes emails to specific mail servers.
- ***CNAME***: Canonical name record. It is used when a domain name is an alias for another domain name.
- ***DNSKEY***: It holds a public key that resolvers use to verify DNSSEC signatures.
- ***DS***: Delegation Signer record. It is used to verify DNSKEY records of child DNS zones.
- ***TXT***: Stores text notes.
- ***SRV***: Service record. It specifies the server/port number of a specific service.
- ***CAA***: Certificate Authority Authorization record. It allows domain name owners to specify which certificate authorities are allowed to issue TLS certificates for their domain name.

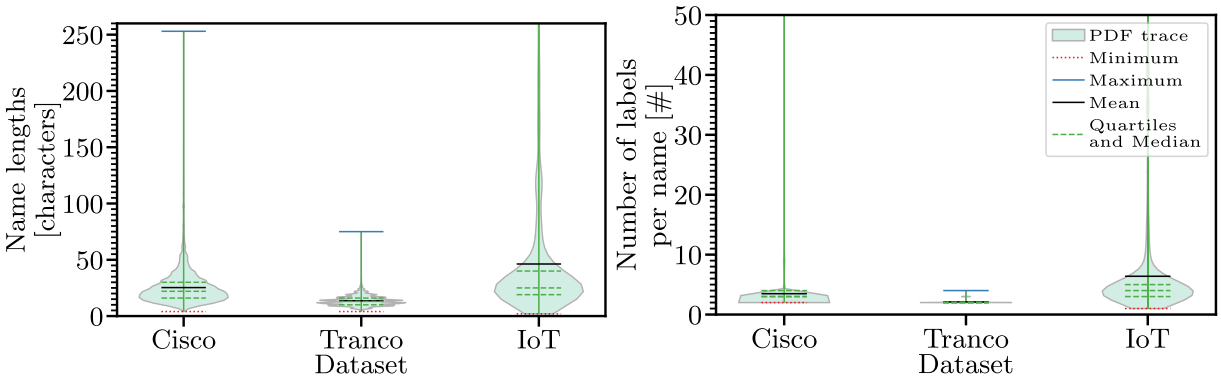
We use Google's public DNS resolver (`'1.1.1.1'`), and the study is done on the 4145 domain names of the IoT dataset and the top 4145 domain names of the Cisco and Tranco datasets. The results are presented in Table 3. The *Record existence [%]* is the percentage of domain names in each dataset with that RR. The rest of the values, namely *Query duration [s]*, *TTL [s]*, and *Response size [bytes]*, are average values over each dataset.

A. RECORD EXISTENCE

The IoT dataset has the lowest existence rate for all the RR types. In particular, only 46.66% of the IoT domain names have an A RR, and only 14.60% of them have a AAAA RR. This is partly due to the nature of domain names in the IoT dataset, which includes local (Multicast DNS) addresses that do not resolve to A and AAAA RRs in the global DNS, *i.e.*, outside the local network but may have such records in a local network context. As for MX records, only 1.01% of IoT domain names have this RR, while it is 12.30% and 71.70% for the Cisco and Tranco datasets, respectively. The low presence of MX records in the DNS zones of IoT domain names indicates that these domain names are less likely to be associated with email services. Security-related RRs are also sparsely present in IoT-related DNS zones, with only 0.12% of IoT domain names having DNSSKEY or DS RRs, and only 0.53% of them having a CAA RR. Meanwhile, the non-IoT datasets show a higher—but nevertheless low—percentage for these RRs. The Cisco dataset has a 1.83%, 1.64%, and 5.33% presence of DNSKEY, DS, and CAA RRs, respectively, among its domain names, while the percentages are 10.11%, 8.69%, 20.46% for the Tranco dataset. This signifies security concerns in IoT environments, which already face several security vulnerabilities due to the constrained nature of a lot of IoT devices [60].

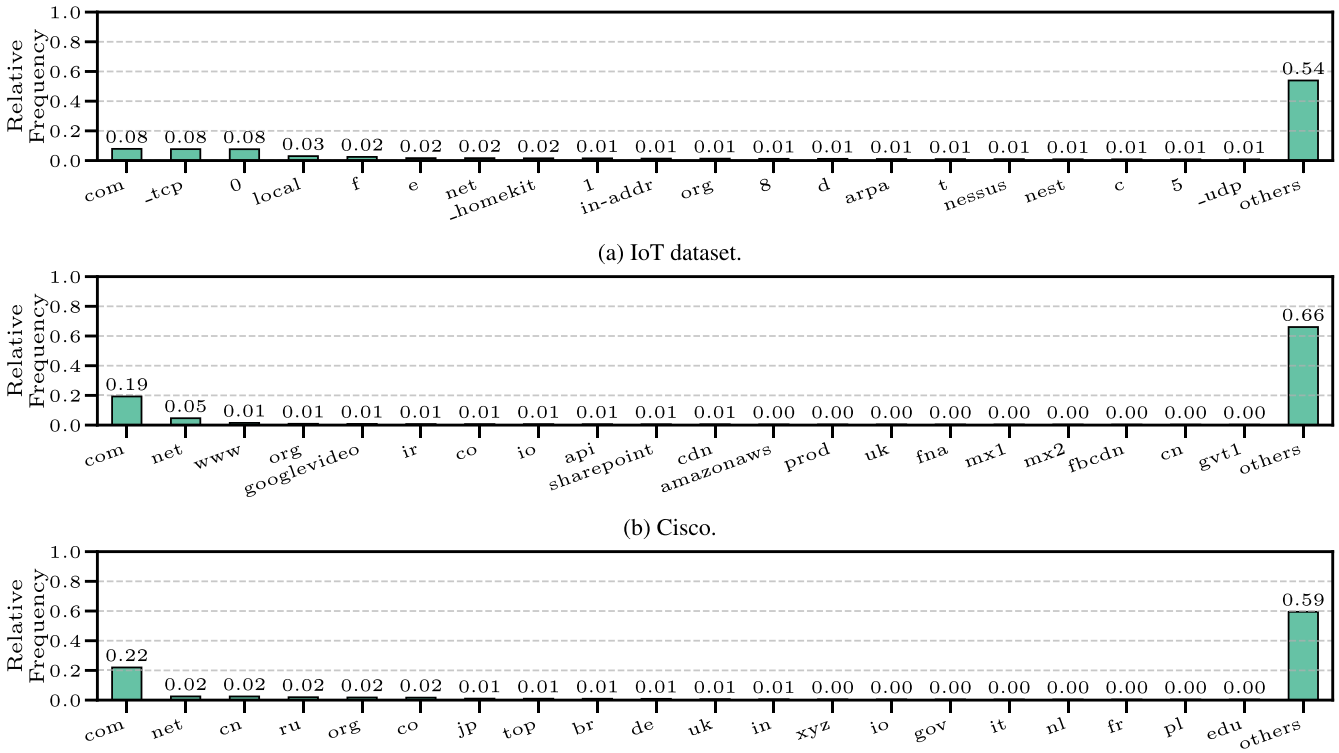
B. TIME TO LIVE (TTL)

Other than RRs that are inherently not adopted by IoT domain names, as demonstrated in the previous paragraph,



(a) Name lengths. (b) Number of labels.

FIGURE 4. Violin plots illustrating domain name properties across the IoT, Cisco, and Tranco datasets.



(a) IoT dataset.

(b) Cisco.

(c) Tranco.

FIGURE 5. Relative frequencies of the top labels in the IoT, Cisco, and Tranco datasets.

and whose TTL values are not accounted for, the TTL values of the RRs of IoT domain names exceed those of domain names from the Cisco and Tranco datasets or are at least comparable. This indicates a more static DNS environment for IoT domain names with less frequent updates than non-IoT domain names. In addition, higher TTL values introduce some security risks, such as slower propagation of security updates (*e.g.,* key rollovers in DNSSEC) and prolonged exposure to DNS cache poisoning if a malicious RR is cached for an extended period.

C. QUERY DURATION AND RESPONSE SIZE

No notable differences are observed when comparing the query durations and the response sizes between the RRs across the three datasets. This is interesting, considering that a large portion of IoT technologies include constrained devices with a limited power budget. It seems, however, that DNS response size is not yet optimized to suit the limited resources some of these devices have. IoT-friendly DNS protocols such as DNS-over-CoAP (DoC) [61] could be useful to address this issue.

TABLE 3. The results (averages) of the DNS analysis of the IoT and non-IoT datasets.

	A				AAAA				MX				CNAME				DNSKEY				DS				TXT				SRV				CAA			
	Record existence [%]	Query duration [s]	TTL [s]	Response size [bytes]	Record existence [%]	Query duration [s]	TTL [s]	Response size [bytes]	Record existence [%]	Query duration [s]	TTL [s]	Response size [bytes]	Record existence [%]	Query duration [s]	TTL [s]	Response size [bytes]	Record existence [%]	Query duration [s]	TTL [s]	Response size [bytes]	Record existence [%]	Query duration [s]	TTL [s]	Response size [bytes]	Record existence [%]	Query duration [s]	TTL [s]	Response size [bytes]	Record existence [%]	Query duration [s]	TTL [s]	Response size [bytes]	Record existence [%]	Query duration [s]	TTL [s]	Response size [bytes]
IoT	**46.66**	77.97	**1165846.37**	134.05	**14.60**	60.62	**6583.37**	175.23	**1.01**	187.70	15532.50	125.40	28.97	76.12	8540.39	96.53	**0.12**	38.79	2943.40	276.60	**0.12**	166.17	69180.0	88.80	10.74	148.96	4242.48	184.07	0.0	0.0	0.0	0.0	**0.53**	40.85	18204.50	190.82
Cisco	**78.41**	51.27	**999.08**	138.04	**29.67**	52.03	**564.32**	164.04	**12.30**	60.41	12911.55	126.97	40.97	69.20	6986.38	96.92	**1.83**	65.99	1622.97	275.84	**1.64**	125.06	54651.0	94.26	19.47	74.88	7274.32	598.11	0.07	159.15	58860.0	115.67	**5.33**	56.76	15958.43	182.28
Tranco	**79.88**	76.21	**4164.59**	75.18	**24.34**	53.57	**3190.43**	108.61	**71.70**	99.69	19324.49	119.80	0.55	180.26	635.22	76.26	**10.11**	51.31	40029.60	370.47	**8.69**	121.86	55525.03	99.79	83.43	109.70	9388.62	738.17	0.29	111.77	8810.0	118.42	**20.46**	75.58	18361.85	225.75

TABLE 4. Number of unique domain names after the syntax check.

Dataset	*IoT*	*Cisco*	*Tranco*
Accepted [#]	3 953	993 065	1 000 000
Discarded [#]	192	6 935	0
Total [#]	4 145	1 000 000	1 000 000

TABLE 5. Number of unique domain names after removing the IoT domain names from the non-IoT datasets.

Dataset	*Cisco*	*Tranco*
Common with IoT Dataset [#]	1 565	32
Remaining [#]	991 500	999 968
Total [#]	993 065	1 000 000

TABLE 6. Number of unique domain names in the final datasets.

Dataset	*IoT*	*Cisco*	*Tranco*
Domain Names [#]	3 953	991 500	999 968

After the statistical and DNS analysis, we move to classifying IoT and non-IoT domain names using machine learning in the next section.

VIII. CLASSIFYING IoT AND NON-IoT DOMAIN NAMES: PREPROCESSING AND WORD EMBEDDING

A. DATA PREPROCESSING

We preprocess the data to ensure the validity and consistency of the domain names in each dataset. We perform the following tests:

1) SYNTAX CHECK

The first step involves a syntax check to ensure consistency across our datasets, verifying that all domain names follow the same syntax rules. For this purpose, we use the syntax checking used by Zonemaster [62]. See Figure 6. The process starts with a normalization procedure that replaces all the dots with the regular full stop of Unicode `'\u002E'` (or `'.'` as a character). The next step removes leading and trailing spaces. Next, a sequence of tests is conducted.

- Check if the domain name starts with a dot,
- Check if the domain name has consecutive dots,
- Remove trailing dots if found,
- Check if any label in the domain name is longer than 63 characters,
- Check if the total length of the domain name is more than 253 characters,
- Check if the domain name has only one label,
- Check if any label starts or ends with a hyphen (`'-'`), and, finally,
- Check if the domain name has double hyphen (`'--'`) at positions 3 and 4 without it starting with `'xn'`.[1]

If one of the checks fails, the domain name is discarded. The results of the syntax check are presented in Table 4.

2) REMOVE COMMONS

Some domain names from the IoT dataset might appear in the Cisco or Tranco datasets. Therefore, we remove the common domain names between the IoT and other datasets from the other datasets. The resulting dataset sizes can be seen in Table 5.

3) FINAL LISTS

After data preprocessing, we obtain the final datasets, which will be used in the following steps. The final sizes of the datasets can be seen in Table 6.

[1] *i.e.*, not an Internationalized Domain Name (IDN).

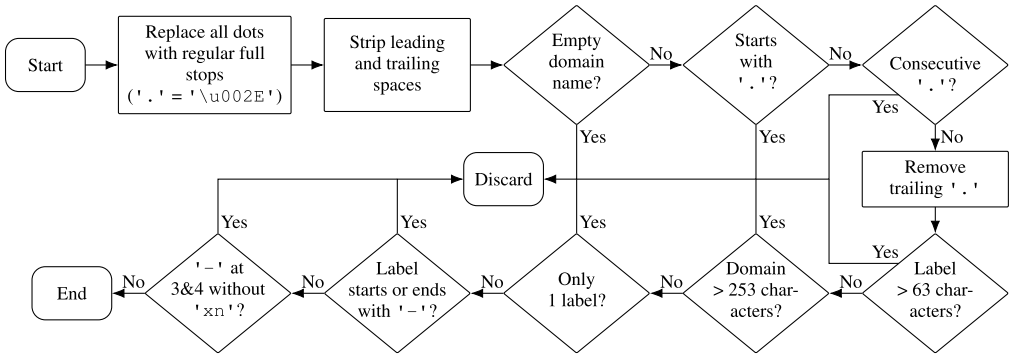


FIGURE 6. Domain name syntax check.

B. Word2vec: REAL-VALUED VECTOR REPRESENTATION OF DOMAIN NAMES

We use Word2vec to represent domain names as real-valued vectors, which can then be fed into machine learning algorithms. Word2vec expects prose text as input, *i.e.*, in the form of full documents with connected sentences and ideas where it can be used to capture the semantic relations. The challenge we face when using Word2vec with domain names is that these domain names do not form prose text. Instead, they are individual labels separated by periods. As such, we treat each domain name as a sentence and each label as a word. For example, `iot.backend.org` contains three labels and is transformed to "iot", "backend", and "org". Another challenge is the limited size of domain names, which results in limited context and explains our choice of a *window size* of 3. After the Word2vec algorithm completes, we obtain a real-valued vector representation of each label. The dimensions of each vector are set in advance. Before applying Word2vec, and to have a consistent dataset in terms of size for training the machine learning models, we pad the domain names by adding `'*'` as a dummy label on the left of each domain name. We pad all the domain names to have 120 labels to account for the maximum number of labels per domain in the three datasets, which is 117.

The parameters we used are as follows:

- **Padding:** To each domain name, we added `'*'` on the left. Each `'*'` was treated as a dummy label (*i.e.*, a word), and they were added until all the domain names were of length 120 labels (words).
- **Word2vec:** We used CBOW (Continuous Bag-of-Words Model) with a *window size* of 3.
- **Vectors:** Each label (word) was represented by a vector $\in \mathbb{R}^{32}$.

This vector representation can then be used to map each domain name to a 32×120 real-valued vector ($\in \mathbb{R}^{32\times120}$). The Word2vec process is depicted in Figure 7.

IX. RESULTS: DOMAIN NAME CLASSIFICATION

We train six machine learning models to classify IoT and non-IoT domain names. We use the following models and refer to them using the acronyms in parentheses: Naïve Bayes (NB), Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF).

After preprocessing the data, the IoT dataset contains 3953 domain names. We then select 3953 domain names individually from the Cisco and Tranco datasets and create an additional list of 3953 domain names by uniformly sampling a Mix of the Cisco and Tranco datasets.

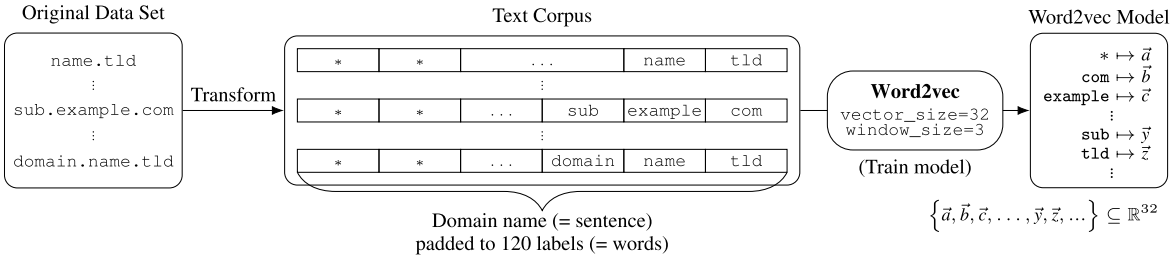
We select the 3953 domain names from the Cisco and Tranco datasets in two ways:

- We select the top 3953 domain names or
- randomly choose them, uniformly distributed, from the whole list.

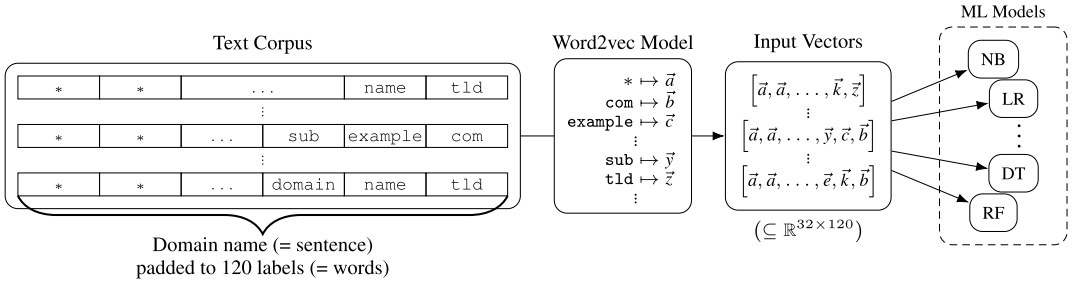
After selecting 3953 domain names from each dataset, the domain names are labeled accordingly, and a combined dataset is constructed. The combined dataset is then processed via Word2vec to obtain the real-valued vector representation of each domain name. These real-valued vectors are used to train the machine learning models. The models are trained as binary classifiers between two classes, namely IoT and non-IoT domain names, where the non-IoT domain names come from the Cisco dataset, Tranco dataset, or a Mix of them. To evaluate the performance of each of the models, we calculate the resulting accuracy, precision, recall, and the F_1 score in subsection IX-A. Moreover, in subsection IX-B, we perform cross-validation to assess the robustness of the models and their ability to generalize to unseen data. Lastly, we perform in subsection IX-C an ablation test to analyze the impact of the different labels of the domain names on the performance of the models.

A. PERFORMANCE EVALUATION

In this section, we present our results after training several machine learning models to classify IoT and non-IoT domain names. In each scenario, each dataset was processed with Word2vec to obtain the real-valued vector representation of each domain name of size 32×120. We used an 80-20 train-test split.



(a) Step 1: Train the Word2vec model to map labels to real-valued vectors.



(b) Step 2: Use the Word2vec model to generate the word embedding of each domain name, which will serve as input for machine learning models.

FIGURE 7. Word embedding: After prepending '*' to each domain name until it has 120 labels, Word2vec is used to generate a real-valued vector representation of 32 × 120 real numbers of each domain name.

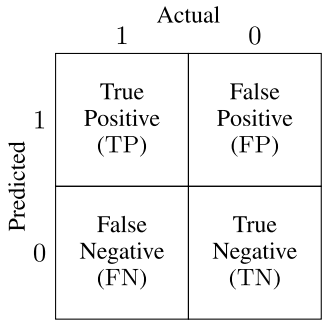


FIGURE 8. The 2 × 2 confusion matrix for binary classifiers.

We train the machine learning models: NB, LR, KNN, SVM, DT, and RF. For each model, we calculate four parameters: Accuracy, precision, recall, and the F_1 score. We first calculate the confusion matrix to identify true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). See Figure 8. The values for our parameters are then computed using the following formulas:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$F_1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{\text{TP}}{\text{TP} + \frac{\text{FN}+\text{FP}}{2}} \tag{4}$$

Accuracy measures the ratio of correct predictions (TP and TN) to all the predictions made by the model, see Equation 1.

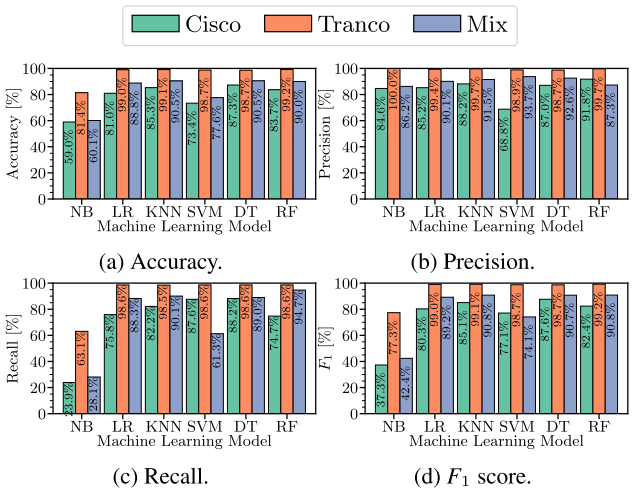


FIGURE 9. Accuracy, precision, recall, and F_1 score of each machine learning model for the top 3953 domain names from the Cisco and Tranco datasets, plus a uniformly sampled Mix of 3953 domain names from the two datasets, each vs. the 3953 domain names from the IoT dataset.

Precision measures the ratio of true positive predictions (TP) to all the positive predictions (TP and FP) made by the model, see Equation 2. Recall measures the ratio of true positive predictions (TP) to all the actual positive instances in the dataset (TP and FN), see Equation 3. Finally, the F_1 score is the harmonic mean of precision and recall, see Equation 4.

The results for using the top 3953 domain names can be seen in Figure 9. For the random selection of domain names, see Figure 12.

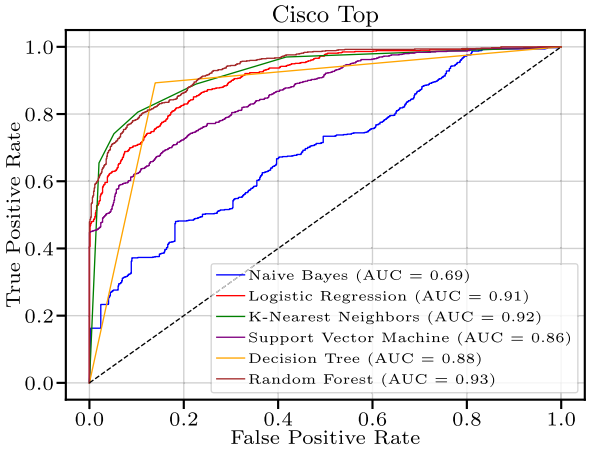


FIGURE 10. Receiver Operation Characteristic (ROC) Curves for the top 3953 domain names from the Cisco dataset. The Area Under the Curve (AUC) is provided in the legend.

1) RESULTS WHEN USING TOP DOMAIN NAMES FROM THE CISCO AND TRANCO DATASETS, AND A MIX OF THE TWO DATASETS

The results of training the models using the top 3953 domain names from the Cisco and Tranco datasets are presented in Figure 9. Each graph represents one of the four parameters obtained by the different models: accuracy, precision, recall, and F_1 score. The six models we trained exhibited the strongest performance when the Tranco dataset was used, achieving, for the four parameters, values $\geq$ 98% for all models but one. The lowest performing model when the Tranco dataset was used was NB, which achieved 81.4% in accuracy, 100% in precision, 63.1% in recall, and 77.3% in F_1 score. On the other hand, the rest of the models showed excellent—and comparable—performances, achieving for RF, for example, 99.2% in accuracy, 99.7% in precision, 98.6% in recall, and 99.2% in F_1 score. The lowest-performing model overall, regardless of the non-IoT dataset used, is NB. NB exhibited a random-classifier-like performance when the Cisco and Mix datasets were used achieving 59.0% and 60.1% in accuracy, 84.6% and 86.2% in precision, 23.9% and 28.1% in recall, and 37.3% and 42.4% in F_1 score for the Cisco and Mix datasets, respectively. The best-performing models are DT and RF. Both models achieved close to 99% in accuracy, precision, recall, and F_1 score when the Tranco dataset was used. In addition, DT achieved 87.3% and 90.5% in accuracy, 87.0% and 92.6% in precision, 88.2% and 89.0% in recall, and 87.6% and 90.7% in F_1 score for the Cisco and Mix datasets, respectively, while RF achieved 83.7% and 90.0% in accuracy, 91.8% and 87.3% in precision, 74.7% and 94.7% in recall, and 82.4% and 90.8% in F_1 score for the Cisco and Mix datasets, respectively. RF is usually preferred between the two models as DT tends to overfit and not perform as well—*i.e.*, it does not generalize well—when exposed to unseen data.

To better compare the best and worst performing models, *i.e.*, NB, DT, and RF, we present their confusion matrices. See

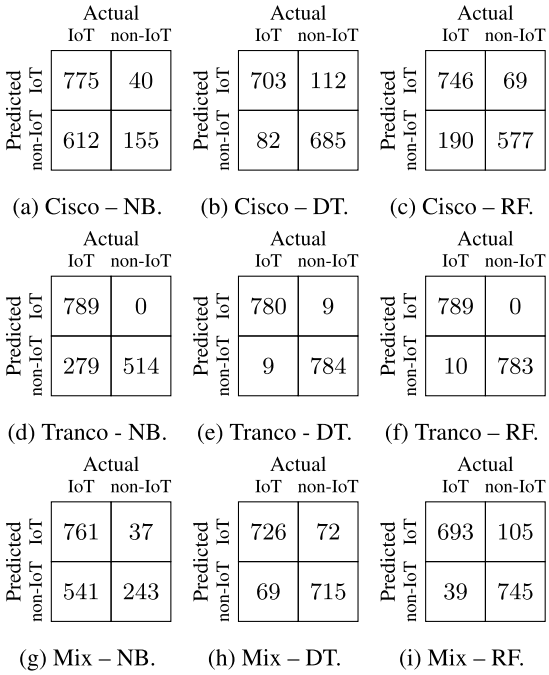


FIGURE 11. Confusion matrices for NB, DT, and RF when trained on the Cisco, Tranco, and Mix datasets, showing the classification performance for IoT and non-IoT domain names.

Figure 11. Among the three models, RF stands out as the best-performing model, outperforming DT, which, as previously mentioned, does not generalize well to unseen data, and NB, which exhibits a random-classifier-like performance. For the Cisco dataset, NB (Figure 11a) misclassifies 612 IoT domain names as non-IoT (FN) and 40 non-IoT domain names as IoT (FP), making it the least effective model. DT (Figure 11b) performs better but still has 112 FP and 82 FN. RF (Figure 11c) achieves the best performance, correctly classifying 746 IoT domain names (TP) but still misclassifying 69 non-IoT domain names as IoT (FP) and 190 IoT domain names as non-IoT (FN).

For the Tranco dataset, RF (Figure 11f) remains superior to NB and DT, correctly classifying 789 IoT domain names (TP) and 783 non-IoT domain names (TN) while achieving 0 (FP) and only 10 (FN). DT (Figure 11e) is not far behind, correctly classifying 780 IoT domain names (TP) and 784 non-IoT domain names (TN) while achieving only 9 (FP) and 9 (FN). NB (Figure 11d) remains the worst-performing model, misclassifying 279 IoT domain names as non-IoT (FN).

Finally, the performance of the models remains consistent for the Mix dataset with NB (Figure 11g) continuing to be the worst-performing model, misclassifying 541 IoT domain names (FN). DT and RF (Figures 11h and 11i) exhibit comparable performance, with DT achieving 72 FP and 69 FN, while RF achieving 105 FP and 39 FN.

As seen in Figure 11, with a few exceptions, the errors we observed were predominantly FN, meaning IoT domain names were incorrectly classified as non-IoT domain names. This could be attributed to IoT devices not exclusively resolving IoT-specific domain names; instead, they sometimes

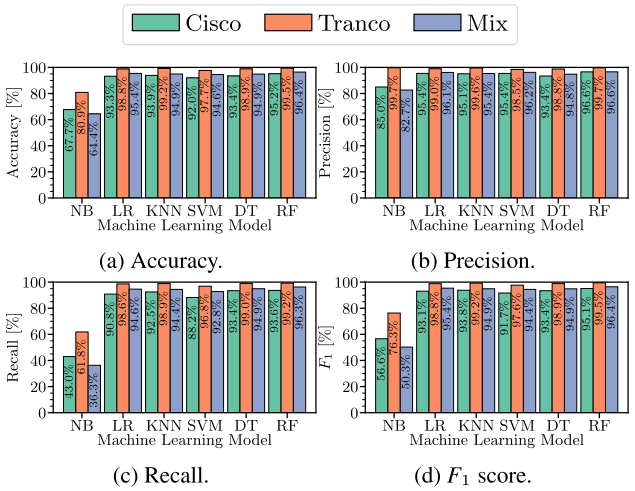


FIGURE 12. Average accuracy, precision, recall and F_1 score of each machine learning model for 100 random selections of 3953 domain names from the Cisco and Tranco datasets, plus a uniformly sampled Mix of 3953 domain names from the two datasets, each vs. the 3953 IoT domain names.

resolve generic domain names used by various devices and users. These domain names lack the distinctive features of IoT-specific domain names and, therefore, are not recognized as such by our models. The overlap between the domain names used by both IoT and non-IoT devices contributes to the most errors in our models. One potential solution would be to introduce an additional *Generic* class to detect IoT domain names that are also visited by generic devices and users. This turns the problem into a multi-class classification task, which is beyond the scope of this work.

The performance of the models is also visualized in Figure 10, which shows the Receiver Operating Characteristic (ROC) curves plotted for every model when the Cisco dataset is used. ROC curves show the performance of the models at different classification thresholds. The performance of the models can be compared by comparing the Area Under the Curve (AUC) of each one. In our case, the ROC curves in Figure 10 further show the superiority of RF compared to the other models where its AUC $= 0.93$. As expected from the previous measurements, NB has the lowest AUC of 0.69 and, therefore, has the lowest performance between the six models.

2) RESULTS WHEN USING RANDOM DOMAIN NAMES FROM THE CISCO AND TRANCO DATASETS, AND A MIX OF THE TWO DATASETS

The results of training the models using randomly selected 3953 domain names from the Cisco and Tranco datasets are presented in Figure 12. Each graph represents one of the four parameters obtained by the different models: accuracy, precision, recall, and F_1 score. We trained the six models by randomly selecting 3953 domain names from each list to generalize our results further. This is particularly interesting, as both the Cisco and Tranco datasets contain nearly 1 million domain names each. For each dataset, 100 random selections of 3953 domain names were made, and the results presented represent the average of the four parameters across these 100 selections.

The results over the 100 random selections are consistent with the previous results obtained when using the top domain names of the Cisco and Tranco datasets. The six models we trained exhibited the strongest performance when the Tranco dataset was used, achieving, for the average of the four parameters, values $\geq$ 96% for all models but one. The lowest performing model when the Tranco dataset was used was NB, which achieved 80.9% in average accuracy, 99.7% in average precision, 61.8% in average recall, and 76.3% in average F_1 score. On the other hand, the rest of the models showed excellent—and comparable—performances, achieving for RF, for example, 99.5% in average accuracy, 99.7% in average precision, 99.2% in average recall, and 99.5% in average F_1 score. The lowest-performing model overall, regardless of the non-IoT dataset used, is NB. NB exhibited a random-classifier-like performance when the Cisco and Mix datasets were used achieving 67.7% and 64.4% in average accuracy, 85.0% and 82.7% in average precision, 43.0% and 36.3% in average recall, and 56.6% and 50.3% in average F_1 score for the Cisco and Mix datasets, respectively. The best-performing models are DT and RF. Both models achieved close to 99% in average accuracy, precision, recall, and F_1 score when the Tranco dataset was used. In addition, DT achieved 93.4% and 94.9% in average accuracy, 93.4% and 94.8% in average precision, 93.4% and 94.9% in average recall, and 93.4% and 94.9% in average F_1 score for the Cisco and Mix datasets, respectively, while RF achieved 95.2% and 96.4% in average accuracy, 96.6% and 96.6% in average precision, 93.6% and 96.3% in average recall, and 95.1% and 96.4% in average F_1 score for the Cisco and Mix datasets, respectively.

B. CROSS VALIDATION

We use cross-validation to assess the robustness of the models and their ability to generalize to unseen data. When assessing a model using cross-validation, the dataset containing all the classes is divided into K folds or subsets, and the model is trained K times. One of the K folds is used as a testing dataset during every training instance, while the remaining $K-1$ are used for training. We use Stratified K-fold cross-validation to ensure that the distribution of classes in the folds is similar to their distribution in the original dataset. Given the size of the IoT dataset, we used $K = 5$ to ensure that each fold contained enough entries to provide a reliable performance estimate. $K = 5$ allows each model to be trained five times. From the Cisco and Tranco datasets, we select the top 3953 domain names, which are added to the 3953 domain names of the IoT dataset. We show the results in Figure 13 as averages and standard deviation values of the evaluation parameters over the five folds.

The colored bars in Figure 13 represent the mean of the four parameters over the five folds, and the error bars at the top of each colored bar represent the standard deviation.

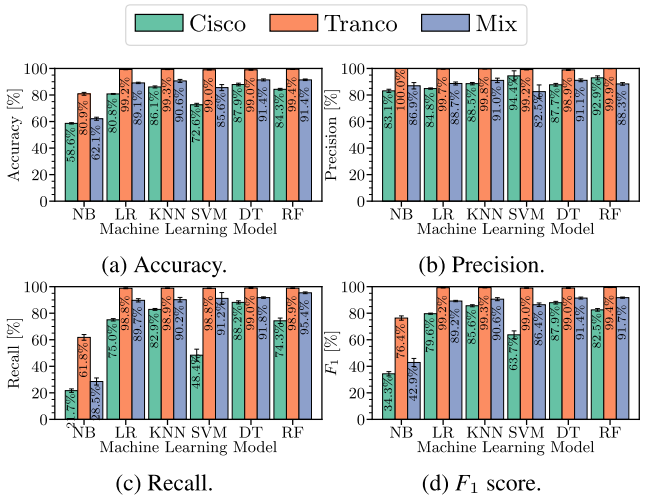


FIGURE 13. Mean (colored bars) and standard deviation (error bars) of accuracy, precision, recall and F_1 score of the ML models over five folds for the top 3953 domain names from the Cisco and Tranco datasets, plus a uniformly sampled Mix of 3953 domain names from the two datasets, vs. the 3953 IoT domain names.

We notice that the means of the four parameters over the five folds are consistent with the results from the performance evaluation we performed in Section IX-A while having a low standard deviation, which indicates that the models are stable across the folds and that they are likely to generalize well to unseen data.

C. ABLATION TEST

An ablation test includes removing elements from the machine learning model or suppressing a subset of the features to study their possible effect on performance. We perform the ablation test by removing one label at a time by replacing the 32-dimensional vector representing the label with zeros. Since we padded each domain name up to 120 labels, we conduct 120 training and testing sessions, ablating one label from both the training and testing datasets each time before training and evaluating the models. The results for RF when used with the Cisco and Tranco datasets are presented in Figures 14a and 14b.

For both figures, the stable performance observed when ablating the dummy labels (`'*'`) demonstrates that the padding we added to the left of each domain name held no information and did not alter the performance of the models. The performance begins to decline as the last labels (starting from label 115) are ablated. The most significant drop in performance occurs when label 119 is ablated. The second-to-last label, *i.e.*, the second-level domain (label 119), appears to have the highest impact on the performance as the values drastically dropped in both figures. For example, the accuracy, recall and F_1 score in Figure 14 dropped by around 15 and 25 percent, respectively. When the last label—the TLD of the domain name (label 120)—was ablated, however, the values of the parameters did not experience the same drastic decrease, and the effect of ablating label 120 seemed almost equivalent to ablating the dummy labels. This shows that the second-level domain of a domain name is the most indicative of its class and that third-level domains and above gradually become less informing about the class of a domain name until ablating them becomes equivalent to ablating a dummy label. Moreover, the TLD of a domain name also does not seem to provide significant information about the class of the domain name, as ablating it (label 120) only slightly affected the performance.

X. DISCUSSION

A. THE SIZE OF THE IoT DATASET

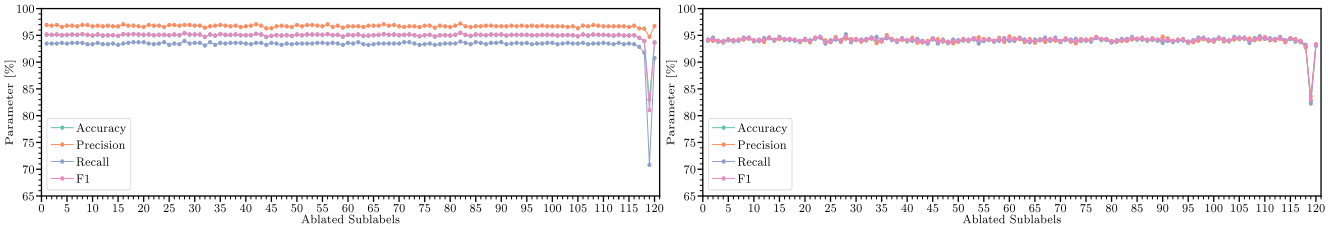
Despite the large amounts of raw data we started with, the size of the IoT dataset remained relatively modest. This is primarily due to the scope of our study, which mainly covers IoT devices that engage in machine-to-machine communications. Such devices exhibit limitations in their functionalities compared to generic devices and IoT devices that are not strictly machine-to-machine. This explains the low number of servers on the Internet these devices contact. Hence, we noticed a low number of frequently contacted servers instead of numerous servers that are less regularly or rarely contacted.

B. USING TOP-LISTS AS NON-IoT DATASETS

We used two known lists of top-visited websites, namely the Cisco and Tranco datasets. We noticed that the domain names in the Tranco dataset are less representative of real non-IoT domain names, as most of them are second-level domains, which does not reflect how domain names appear in DNS traffic. The Cisco dataset, however, is suitable as its entries are not limited to second-level domains and are included in the dataset as seen in DNS traffic. The Cisco dataset also statistically resembles the domain names in the IoT dataset. The difference between the Cisco and Tranco datasets was most visible when training and testing the machine learning models. The Tranco dataset is easily distinguishable, so the models almost achieved perfect scores. The performance was different with the Cisco dataset, which achieved slightly lower, but arguably more realistic, performance. Since it is more representative of real-world domain names, we find that the Cisco dataset is the better option for use as a negative class in domain name classification problems with similar contexts.

C. EXTENDED SECURITY CONSIDERATIONS

While our study focuses solely on analyzing IoT domain names, providing some insights into IoT security, future research could explore additional security measures currently being implemented and standardized to enhance various aspects of network infrastructure, authentication mechanisms, and DNS, ultimately contributing to IoT security. In our DNS analysis, we observed a low adoption of security-related resource records in IoT domain names, such as DNSSEC-related records, highlighting a notable security



(a) Cisco. (b) Tranco.

FIGURE 14. Ablation test with Random Forest (RF).

limitation in IoT environments. However, in addition to the importance of wider adoption of these security mechanisms, their effectiveness could be augmented through modern cryptographic techniques [63]. For example, post-quantum Key Encapsulation Mechanisms (KEMs) like Kyber [64] and signature algorithms such as Dilithium [65] could fortify DNSSEC against emerging quantum threats while maintaining acceptable computational efficiency. No discussion or effort toward IoT security should overlook these essential tools.

Moreover, while machine learning techniques are frequently employed to distinguish between phishing and benign domain names, enhancing security, the models themselves could serve as potential backdoors for threats. Attackers may attempt poisoning attacks by injecting malicious entries into the training datasets or could craft adversarial inputs to evade detection by studying the feature extraction techniques used. These risks require careful attention to data preprocessing and sanitization, the use of reliable anomaly detection mechanisms, and choosing robust models that fortify the machine learning pipeline against such threats. Finally, the intersection of cryptography and fault tolerance is another promising direction, as side-channel attacks (SCAs) and fault-injection methods may compromise cryptographic operations at the DNS or machine learning model level, necessitating countermeasures like masked implementations and anomaly detection.

XI. LIMITATIONS OF THE STUDY

A. CHOICE OF IoT AND NON-IoT DATASETS

We constructed the IoT dataset using packet captures from 12 past studies that included testbeds of real IoT devices. The goal of using multiple data sources was to expand the range of IoT devices represented in the final dataset while also increasing its overall size. This helped improve the robustness of machine learning models and their ability to generalize to unseen data. However, the final dataset still leaned toward smart home IoT devices, as most fell under this category to some extent. In terms of size, using the 4,145 unique IoT domain names, the machine learning models demonstrated promising performance and good generalizability. That said, increasing the size and diversity of the dataset—making it more representative of the IoT ecosystem—would further enhance the robustness of the trained models and improve their ability to generalize even more effectively.

For the non-IoT datasets, we used two lists of top-visited websites. Such lists are commonly used by researchers in domain name classification problems, particularly in detecting DGA-generated or phishing domain names, and are typically treated as the negative class or benign domain names. While these lists contain the most popular websites and are considered reliable in this context, they will inevitably miss newly registered domain names and emerging trends. This limitation could be addressed by further diversifying the sources of non-IoT domain names and incorporating newly registered and lower-ranked existing ones through lists that capture these evolving trends.

B. REAL-WORLD DEPLOYMENT

We trained the classification models using a balanced dataset, evenly split between IoT and non-IoT domain names. This approach is ideal for machine learning models, as it helps prevent overfitting and bias toward one class. In real-world scenarios, however, such balance is rarely observed, as one class of domain names often dominates depending on the nature of the network. Furthermore, the trained models should not be relied upon indefinitely, as trends in IoT and non-IoT domain names change constantly. To remain effective, the models should support continuous online training and regular updates to their parameters.

XII. CONCLUSION AND FUTURE OUTLOOK

In this paper, we conducted a study on domain names of IoT backend servers. We constructed a dataset of IoT domain names using 12 public datasets of network traffic from past studies that included testbeds of real IoT devices. To contrast IoT domain names with non-IoT ones, we used two lists of top-visited websites, Cisco Umbrella and Tranco, as non-IoT datasets. Our study followed a three-phase comparative approach: analyzing the structural and statistical differences between IoT and non-IoT domain names, evaluating their DNS properties, and assessing the effectiveness of machine learning models in classifying IoT and non-IoT domain names. Our results showed that IoT domain names exhibit differences in their statistical properties compared to non-IoT domain names, and they usually contain indicative keywords—such as protocol- or technology-related terms—that distinguish them from non-IoT domain names. In addition, the DNS zones associated with IoT domain names exhibited less dynamism and a reduced presence

of security-related resource records compared to non-IoT domain names. As for classification using machine learning, the models we trained demonstrated high effectiveness in distinguishing between the two classes of domain names with Random Forest achieving the best overall performance in terms of accuracy, precision, recall, and F_1 score while exhibiting minimal overfitting. In addition, the models exhibited the ability to generalize well to unseen data. The ablation test demonstrated that the second-level domain is the most indicative part of the domain name for determining its class.

Looking forward, the paths to improve this work include, in the short term, increasing the size of the IoT dataset to make it more representative. While the current IoT dataset is a good representation of the devices that users interact with daily—such as those found in smart homes—increasing the size and diversity of the dataset would generalize our findings and make them more inclusive of the overall IoT landscape. This expansion would involve incorporating network traffic from a broader range of devices, including wearables, smart city infrastructure, Internet of Medical Things (IoMT) devices, and Industrial Internet of Things (IIoT) systems. In addition, for domain name classification, we aim to train our models against lists of known malicious domain names and domain names generated by DGAs to enhance the security aspect of our classifier. Future work may also include applying different word embedding and feature extraction techniques. Furthermore, we plan to explore the capabilities of deep learning-based models in classifying IoT and non-IoT domain names, as these models promise robust performance. Deep learning techniques also facilitate more easily explainable decisions through eXplainable AI (XAI), providing deeper insight into model behavior and clarifying why a domain name was classified into a particular class.

REFERENCES

[1] S. J. Saidi, S. Matic, O. Gasser, G. Smaragdakis, and A. Feldmann, "Deep dive into the IoT backend ecosystem," in *Proc. 22nd ACM Internet Meas. Conf.*, Oct. 2022, pp. 488–503, doi: 10.1145/3517745.3561431.

[2] S. Kumar Svn, S. Munuswamy, and K. Arputharaj, "A comprehensive survey on machine learning-based intrusion detection systems for secure communication in Internet of Things," *Comput. Intell. Neurosci.*, vol. 2023, pp. 1–24, Jan. 2023., doi: 10.1155/2023/8981988.

[3] H. Xu, Z. Sun, Y. Cao, and H. Bilal, "A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things," *Soft Comput.*, vol. 27, no. 19, pp. 14469–14481, Jul. 2023, doi: 10.1007/s00500-023-09037-4.

[4] H. Asgharzadeh, A. Ghaffari, M. Masdari, and F. S. Gharehchopogh, "Anomaly-based intrusion detection system in the Internet of Things using a convolutional neural network and multi-objective enhanced capuchin search algorithm," *J. Parallel Distrib. Comput.*, vol. 175, pp. 1–21, May 2023, doi: 10.1016/j.jpdc.2022.12.009.

[5] A. Ahmim, F. Maazouzi, M. Ahmim, S. Namane, and I. B. Dhaou, "Distributed denial of service attack detection for the Internet of Things using hybrid deep learning model," *IEEE Access*, vol. 11, pp. 119862–119875, 2023, doi: 10.1109/access.2023.3327620.

[6] G.-P. Fernando, A.-A.-H. Brayan, A. M. Florina, C.-B. Liliana, A.-M. Héctor-Gabriel, and T.-S. Reinel, "Enhancing intrusion detection in IoT communications through ML model generalization with a new dataset (IDSAI)," *IEEE Access*, vol. 11, pp. 70542–70559, 2023, doi: 10.1109/access.2023.3292267.

[7] N. Moustafa, N. Koroniotis, M. Keshk, A. Y. Zomaya, and Z. Tari, "Explainable intrusion detection for cyber defences in the Internet of Things: Opportunities and solutions," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 3, pp. 1775–1807, 3rd Quart., 2023, doi: 10.1109/comst.2023.3280465.

[8] M. Karthikeyan, D. Manimegalai, and K. RajaGopal, "Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection," *Sci. Rep.*, vol. 14, no. 1, pp. 1–10, Jan. 2024, doi: 10.1038/s41598-023-50554-x.

[9] N. Charef, A. Ben Mnaouer, M. Aloqaily, O. Bouachir, and M. Guizani, "Artificial intelligence implication on energy sustainability in Internet of Things: A survey," *Inf. Process. Manage.*, vol. 60, no. 2, Mar. 2023, Art. no. 103212, doi: 10.1016/j.ipm.2022.103212.

[10] A. K. Sangaiah, A. Javadpour, F. Ja'fari, H. Zavieh, and S. M. Khaniabadi, "SALA-IoT: Self-reduced Internet of Things with learning automaton sleep scheduling algorithm," *IEEE Sensors J.*, vol. 23, no. 18, pp. 20737–20744, Sep. 2023, doi: 10.1109/jsen.2023.3242759.

[11] D. Gao, H. Wang, X. Guo, L. Wang, G. Gui, W. Wang, Z. Yin, S. Wang, Y. Liu, and T. He, "Federated learning based on CTC for heterogeneous Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 22673–22685, Dec. 2023, doi: 10.1109/jiot.2023.3305189.

[12] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1251–1275, 2nd Quart., 2020, doi: 10.1109/comst.2020.2964534.

[13] I. Cvitić, D. Peraković, M. Perića, and B. Gupta, "Ensemble machine learning approach for classification of IoT devices in smart home," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 11, pp. 3179–3202, Nov. 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:230109895

[14] Y. Wang, R. Pan, Z. Wang, and L. Li, "A classification method based on CNN-BiLSTM for difficult detecting DGA domain name," in *Proc. 2023 IEEE 13th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2023, pp. 17–21, doi: 10.1109/iceiec58029.2023.10200702.

[15] E. Zhu, Q. Yuan, Z. Chen, X. Li, and X. Fang, "CCBLA: A lightweight phishing detection model based on CNN, BiLSTM, and attention mechanism," *Cognit. Comput.*, vol. 15, no. 4, pp. 1320–1333, May 2022, doi: 10.1007/s12559-022-10024-4.

[16] H. Yang, T. Zhang, Z. Hu, L. Zhang, and X. Cheng, "A DGA domain name detection method based on two-stage feature reinforcement," in *Proc. IEEE 22nd Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Nov. 2023, pp. 652–659, doi: 10.1109/TRUSTCOM60117.2023.00099.

[17] Y. Wang, W. Zhu, H. Xu, Z. Qin, K. Ren, and W. Ma, "A large-scale pretrained deep model for phishing URL detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10095719.

[18] S. U. M. Rao, V. R. Babu, C. H. Divya, C. R. Naidu, G. J. Moses, and A. Lakshmanarao, "A novel approach to DGA detection combining machine learning and transformer-based techniques," in *Proc. Int. Conf. IoT Based Control Netw. Intell. Syst. (ICICNIS)*, Dec. 2024, pp. 1416–1420, doi: 10.1109/icicnis64247.2024.10823341.

[19] H. Lee, J. Do Yoo, S. Jeong, and H. K. Kim, "Detecting domain names generated by DGAs with low false positives in Chinese domain names," *IEEE Access*, vol. 12, pp. 123716–123730, 2024, doi: 10.1109/access.2024.3454242.

[20] L. Yang, G. Liu, J. Wang, H. Bai, J. Zhai, and Y. Dai, "Fast3DS: A real-time full-convolutional malicious domain name detection system," *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102933, doi: 10.1016/j.jisa.2021.102933.

[21] C. Ahmadi and J.-L. Chen, "Enhancing phishing detection: A multi-layer ensemble approach integrating machine learning for robust cybersecurity," in *Proc. 2024 IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2024, pp. 1–6, doi: 10.1109/iscc61673.2024.10733689.

[22] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," 2016, *arXiv:1611.00791*.

[23] Y. Qiao, B. Zhang, W. Zhang, A. K. Sangaiah, and H. Wu, "DGA domain name classification method based on long short-term memory with attention mechanism," *Appl. Sci.*, vol. 9, no. 20, p. 4205, Oct. 2019, doi: 10.3390/app9204205.

[24] R. S. Rao, T. Vaishnavi, and A. R. Pais, "CatchPhish: Detection of phishing websites by inspecting URLs," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 2, pp. 813–825, Feb. 2020, doi: 10.1007/s12652-019-01311-4.

[25] Cisco Umbrella. *Cisco Umbrella 1 Million*. Accessed: Jul. 2024. [Online]. Available: https://s3-us-west-1.amazonaws.com/umbrella-static/index.html

[26] Tranco. *Tranco*. Accessed: Jul. 2024. [Online]. Available: https://tranco-list.eu/list/K273W

[27] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Feb. 2019, pp. 1–15, doi: 10.14722/ndss.2019.23386.

[28] M. S. Lenders, C. Bormann, T. C. Schmidt, and M. Wahlisch. (Nov. 2024). *A Concise Binary Object Representation (CBOR) of DNS Messages*. Internet Eng. Task Force. [Online]. Available: https://datatracker.ietf.org/doc/draft-lendersdns-cbor/10/

[29] H. Shirazi, S. R. Muramudalige, I. Ray, A. P. Jayasumana, and H. Wang, "Adversarial autoencoder data synthesis for enhancing machine learning-based phishing detection algorithms," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 2411–2422, Jul. 2023, doi: 10.1109/tsc.2023.3234806.

[30] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing detection system through hybrid machine learning based on URL," *IEEE Access*, vol. 11, pp. 36805–36822, 2023, doi: 10.1109/access.2023.3252366.

[31] C. Hesselman, M. Kaeo, L. Chapin, K. Claffy, M. Seiden, D. McPherson, D. Piscitello, A. McConachie, T. April, J. Latour, and R. Rasmussen, "The DNS in IoT: Opportunities, risks, and challenges," *IEEE Internet Comput.*, vol. 24, no. 4, pp. 23–32, Jul. 2020, doi: 10.1109/mic.2020.3005388.

[32] M. Liu and L. Yang, "IoT network traffic analysis with deep learning," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops Other Affiliated Events (PerCom Workshops)*, Mar. 2024, pp. 184–189, doi: 10.1109/percomworkshops59983.2024.10502498.

[33] B. M. Schwartz, M. Bishop, and E. Nygren, *Service Binding and Parameter Specification Via the DNS (SVCB and HTTPS Resource Records)*, document RFC 9460, Nov. 2023, doi: 10.17487/RFC9460. [Online]. Available: https://www.rfc-editor.org/info/rfc9460

[34] C. Amsuss and M. S. Lenders. (Oct. 2024). *CoAP Transport Indication*. Internet Eng. Task Force. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-coretransport-indication/07/

[35] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," *IEEE Access*, vol. 10, pp. 36429–36463, 2022, doi: 10.1109/access.2022.3151903.

[36] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: A systematic literature review," *Knowl. Inf. Syst.*, vol. 64, no. 6, pp. 1457–1500, Jun. 2022, doi: 10.1007/s10115-022-01672-x.

[37] S. Srinivasan, R. Vinayakumar, A. Arunachalam, M. Alazab, and K. Soman, "DURLD: Malicious URL detection using deep learning-based character level representations," in *Malware Analysis Using Artificial Intelligence and Deep Learning*. Cham, Switzerland: Springer, Dec. 2020, pp. 535–554, doi: 10.1007/978-3-030-62582-5_21.

[38] A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards lightweight URL-based phishing detection," *Future Internet*, vol. 13, no. 6, p. 154, Jun. 2021, doi: 10.3390/fi13060154.

[39] S. Raschka, "Naive Bayes and text classification I–introduction and theory," 2014, *arXiv:1410.5329*.

[40] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Cham, Switzerland: Springer, 2009, doi: 10.1007/978-0-387-21606-5.

[41] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers—A tutorial," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–25, Jul. 2021, doi: 10.1145/3459665.

[42] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998, doi: 10.1109/5254.708428.

[43] Y. Izza, A. Ignatiev, and J. Marques-Silva, "On explaining decision trees," 2020, *arXiv:2010.11034*.

[44] Y. Mansour and M. Schain, "Random Forests," *Mach. Learn.*, vol. 45, no. 2, pp. 123–145, Oct. 2001, doi: 10.1023/a:1010950718922.

[45] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," 2013, *arXiv.1309.4168*.

[46] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Workshop ICLR*, Jan. 2013, pp. 1–12.

[47] R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis, "IoTFinder: Efficient large-scale identification of IoT devices via passive DNS traffic analysis," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Sep. 2020, pp. 474–489, doi: 10.1109/eurosp48549.2020.00037.

[48] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "SoK: Security evaluation of home-based IoT deployments," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, p. 1362, doi: 10.1109/SP.2019.00013.

[49] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information exposure from consumer IoT devices: A multidimensional, network-informed measurement approach," in *Proc. Internet Meas. Conf.*, Oct. 2019, pp. 267–279, doi: 10.1145/3355369.3355577.

[50] M. T. Paracha, D. J. Dubois, N. Vallina-Rodriguez, and D. Choffnes, "IoTLS: Understanding TLS usage in consumer IoT devices," in *Proc. 21st ACM Internet Meas. Conf.*, Nov. 2021, pp. 165–178, doi: 10.1145/3487552.3487830.

[51] ANT Lab. *IoT Devices' First-Time Bootup Traces, PREDICT ID: USC-LANDER/IoT_Bootup_Traces-20161207*. Accessed: Mar. 2023. [Online]. Available: http://www.isi.edu/ant/lander

[52] ANT Lab. *IoT Devices' First-Time Bootup Traces, PREDICT ID: USC-LANDER/IoT_Bootup_Traces-20181107*. Accessed: Mar. 2023. [Online]. Available: http://www.isi.edu/ant/lander

[53] ANT Lab. *10-day Operantional IoT Traces, PREDICT ID: USC-LANDER/IoT_Operation_Traces-20200127*. Accessed: Mar. 2023. [Online]. Available: http://www.isi.edu/ant/lander

[54] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[55] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated device-type identification for security enforcement in IoT," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 2177–2184, doi: 10.1109/ICDCS.2017.283.

[56] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, "IoT network intrusion dataset," *IEEE Dataport*, 2019, doi: 10.21227/q70p-q449.

[57] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1745–1759, Aug. 2019, doi: 10.1109/tmc.2018.2866249.

[58] A. Hamza, H. H. Gharakheili, T. A. Benson, and V. Sivaraman, "Detecting volumetric attacks on loT devices via SDN-based monitoring of MUD activity," in *Proc. ACM Symp. SDN Res.*, San Jose, CA, USA, Apr. 2019, pp. 36–48, doi: 10.1145/3314148.3314352.

[59] *Domain Names–implementation and Specification*, document RFC 1035, Nov. 1987, doi: 10.17487/RFC1035. [Online]. Available: https://www.rfc-editor.org/info/rfc1035

[60] O. Garcia-Morchon, S. Kumar, and M. Sethi, *Internet of Things (IoT) Security: State of the Art and Challenges*, document RFC 8576, Apr. 2019, doi: 10.17487/RFC8576. [Online]. Available: https://www.rfc-editor.org/info/

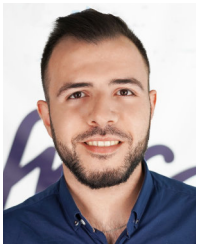
[61] M. S. Lenders, C. Amsuss, C. Gündogan, T. C. Schmidt, and M. Wahlisch. (Feb. 2025). *DNS Over CoAP (DoC)*. Internet Eng. Task Force. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-core-dns-overcoap/12/

[62] Afnic Swedish Internet Found. *Zonemaster: Requirements and Normalization of Domain Names in Input*. Accessed: Jul. 2024. [Online]. Available: https://github.com/zonemaster/zonemaster/blob/4ae8a6e/docs/specifications/tests/RequirementsAndNormalizationOfDomainNames.md

[63] A. Fregly, R. van Rijswijk-Deij, M. M′uller, P. Thomassen, C. Schutijser, and T. Chung. (Dec. 2024). *Research Agenda for a Post-Quantum DNSSEC*. Internet Eng. Task Force. [Online]. Available: https://datatracker.ietf.org/doc/draft-fregly-research-agendafor-pqc-dnssec/02/

[64] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, "CRYSTALS–kyber: A CCA-secure module-lattice-based KEM," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Apr. 2018, pp. 353–367, doi: 10.1109/EUROSP.2018.00032.

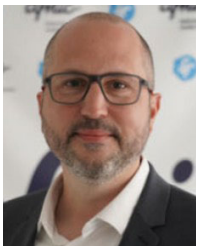
[65] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystalsdilithium: A lattice-based digital signature scheme," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2018, no. 1, pp. 238–268, Feb. 2018, doi: 10.13154/tches.v2018.i1.238-268. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/839



IBRAHIM AYOUB received the degree in computer and communications engineering from the Faculty of Engineering, Lebanese University, in 2016, and the master's degree from the École Supérieure d'Ingénieurs de Beyrouth (ESIB), in 2021. He is currently pursuing the integrated Ph.D. degree with the Association Française pour le Nommage Internet en Coopération (Afnic) and Université Paris-Saclay, Saint-Quentin-en-Yvelines, France. He was with GlobalCom Holding as an IP/MPLS Engineer, from 2017 to 2021. His research interests include cyber security and the IoT.



MARTINE S. LENDERS received the B.Sc. and M.Sc. degrees in computer science from Freie Universität Berlin (FU Berlin), where she is currently pursuing the Ph.D. degree. She is a Research Associate with the Chair of Distributed and Network Systems, Technische Universität Dresden (TU Dresden), supervised by Prof. Dr. Matthias Wählisch. She is particularly interested in privacy-friendly name resolution in constrained environments and what we can learn from those environments for the big internet. Through her engagement in all things IoT and network protocols, she has been active in the Internet Engineering Task Force (IETF), since 2015, and is the co-author on several Internet Drafts regarding secure name resolution in constrained networks. Her research interest includes networking and programming for the Internet of Things (IoT).



BENOÎT AMPEAU received the Graduate Engineering Diploma degree in information systems management from the University of Technology of Troyes. He began his career in the industry and then joined a consulting company as a Partner and Strategic Business Unit Manager, in 2005. In 2013, he joined the Technical Department, Engineering and Development Department, Association Française pour le Nommage Internet en Coopération (Afnic), before leading Afnic Labs and coordinating the AFNIC Scientific Council, in 2016. Appointed in Spring 2018 as the Director of Partnerships and Innovation, he is developing research projects with institutional and academic partners. His strategic priorities in research and development and knowledge transfer are focusing on leveraging the DNS infrastructure and protocol in its capability to increase connectivity, interoperability, scalability, and security and privacy on various typologies of networks: internet, LPWAN, and edge computing architectures.



SANDOCHE BALAKRICHENAN received the Ph.D. degree in computer science and networks from the Université Pierre-et-Marie-Curie. He is currently the Head of the Research and Development Partnerships, Association Française pour le Nommage Internet en Coopération (Afnic). He has been an Invited IoT Expert at the European Commission representing the European ccTLD community, the IoT Expert Reviewer at Cap Digital, the RIPE IoT Working Group Co-Chair, and currently the LoRa Alliance Academic Working Group Chair. He actively participates/contributes to standardization and associated organizations, such as GS1, LoRa-alliance, IETF, RIPE, and AIOTI. He is an Advisor for Ph.D. students. He has worked on DNS and distributed systems for 19 years, publishing six journal articles and 16 peer-reviewed articles, with one of the best papers on distributed systems, the IoT, and performance evaluation. His research interests include networked computer systems, the IoT identity management, and security and privacy.



KINDA KHAWAM received the engineering degree from the École Supérieure d'Ingénieurs de Beyrouth (ESIB), in 2002, and the master's and Ph.D. degrees in computer networks from Telecom ParisTech, Paris, France, in 2003 and 2006, respectively. She was a Postdoctoral Fellow Researcher with France Télécom, Issy-Les-Moulineau, France, in 2007. She is currently an Associate Professor and a Researcher with the Université de Versailles Saint-Quentin-en-Yvelines, France. Her research interests include radio resource management, modeling and performance evaluation of mobile networks, and the IoT.



THOMAS C. SCHMIDT (Member, IEEE) received the Ph.D. degree in mathematical physics from FU Berlin. He is currently a Professor of computer networks and internet technologies with Hamburg University of Applied Sciences (HAW Hamburg), where he heads the Internet Technologies Research Group. Before moving to Hamburg, he was the Director of the Scientific Computer Centre, Berlin. Since then, he has continuously conducted numerous national and international research projects. He was the Principal Investigator in a number of EU, nationally funded, and industrial projects as well as a Visiting Professor with the University of Reading, U.K. He is the Co-Founder of several large open-source projects and a Coordinator of the community developing the RIOT operating system—the friendly OS for the Internet of Things (IoT). His research interests include the development, measurement, and analysis of large-scale distributed systems like the Internet. He serves as a co-editor and a technical expert on many occasions and he is actively involved in the work of IETF and IRTF. Together with his group, he pioneered work on an information-centric industrial IoT and the emerging data-centric Web of Things.



MATTHIAS WÄHLISCH (Member, IEEE) currently holds the Chair of Distributed and Networked Systems, Technische Universität Dresden. He is also a Research Fellow of the Barkhausen Institut. His research interests include scalable, reliable, and secure internet communication. This includes the design and evaluation of networking protocols and architectures, as well as internet measurements and analysis. He has been involved in the IETF, since 2005, and co-founded multiple successful open-source projects.