2020 12th International Conference on Cyber Conflict
20/20 Vision: The Next Decade
T. Jančárková, L. Lindström,
M. Signoretti, I. Tolga, G. Visky (Eds.)
2020 © NATO CCDCOE Publications, Tallinn

Towards Classifying Devices on the Internet Using Artificial Intelligence

Artūrs Lavrenovs
NATO CCD COE
Tallinn, Estonia
arturs.lavrenovs@ccdcoe.org

Roman Graf
AIT Austrian Institute of Technology
Vienna, Austria
roman.graf@ait.ac.at

Kimmo Heinäaro
NATO CCD COE
Tallinn, Estonia
kimmo.heinaaro@ccdcoe.org

Abstract: Hundreds of millions of devices are directly reachable by anyone on the Internet. Security researchers and malicious actors are highly interested in ICS, IoT, and building automation and networking devices that can be compromised to negatively affect either a specific person or organization or a whole country at once. The current approach for determining a class of individual device is to conduct a manual investigation or apply static rules to large sets of devices, which is time-consuming and ineffective. We are proposing to utilize neural networks for automated classification.

Many devices have a generic web interface supporting HTTP protocol. We have investigated which features of the HTTP responses from these devices are meaningful for training the neural network model and enabling classification of devices. We have trained neural network models and assessed their accuracy to be 87%. We are analysing the classified sets of the whole Internet scans consisting of tens of millions of devices and comparing them between the years 2018 and 2019 to identify the changes. This kind of all-encompassing view might reveal positive and negative trends that are

happening to specific classes of devices, which might be correlated with real-world events, e.g. new policies issued by governments.

Keywords: *devices on the Internet, classifying devices, machine learning, neural network*

1. INTRODUCTION

Billions of different devices are connected to the Internet and predictions for the next decade expect geometric growth. Statista projects that there will be 75 billion IoT devices by 2025 [1]. The way these devices are connected to networks varies, and only a small portion of all devices on the Internet are publicly reachable by anyone. Unsophisticated actors can access, abuse and exploit reachable devices with known vulnerabilities. Understanding the potential risks and corresponding impacts, or assessing the current state, requires knowledge of classes of devices and their location. Academic and technical research can benefit from this understanding, and it can also provide sufficient background to help policymakers address security concerns regarding these devices.

Identification and classification of reachable devices on the Internet has traditionally been a straightforward process. The targeted protocol port gets tested to check it is open, and possibly a protocol payload is sent and the response processed. Depending on the case, the investigation stops here or continues with additional protocol requests that extract the properties of the devices, possibly identifying the manufacturer or model. If different classes of devices use the targeted port, then classification can be attempted using static rules. Heterogeneity of devices has grown over time, and it has become unfeasible to achieve a high coverage and accuracy rate when classifying large sets of devices. We are attempting to solve this problem by creating a neural network that replaces the static rule stage in the network research.

In Chapter 2, we explore what kind of devices are available on the Internet and why, as well as how they can be classified. Chapter 3 describes our application of machine learning to solve the device classification problem. Chapter 4 explores the results of the classification and compares them between standard and alternative HTTP ports between the years 2018 and 2019. Conclusions and future work are discussed in Chapter 5.

2. DEVICES ON THE INTERNET

In this research, we are attempting to begin to ask what exactly is on the Internet and what the risks are. We are only investigating devices that are reachable on the Internet – reachable meaning that the device receives, processes, and responds to network packets coming from anywhere on the Internet. In general, these packets target specific ports corresponding to known and common protocols. Only a small fraction of all the devices on the Internet are reachable in this way.

A significant number of different services are required to be reachable on the Internet for anyone in order to function properly, e.g., web sites on HTTP and HTTPS, authoritative DNS. Some services are required only for use in a home, office or ISP local network, e.g. DNS resolver, UPnP discovery. The core issue is that the number of devices that are reachable on the Internet far outweighs the number that is required. The leading causes of unnecessary reachable devices are manufacturers' default configurations and network misconfiguration while installing a device.

Reachability significantly increases the attack surface of these devices. Some services can be abused by default, e.g. DNS resolver without rate-limiting for reflected DDoS attacks. Some devices are entirely unprotected while others might contain a publicly known vulnerability that an attacker has to exploit. Depending on the vulnerability, the attacker might achieve a different level of access, from leaking insignificant information up to full control of the device. Depending on the class of the device, the impact of the compromisation can vary drastically. A compromised ICS device can interrupt essential services to vast regions, affecting millions of people, while an unprotected printer might only waste printing toner, causing inconvenience to a single person.

Even if there are protection mechanisms in place like authentication, no immediately abusable services and no known exploitable vulnerabilities, the risk that new vulnerabilities can be discovered in future is ongoing. Unnecessary reachability is already an indication of poor device management practices. No security updates for most devices is the norm; many of the newly installed devices are left untouched until the end of their life for as long as they serve the required purpose.

There are a variety of protocols worthy of investigation for classification. In this research, we are only focusing on the HTTP protocol being utilized on standard port 80 and common alternative port 8080. Implementing a web control panel utilizing HTTP protocol is the cheapest and easiest way that manufacturers can provide a control interface for a device being sold to consumers. This is a ubiquitous protocol supported by every investigated device class, justifying the choice.

A. Classifying Reachable Devices

Multiple approaches suitable for classification of remotely reachable devices exist, but they can all be reduced to acquiring properties of devices and applying a set of static rules to them. The most common property is a check to verify if a specific port or range of ports is open. After this check, port-specific negotiations can occur, and additional information, varying drastically in quality and quantity, can be acquired. At the very least, it can be confirmed if the specific device on the specific port supports the tested protocol. In best-case scenarios, the manufacturer, model, version and even location and purpose of the device can be determined.

After possible properties are acquired and investigated, rules can be developed to match these properties and to locate all matching devices in large data sets, e.g. a full Internet scan. These rules can be something as simple as a unique and rare port being open, up to matching the manufacturer and model returned in the response. These rules are made by humans and usually target common or high impact devices. As many devices require thorough manual investigation to classify them, it is unfeasible that full coverage can be achieved. This is the most common approach for classifying devices in academic and industry research, including device search engines such as Shodan and Censys.

Additional properties can be gathered indirectly by fingerprinting the scanning and communication process or independently by identifying a network, its location and DNS name. These properties are primarily used in the manual investigation of the individual devices and rarely for creating static rules because of the high variability of this data.

This approach has a major drawback. It works perfectly for locating a specific subset of a specific device class using its properties and their values, which are known in advance and were acquired through manual investigation. But what happens when there is a large set of devices or even a single device that has to be classified? The set of available static rules can be applied to it, and there might be a match; in that case, there is no issue. However, if there is no match, then the device is left unclassified and requires manual investigation, which is time-consuming and does not guarantee success. Utilizing a machine learning classifier can solve these types of questions.

B. Related Work

Until recently, classifying devices on the Internet was done in a static way (described in 2.A) both for academic research and industry purposes. Only in recent years have researchers attempted to address this issue using machine learning. Two vantage points are being investigated: reachable device classification using data sets from Internet scanning, and device classification using an observer data set, which includes

all communicating devices, including non-reachable ones. The latter does not provide a full Internet view but provides highly valuable information for internal networks where observer access is possible.

The observer's vantage point enables data to be gathered over long periods of time, from which behavioral profiles can be created. It is also possible to create profiles without decoding the appropriate protocols, and in some cases, it is even impossible because of encryption, e.g., HTTPS. These profiles allow not only the classification of devices but also the identification of misbehaving compromised devices. Sivanathan et al. created a classifier based on existing campus network data that was able to distinguish IoT and non-IoT devices [2]. Bezawada et al. acquired fingerprints from different levels of the same network traffic and combined these into behavioral profiles suitable for machine learning [3].

Yang et al. trained classifiers on data acquired from multitude scanned protocols commonly used by IoT and ICS devices, which were augmented with fingerprints extracted from the network layer communications [4]. This research introduced a significant improvement in labelling training set by the automated scraping of manufacturer and model names of devices from the Internet and matching them against protocol responses in the data set. This developed model has been applied by Jia et al. to determine ownership of devices [5], therefore demonstrating the value of a universal device classifier in helping to solve various research problems.

C. Classes of Devices

Multiple different classifications have been proposed for the devices on the Internet, varying significantly in terms of set size [3], [4], [6]. We propose a small set of 10 classes where every class is selected based on the role, impact, and size of the reachable device set as well as its historical prevalence.

Setting device class definitions is a balancing act, as these can be viewed from the user, functionality, impact and observer perspectives. Creating more classes requires a larger and more precise labelled training set without guaranteed improvement of the total overview. We have identified indistinguishably similar behavior even within small class sets because of the generic HTTP protocol requiring a special class for these devices. At the same time, some of the proposed classes have small subsets of devices, which vary drastically in their behavior and specific purpose. Although the labelled set is significant and proportional to the whole data set, it is not sufficiently representative of various rarer devices and subclasses to train the classifier. When combined with hard-to-distinguish protocol responses, this can introduce even more uncertainty. These issues can be mitigated by augmenting data sets with features from other protocols.

The ICS class contains the most impactful devices which can affect not only individual users but potentially whole regions. It includes industrial control systems, SCADA, and building automation devices. The role and software vary drastically for devices in this class. Although through significant scanning and notification efforts the number of reachable devices has fallen, we are keeping this class.

Network devices are classified as the NET class, which includes all the wired and wireless devices used in individual residential installations and most of the devices serving a more significant role on the network, providing connectivity to organizations and other networks. These are primarily routers, switches, and firewalls. The impact of attacks on these devices cannot be overstated, as not only detectable network interruptions but also hidden MITM attacks can be executed. Other devices in this class include network storage, televisions, and streaming set-top boxes. The INFRA class encompasses data center infrastructure devices affecting the physical properties of the server hardware. These are high-impact devices providing server control panels and virtualization solution control panels.

Although a variety of IoT devices are significant from the serving role viewpoint, we classify all of these in one IOT class. The ratio of IoT devices connected to the Internet versus directly reachable devices is lower than for most other classes. This can be explained by the different ways in which different devices are connected to networks.

The historically prevalent device classes PRINTER, IPCAM, and VOIP are kept separate. These classes had historic public mass attacks that negatively affected a large number of people, e.g. wasting toner printing unwanted documents, leaking private video feeds. Thus their reachability should have decreased over time. The IPCAM class includes not only IP cameras but also DVR and NVR devices that provide recording and viewing functionality. The PRINTER class includes printers and network print servers. The VOIP class includes phone sets, conferencing solutions and VoIP gateways.

It is possible to determine with a high degree of likelihood whether or not a specific device is a generic web server. Features like unsupported HTTP protocol version 1.1, the wrong clock which starts to count time from Unix 0 seconds, and the lack of any headers indicate custom or outdated server software, which usually suggests an embedded device and only in rare cases serves a generic web server role. If we are unable to classify these devices into any other category because response features are insufficient, we classify them as UNCLEAR. This class also includes manufacturers that are represented in multiple classes but where no clear dominant class is established and it is not possible to distinguish device classes from responses, e.g., the same web interface is re-used across classes. In the remaining cases where

we are unable to confirm that the device is not a generic web server, we classify them as UNCATEGORIZED.

From a security research perspective, generic web servers hosting various web applications are often the least exciting class of reachable devices. These devices are much more often properly managed and automatically updated, as they are usually reachable on purpose. The most vulnerable parts of these devices are web applications themselves, not the HTTP servers, but these applications in most cases are reachable using the domain instead of the IP address, which involves a different kind of scanning. There are web applications that are configured to process requests received without the domain name, but quantity-wise they are a minority. We classify all generic web servers, web applications, and services related to these, e.g., CDN, as WEB class.

3. NEURAL NETWORK

The scanning output is HTTP responses that are text in a JSON format. The text classification task in the cybersecurity realm is implemented by a number of text classification methods. Often, classification methods suffer from large vector sizes and are less effective as the number of samples rises. The autoencoder makes use of neural networks which are already in use by latent semantic analysis for text categorization [7] to reduce dimensionality and to improve performance. Another application [8] employs an artificial neural network to improve text classifier scalability. The advantage of the autoencoder method is that it learns automatically from examples.

The main advantage of existing text classification methods, such as Support Vector Machine (SVM) [9], Word Embeddings Neural Networks or the Gensim tool, is that they perform better with a massive database for training to provide meaningful results, and we have a big dataset. However, the common disadvantage of these techniques is the lack of results transparency due to employing vectors containing real-valued numbers. These tools provide results, but it is difficult to explain how the results are calculated. Another disadvantage is the inability to handle unknown words or words which were not included previously in the training vocabulary. The SVM approach is limited by choice of the kernel, which is a general weak point of SVM applications.

Alternative algorithms employing categorical features and labels are Naive Bayes [10], Logistic Regression [11], and Random Forests [12]. Approaches based on decision trees such as Random Forests are very fast to train but quite slow to create predictions once trained. A higher degree of accuracy requires additional trees, which means losing performance. Naive Bayes often serves as a robust method for data classification, but the vectors representing an incident in Naive Bayes are larger than

in word-embedding methods, and also Naive Bayes classifiers make a very strong assumption on the shape of the data distribution. Further problems may result due to data scarcity, which can result in probabilities going towards 0 or 1, leading to numerical instabilities and worse detection results. Logistic regression, like a Naive Bayes method, requires each feature in an incident to be independent of all other features. Logistic regression models are also vulnerable to overconfidence as a result of sampling bias. Consequently, for the particular use case of classifying IoT devices, we suggest using the simplest neural network for text classification that scales well because of the small vector size while maintaining a high level of accuracy.

A. Features Used for Classification

Features of the HTTP responses suitable for the classification have previously been explored by Lavrenovs et al. [13], [14]. For this research, we have decided to use all HTTP response headers and their values, Autonomous system (AS) name, HTML structure hash, body title, body keywords, SSL certificate issuer, and subject.

Specific features are extracted from the response body. HTML tree, in many cases, uniquely identifies groups of the same devices as long as the tree is large enough. To decrease data pollution, we are using only the hash of the HTML tree. The first title is extracted from the HTML body. These titles can often identify specific device models, manufacturers and functionality. The body of the response contains a significant number of mark-up language elements, which do not necessarily benefit us as separate features if the body tree hash is being utilized. We keep only the 1,000 most common words.

Although HTTPS protocol is not being targeted specifically, a small subset of the devices with redirects to HTTPS have numerous TLS properties. However, most of them are usually not uniquely identifying device classes on their own. Even supported ciphers and their order can be used as features, and all of these properties are worthy of investigation in the future for the HTTPS device scan on the Internet. For this research, we use only SSL certificate issuer and subject as those were used for manually labelling the sample and often identified the class of the device on their own.

B. Data Sets

We are operating with four data sets created by scanning the Internet using scanning tools commonly used for research: zmap and zgrab. Both HTTP default port 80 and common alternative port 8080 were scanned in December 2018 and one year apart in December 2019. Up to three redirects are being followed to any port including HTTPS, in which case TLS negotiation is being saved as well. For the standard port in 2018, there are 54,811,827 elements, and in 2019 there are 57,131,825 elements. For the alternative port, there are 7,792,077 and 8,100,201 elements, respectively. An element

is a single response or response redirect chain corresponding to a single request that contains at least one proper HTTP response. Specifically targeting HTTPS ports and also analyzing broken responses would identify additional web control panels, but we have excluded that from the scope of the current research.

We have augmented elements in data sets with additional features. AS name is looked up via the Maxmind GeoIP database. HTML tree hash, first title and body words are all generated from the response HTML body itself.

The labelled set consists of 171,791 elements. It was created from random elements of the 2018 port 80 data set and therefore is unbalanced across classes. There are 132,562 WEB, 22,002 NET, 9561 IPCAM, 711 INFRA, 265 VOIP, 243 ICS, 218 IOT, 153 PRINTER, 4175 UNCLEAR and 1901 UNCATEGORIZED devices in the labelled set.

C. Comparison to the Existing Classification

The overall idea of our solution and [4] is the same: to classify devices on the Internet using artificial intelligence from the remote point of view. The classification model suggested in [4] provides classification on three levels: the type of IoT device, vendor and product. In contrast, the proposed solution aims to classify only by type of IoT device because the vendor and product is just additional information to the class. The approach of crawling additional device information from the Internet, using HTTP queries and analyzing different protocol levels, looks promising but is very unreliable, taking into account the sparse information for such queries. This could be done for the proposed solution as future work, e.g. via query language such as Sparql to compare if this method yields additional value.

Our approach mainly uses information from HTTP headers and body. Yang et al. [4] perform substantial manual pre-training steps. In our approach, we leverage the knowledge and rules developed prior to this research and described in [13], [14]. The existing solution has a very complicated neural network while we propose an alternative solution with possibly more dedicated methods.

Yang et al. classified 15.3 million IoT and ICS devices [4], whereas we analyzed up to 57 million all type devices. Their protocol coverage is higher - 20 protocols (4 ICS). We analyzed HTTP exclusively, but plan to cover additional protocols in the future. Using network-level fingerprinting is extremely unreliable on its own and may produce bias in the overall results. Compared to 41 device types (classes) in the existing research, we make use of 10 classes evaluated from aggregated expert knowledge. The more classes we have, the more unreliable the classification is. The identification of classes itself is a challenging task even for manual analysis and

definition for humans. Therefore, a high number of classes could reduce overall accuracy since there is no common understanding of class definitions.

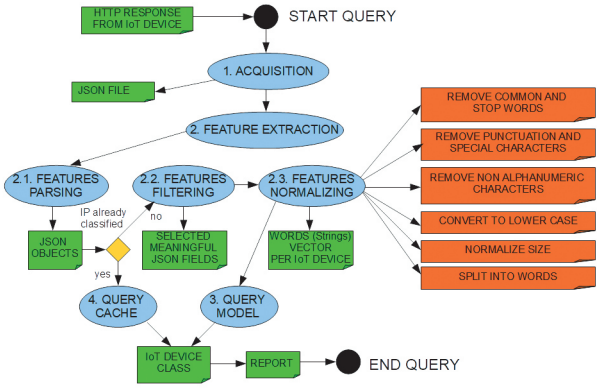
D. General Workflow

Device classification employs features extraction and training of the neural network to produce a model for the queries. Classification predicts previously defined categories for a given sample. There are ten expert-defined classes: ICS, INFRA, IOT, IPCAM, NET, PRINTER, UNCATEGORIZED, UNCLEAR, VOIP, WEB. Supervised learning employs labelled training data to learn mapping functions from a given input (list of words) to the desired output value (class name). A supervised learning algorithm analyzes the data through weights and activation functions that activate neurons and produce an inferred function, which is then used for mapping new samples or correctly determining classification labels for unseen instances.

The workflow process is composed of two parts. One process is neural network model training, where the workflow acquires device data from different sources such as the Internet and domain experts. The model is trained and regularly updated by extended knowledge from new device crawls.

Figure 1 provides an overview of the device classification using neural networks. This approach is based on a knowledge base containing a large number of labelled responses in JSON format (step 1). This data can be provided by different means, collected at different times for particular operating systems, and can be separated by type of application and protocol. The novelty of this approach is that, for typical use cases, we propose to have associated decision rules for initial labelling. All such rules are then aggregated in a common labelled dataset, which supports final classification. We send requests to devices, and the system extracts features (step 2) from the response and stores them for further analysis and queries the model that was trained on the knowledge base. During the feature extraction, we apply parsing, filtering, and normalizing of the content. The final classification result is based on querying the model (step 3) or cache (step 4), if sample hash is already known, and is a report in the form of a particular class name.

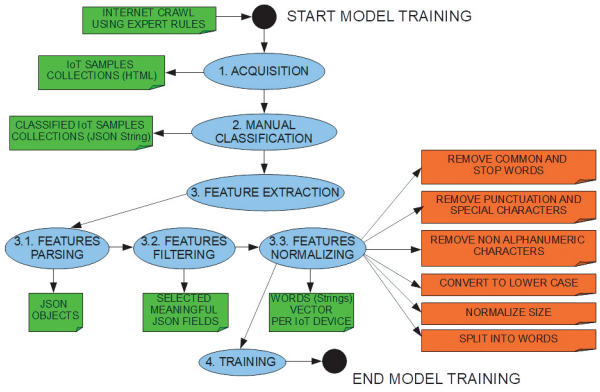
FIGURE 1. THE WORKFLOW FOR FEATURE EXTRACTION AND CLASSIFICATION OF DEVICES USING A NEURAL NETWORK.



E. Model Training

The data for model training is prepared as described in Figure 1 in the previous section. After acquisition and feature extraction, the input for the model is a list of words for each sample. This is then converted into the one-hot vector to be processed in the input level of the neural network model (step 4) in Figure 2. To perform training, features aggregated in text form must be converted into numerical values, since machine learning algorithms and deep learning architectures cannot process plain text. Therefore, each uploaded sample (see Figure 2) is converted into an array of strings, where each string represents a particular feature. Then strings are encoded by indices, and each feature string has a unique index. If this feature repeats in the samples, we re-use its index. Finally, arrays of indexes are converted in one-hot encoded vectors, meaning that the position of each feature in the original feature set is encoded using "1" if a feature exists in the given place or "0" if not.

FIGURE 2. THE WORKFLOW FOR MODEL TRAINING FOR DEVICES USING A NEURAL NETWORK APPROACH.



The neural network used a total of 517,642 parameters during the training. A summary of the neural network training is presented in Table 1. The neural network is composed of an input layer and an output layer. The number of neurons in these layers ranges from 10 to 512. The input layer uses a rectified linear unit (ReLU) as an activation function. The output layer employs a softmax activation function, which provides probabilities as to which of 10 classes a particular sample belongs to.

TABLE 1: SUMMARY OF THE NEURAL NETWORK TRAINING PROCESS.

Layer	Type	Activation Function	Neurons #	Parameters #
Input layer	Dense	ReLU	512	512512
Output layer	Dense	Softmax	10	5,130

We performed a total of 10 training iterations (epochs). The neural network training and accuracy calculation process took 15.723163 seconds (Figure 3). This figure shows that loss and validation loss decreased and accuracy and validation accuracy increased with each epoch.

FIGURE 3. ACCURACY AND LOSS CHARACTERISTICS BY NEURAL NETWORK TRAINING.

We trained two models - one with the full labelled data set (large) and one balanced model (small). Comparing their accuracy (about 87% for small and 97% for the large data set), we noticed by randomly sampling the classified output of the whole data set that the small model performed better due to the bias in the large data set. As the full labelled data set primarily consists of WEB devices, the classified output is significantly skewed towards classifying devices as WEB. To avoid bias

of overrepresented classes in the labelled data set (in total 171,791), such as WEB, we employ a balanced labelled training set (in total 11,479): ICS:243, INFRA:711, IOT:218, IPCAM:1,999, NET:2,000, PRINTER:153, UNCATEGORIZED:1,901, UNCLEAR:1,999, VOIP:265, WEB:1,999. The labelled training data set was divided into a training set (5,628), validation set (2,413), and test set (3,447). The test accuracy is 0.87277.

4. RESULTS

The model was trained using the 2018 standard port labelled data set and applied to the 2019 standard port data set as well as the port 8080 data sets for both years. Although the reachability of devices has been recognized as a poor and high-risk management practice, there was an increase in the data set sizes in 2019.

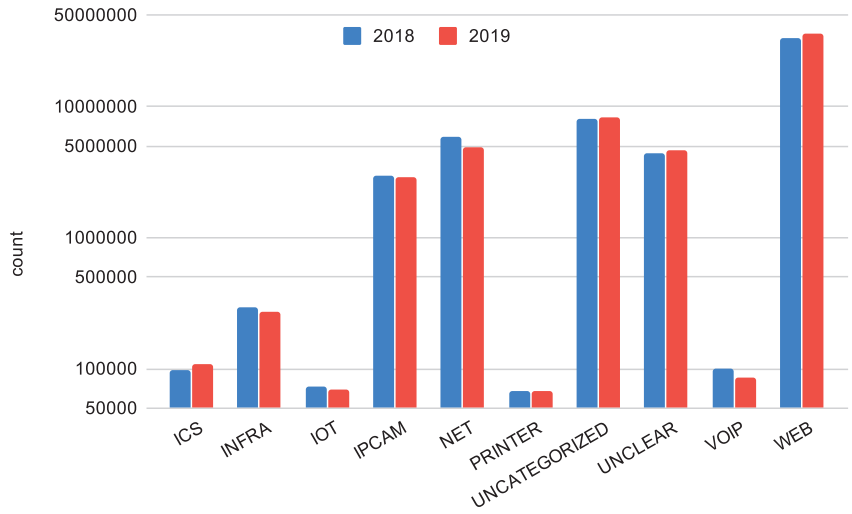
The standard port 80 classification results are provided in Figure 4. As expected from the labelled set, WEB devices were the most prevalent ones. It was not expected that the UNCLEAR and UNCATEGORIZED devices would be so numerous, but that can be explained. UNCLEAR and UNCATEGORIZED devices often have a small set of rare features extracted from the HTTP responses, which makes even manually classifying them challenging and in many cases impossible. While creating the labelled set, many of these devices were categorized. This was done through numerous weak rules utilizing only the available features. These features might be sufficiently rare and unique to not be applicable to the whole data set, in which case HTTP response data on its own might not suffice for accurate classification.

We can observe a slight decrease in reachable INFRA and IOT devices in 2019. As the number of IOT devices is growing significantly, it would be expected that the number of reachable devices would grow over the one-year period. However, this class of devices is the only one of the defined classes that historically could rarely be connected in a way that made them reachable. A more significant decrease in VOIP could be explained by changes in the way this type of device is deployed and managed at the vendor level.

From the publicly well-known attacks targeting IPCAM and PRINTER devices, it could be expected that the number of reachable ones would decrease significantly, but no such trend is observable. One explanation is that the number of newly added reachable devices closely matches the ones that were mitigated. It is currently not clear what portion of these almost 3 million IPCAM devices have to be reachable for remote surveillance and recording purposes.

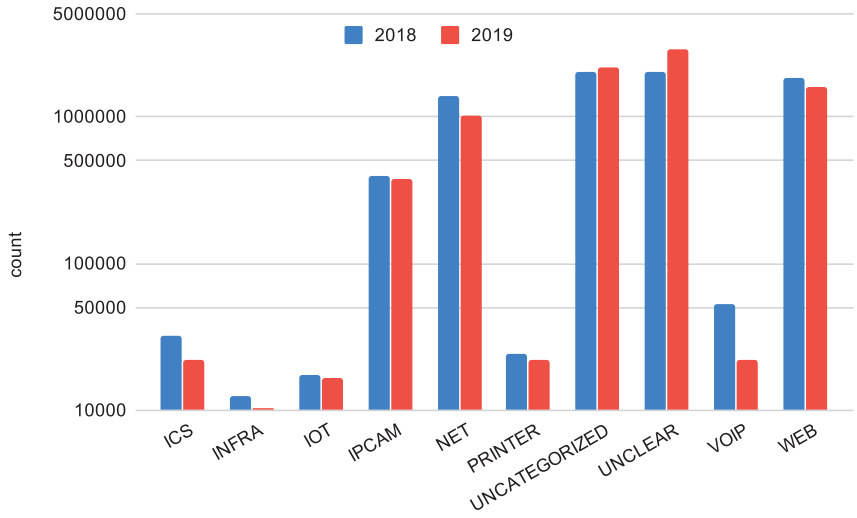
A large number of NET devices was expected. A residential Internet connection device can expose the control panel to the Internet even if the initial setup is done by the ISP technician. A significant drop in the number of these devices might suggest that the device life cycle could be playing a role, with older ones getting replaced and newer ones having a better configuration.

FIGURE 4. DISTRIBUTION OF DEVICE CLASSES FOR PORT 80 FOR 2018 AND 2019.



The alternative port 8080 classification results are presented in Figure 5. As expected, the WEB devices are a proportionally smaller class than on the port 80 where generic websites usually reside. UNCLEAR and UNCATEGORIZED are the two largest classes and show significant growth over the one-year period, which might suggest that the feature difference is significant enough between the two ports that the model needs to be augmented with the alternative port data as well. We can observe much more significant proportion changes among the classes on the alternative port.

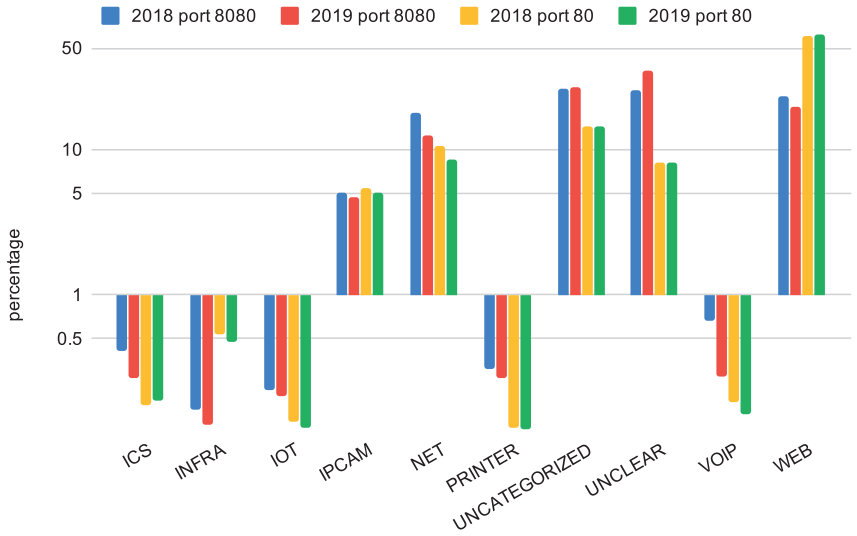
FIGURE 5. DISTRIBUTION OF DEVICE CLASSES FOR PORT 8080 IN 2018 AND 2019.



The relative class distribution for all four classified data sets is presented in Figure 6. This view enables us to make a comparison between the utilization of different devices on different ports. There are other discernible differences besides the already identified WEB, UNCATEGORIZED and UNCLEAR classes. INFRA devices are proportionally about four times less prevalent on the alternative port; this could be explained by the fact that there are a small number of manufacturers whose devices were identified and labelled on the port 80. These devices might use the default port setting, and there might be unidentified INFRA devices defaulting to 8080 port.

Interestingly, IPCAM has almost the same proportion across the ports with the same decrease over the one year. Proportionally, there are significantly more PRINTER devices on the alternative port, and that is explainable with the high variance of device models and default configurations even among individual manufacturers. VOIP, ICS, IOT and NET devices are also proportionally more represented on the alternative port. This might be the result of manufacturers' concerns about creating port conflicts on a single device. This concern is especially valid for NET devices, which are handling networking traffic and possibly forwarding the port 80 to another device.

FIGURE 6. PROPORTIONAL DISTRIBUTION OF DEVICES FOR PORT 80 AND 8080 IN 2018 AND 2019.



5. CONCLUSIONS

We have successfully trained a machine learning classifier for web interfaces achieving 87% test accuracy without the use of a rule engine. Although using the full labelled set to train the neural network achieved higher test accuracy of 97%, further research is needed to determine if this higher accuracy can be achieved while avoiding the bias caused by an unbalanced data set. A large proportion of devices being classified as UNCLEAR and UNCATEGORIZED was unexpected but explainable and can be addressed through augmenting data with features from other protocols. Although the model for the standard port functioned for the alternative port, the increase in UNCLEAR and UNCATEGORIZED devices indicates that there might be a sufficient number of devices unique to the alternative port. This therefore requires the data from the alternative port to be included into the labelled training set or a separate model created.

Our future work will include augmenting the model with HTTPS web interfaces and additional common or high impact port checks and appropriate protocol communication responses. Reverse and forward DNS as an additional source of features could more precisely filter out WEB servers that are currently UNCATEGORIZED. Fingerprinting TCP communications as an additional feature is worthy of investigation as well. Redeveloping rules used for labelling the sample set into a rule engine should significantly increase the accuracy of the classification.

This type of classifier could provide the full Internet view of the reachable devices, with details of individual countries and networks. It has significant value not only for research purposes but also to provide overview reports to decision-makers about which security concerns require the most attention. The same classifier can also be used for internal networks, by-passing firewall restrictions and classifying devices with open ports, thus competing with the observer approach.

The application of machine learning to various research problems is currently hard to replicate in most cases. We are planning to develop the classifier with the discussed improvements as an API available to researchers to help others to address a vast range of network-related research questions more precisely.

REFERENCES

[1] Statista Inc., "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025," November 2019. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ [Accessed: 16-12-2019].

[2] Sivanathan, A. *et al*., "Characterizing and classifying IoT traffic in smart cities and campuses," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Atlanta, GA, May 2017, pp. 559–564, doi: 10.1109/INFCOMW.2017.8116438.

[3] Bezawada, B., M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "Behavioral Fingerprinting of IoT Devices," in *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security - ASHES '18*, Toronto, Canada, 2018, pp. 41–50, doi: 10.1145/3266444.3266452.

[4] Yang, K., Q. Li, and L. Sun, "Towards automatic fingerprinting of IoT devices in the cyberspace," *Computer Networks*, vol. 148, pp. 318–327, Jan. 2019, doi: 10.1016/j.comnet.2018.11.013.

[5] Jia, Y., B. Han, Q. Li, H. Li, and L. Sun, "Who owns Internet of Things devices?," *International Journal of Distributed Sensor Networks*, vol. 14, no. 11, p. 155014771881109, Nov. 2018, doi: 10.1177/1550147718811099.

[6] Cvitić, I., D. Peraković, M. Periša, and M. Botica, "Novel approach for detection of IoT generated DDoS traffic," *Wireless Networks*, Jun. 2019, doi: 10.1007/s11276-019-02043-1.

[7] Yu, B., Z. Xu, and C. Li, "Latent semantic analysis for text categorization using neural network," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 900–904, Dec. 2008, doi: 10.1016/j.knosys.2008.03.045.

[8] Lam, S. L. Y. and Dik Lun Lee, "Feature reduction for neural network-based text categorization," in *Proceedings. 6th International Conference on Advanced Systems for Advanced Applications*, Hsinchu, Taiwan, 1999, pp. 195–202, doi: 10.1109/DASFAA.1999.765752.

[9] Auria, L. and R. A. Moro, "Support Vector Machines (SVM) as a Technique for Solvency Analysis," *SSRN Journal*, 2008, doi: 10.2139/ssrn.1424949.

[10] Manning, C. D., P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York: Cambridge University Press, 2008.

[11] Cox, D. R., "The Regression Analysis of Binary Sequences," *Journal of the Royal Statistical Society: Series B*, vol. 20, no. 2, pp. 215–242, 1958.

[12] Tin Kam Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Montreal, Que., Canada, 1995, vol. 1, pp. 278–282, doi: 10.1109/ICDAR.1995.598994.

[13] Lavrenovs, A. and G. Visky, "Exploring features of HTTP responses for the classification of devices on the Internet," presented at the 2019 27th Telecommunications Forum (TELFOR), Belgrade, Serbia, Nov. 2019, doi: https://doi.org/10.1109/TELFOR48224.2019.8971100.

[14] Lavrenovs, A. and G. Visky, "Investigating HTTP response headers for the classification of devices on the Internet," presented at the 2019 IEEE 7th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Liepaja, Latvia, Nov. 2019, doi: 10.1109/AIEEE48629.2019.8977115.