

April 2019

## Security Framework for the Internet of Things Leveraging Network Telescopes and Machine Learning

Farooq Israr Ahmed Shaikh  
*University of South Florida, farooqshaikh36@gmail.com*

Follow this and additional works at: <https://scholarcommons.usf.edu/etd>

 Part of the [Electrical and Computer Engineering Commons](#)

---

### Scholar Commons Citation

Shaikh, Farooq Israr Ahmed, "Security Framework for the Internet of Things Leveraging Network Telescopes and Machine Learning" (2019). *Graduate Theses and Dissertations*.  
<https://scholarcommons.usf.edu/etd/7935>

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact [scholarcommons@usf.edu](mailto:scholarcommons@usf.edu).

Security Framework for the Internet of Things Leveraging Network Telescopes and  
Machine Learning

by

Farooq Israr Ahmed Shaikh

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
Department of Electrical Engineering  
College of Engineering  
University of South Florida

Co-Major Professor: Nasir Ghani, Ph.D.  
Co-Major Professor: Elias Bou-Harb, Ph.D.  
Ismail Uysal, Ph.D.  
Zhuo Lu, Ph.D.  
Srinivas Katkoori, Ph.D.

Date of Approval:  
March 13, 2019

Keywords: Darknet, Ensemble Learners, Deep Learning, Network Statistics

Copyright © 2019, Farooq Israr Ahmed Shaikh

## Dedication

To my mother, Reshma, for being my mentor, fan, critic, friend and my inspiration.

To my beloved father, Israr Ahmed, for believing in me.

## Acknowledgments

I would like to acknowledge and dedicate my gratitude to the following who made the completion of this work possible:

God, for giving me strength, capability and perseverance.

My mother, Reshma, and my sister, Fatima, for their unconditional love and support.

My supervisor, Dr. Nasir Ghani, for believing in me and his continuous help and advisement throughout my Ph.D. study.

My co-supervisor, Dr. Elias Bou-Harb, for his contribution and continuous assessment throughout my research.

My dissertation committee members, Dr. Ismail Uysal and Dr. Zhuo Lu, for their valuable suggestions and advice on this dissertation work.

My former and current research colleagues, Dr. Hao Bai, Dr. Mahsa Pourvali, M.E. Andrea Wright, Dr. Mohammed Jasim, M.E. Nazli Siasi and Mr. Aldin Vehabovic, for their collaboration.

My friends at the Whitehatters Computer Security Club, especially M.E. Kristopher Willis and Mr. Brad Daniels, for their support and encouragement.

## Table of Contents

List of Tables . . . . .	iii
List of Figures . . . . .	iv
Abstract . . . . .	vi
1 Introduction . . . . .	1
1.1 Background Overview . . . . .	1
1.2 Motivations . . . . .	3
1.3 Problem Statement . . . . .	5
1.4 Proposed Work and Contributions . . . . .	6
2 Background and Related Work . . . . .	8
2.1 Survey on the Internet of Things . . . . .	8
2.2 Survey on IoT Security Threats . . . . .	11
2.3 Survey on Machine/ Deep Learning Applications in IoT . . . . .	14
2.4 Applications in Intrusion Detection System . . . . .	17
2.5 Survey on SDN/NFV Solutions for IoT Security . . . . .	21
2.6 Open Challenges . . . . .	24
3 Correlating Active and Passive Measurements to Infer and Characterize Unsolicited IoT Devices . . . . .	25
3.1 Notation Overview . . . . .	26
3.2 Inferring and Characterizing Internet-Scale Compromised IoT Devices . . . . .	27
3.2.1 Exploiting Network Telescopes . . . . .	28
3.3 Leveraging Internet-Wide Scanning . . . . .	30
3.4 Correlating Active and Passive Measurements . . . . .	31
3.5 Generating IoT-Specific Malicious Signatures . . . . .	32
3.6 Sharing of IoT Unsolicited Empirical Threat Information . . . . .	34
3.7 Empirical Evaluation . . . . .	36
4 Malicious IoT Device Classification using Machine Learning . . . . .	42
4.1 Notation Overview . . . . .	43
4.2 Exploiting Network Telescopes to Infer Malicious IoT Behavior . . . . .	44
4.3 Darknet Data Categorization . . . . .	45
4.3.1 Scanning . . . . .	46
4.3.2 Backscatter . . . . .	47

4.4	Feature Extraction from Network Telescopes . . . . .	48
4.5	Data Processing . . . . .	49
4.6	Machine Learning Models . . . . .	51
4.6.1	Random Forest . . . . .	54
4.6.2	Gradient Boosting . . . . .	55
4.6.3	AdaBoost . . . . .	56
4.7	Performance Evaluation . . . . .	57
5	IoT Threat Detection Leveraging Network Statistics and GAN . . . . .	63
5.1	Experimental Setup . . . . .	64
5.2	Attack Methodology . . . . .	67
5.3	Overview of GAN . . . . .	69
5.3.1	ALI GAN . . . . .	72
5.3.2	AnoGAN . . . . .	73
5.4	Data Processing and Feature Selection . . . . .	75
5.5	Performance Evaluation . . . . .	77
6	Concluding Remarks . . . . .	82
6.1	Summary of Research Findings . . . . .	83
6.2	Future Work . . . . .	86
	References . . . . .	87
	Appendix A Proof of Copyright Permissions . . . . .	96
	Appendix B Glossary . . . . .	98

## List of Tables

Table 1	List of variables . . . . .	27
Table 2	List of variables . . . . .	44
Table 3	Features for training ML algorithms . . . . .	50
Table 4	Recall and precision values for the January 2017 dataset . . . . .	60
Table 5	Recall and precision values for the February 2017 dataset . . . . .	60
Table 6	Recall and precision values for the March 2017 dataset . . . . .	60
Table 7	Exploits against Netgear DGN 2200 . . . . .	68
Table 8	Attacks against IoT network . . . . .	69
Table 9	Description of GAN models . . . . .	75
Table 10	Description of features selected for GAN learning . . . . .	76

## List of Figures

Figure 3.1	Network telescopes capturing Internet-scale IoT unsolicited traffic . . . . .	29
Figure 3.2	Correlation algorithm for unsolicited IoT device identification . . . . .	32
Figure 3.3	Inferring and mitigating Internet-scale unsolicited IoT devices: A network telescope approach . . . . .	35
Figure 3.4	IoT devices: a) signatures and b) distributed by types . . . . .	39
Figure 3.5	Exploited IoT devices: a) by types and b) by type and region . . . . .	40
Figure 3.6	Global distribution of exploited IoT devices . . . . .	41
Figure 4.1	Proposed IoT device classification methodology . . . . .	51
Figure 4.2	Data labelling and feature selection algorithm . . . . .	51
Figure 4.3	Boosting vs Bagging . . . . .	53
Figure 4.4	Random Forest algorithm . . . . .	54
Figure 4.5	Gradient Boosting algorithm . . . . .	55
Figure 4.6	SAMME AdaBoost algorithm . . . . .	56
Figure 4.7	Recall and precision scores for: a) January 2017 and b) February 2017 . . . . .	61
Figure 4.8	a) Recall and precision scores for March 2017 and b) categorization based on activity . . . . .	62
Figure 5.1	Holistic overview perspective of the experimental testbed attack setup . . . . .	64
Figure 5.2	Mirai attack strategy . . . . .	68
Figure 5.3	Overview of generative adversarial network (GAN) . . . . .	71



Figure 5.4	Overview of ALI GAN framework . . . . .	73
Figure 5.5	ALI GAN model for IoT threat detection . . . . .	74
Figure 5.6	GAN performance on a) 3 IoT devices b) 31 IoT devices . . . . .	80
Figure 5.7	a) GAN performance on darknet and b) inference time . . . . .	81

## Abstract

The recent advancements in computing and sensor technologies, coupled with improvements in embedded system design methodologies, have resulted in the novel paradigm called the Internet of Things (IoT). IoT is essentially a network of small embedded devices enabled with sensing capabilities that can interact with multiple entities to relay information about their environments. This sensing information can also be stored in the cloud for further analysis, thereby reducing storage requirements on the devices themselves. The above factors, coupled with the ever increasing needs of modern society to stay connected at all times, has resulted in IoT technology penetrating all facets of modern life. In fact IoT systems are already seeing widespread applications across multiple industries such as transport, utility, manufacturing, healthcare, home automation, etc.

Although the above developments promise tremendous benefits in terms of productivity and efficiency, they also bring forth a plethora of security challenges. Namely, the current design philosophy of IoT devices, which focuses more on rapid prototyping and usability, results in security often being an afterthought. Furthermore, one needs to remember that unlike traditional computing systems, these devices operate under the assumption of tight resource constraints. As such this makes IoT devices a lucrative target for exploitation by adversaries. This inherent flaw of IoT setups has manifested itself in the form of various distributed denial of service (DDoS) attacks that have achieved massive throughputs without the need for techniques such as amplification, etc. Furthermore, once exploited, an IoT device can also function as a pivot point for adversaries to move laterally across the network and exploit other, potentially more valuable, systems and services. Finally, vulnerable IoT

devices operating in industrial control systems and other critical infrastructure setups can cause sizable loss of property and in some cases even lives, a very sobering fact.

In light of the above, this dissertation research presents several novel strategies for identifying known and zero-day attacks against IoT devices, as well as identifying infected IoT devices present inside a network along with some mitigation strategies. To this end, network telescopes are leveraged to generate Internet-scale notions of maliciousness in conjunction with signatures that can be used to identify such devices in a network. This strategy is further extended by developing a taxonomy-based methodology which is capable of categorizing unsolicited IoT behavior by leveraging machine learning (ML) techniques, such as ensemble learners, to identify similar threats in near-real time. Furthermore, to overcome the challenge of insufficient (malicious) training data within the IoT realm, a generative adversarial network (GAN) based framework is also developed to identify known and unseen attacks on IoT devices. Finally, a software defined networking (SDN) based solution is proposed to mitigate threats from unsolicited IoT devices.

## 1 Introduction

This dissertation presents a study of security frameworks and methodologies for the Internet of Things (IoT). The focus is on leveraging recent advances in the area of machine learning (ML) and deep learning (DL), along with novel networking paradigms, to boost IoT security. Hence this introductory chapter presents an overview of security threats to IoT devices and networks, as well as the key motivations for this work. Subsequently, an overview of the core contributions of this research are presented along with a broad outline of the remaining chapters.

### 1.1 Background Overview

The last few years have seen an exponential rise in the popularity of tiny embedded devices collectively known as IoT. These devices have steadily forayed into all aspects of modern society and are changing the way people, businesses and industries operate. One of the primary motivations for development in this domain has been the need to have *smart* devices and applications that can operate autonomously without the need for human intervention (for control and data analysis). IoT growth has been further fueled by advances and cost efficiencies in embedded systems design, optimized application development and improved communication protocols.

The manifold benefits of IoT technology are already being seen across multiple diverse sectors. For example, pervasive medical devices are being used to continuously monitor patient health and independently and immediately communicate any abnormalities to health monitoring centers. The benefits of IoT devices are also visible in the manufacturing sector, leading to overall improvement in production capacity as well as logistics. In addition to

this, the visions for smart homes, intelligent transportation and smart cities are now being realized with the help of IoT devices as well, leading to improved standards of living and convenience in daily life. For example, massive amounts of sensor data gathered by IoT devices can be used by the transport industry to improve efficiency and reduce operational costs. Legacy power distribution infrastructures are also being enhanced with the use of IoT devices, leading to efficient resource utilization and energy distribution. IoT devices are also seeing applications in waste management and environmental monitoring. New communication technologies, such as radio frequency identification (RFID), are also enabling novel concepts such as smart parking and optimized traffic routing. New paradigms like vehicle-to-vehicle communication are also being envisioned, leveraging various sensors connected in an intelligent network to improve road travel safety. Furthermore, IoT devices will also be key enablers in future self-driving cars.

Clearly, the large scale deployment of IoT devices promises to transform many aspects of contemporary life. However, as is the case with any disruptive technology, IoT too brings with it a Pandora's box of challenges [1]. One of the most prominent flaws of IoT has been the lack of built-in security mechanisms akin to those seen in traditional systems, e.g., like servers, personal computers and laptops. This shortcoming arises due to the fact that most embedded devices do not have the computational resources required to implement advanced security policies and encryption. Furthermore, due to increased market competition, manufacturers of IoT devices typically have paid more attention to the "three Ps", i.e., prototype, production and packaging, without giving much importance to security. Software updates and patches for IoT devices are also not automated and in many cases not even supported. In the few instances that one can find certain security mechanisms, they are mostly ad hoc and often do not extend beyond simple software implementations. Note that along with these inherent security deficiencies, the sheer number and type of IoT devices made by different

manufacturers inevitably facilitates exploitation by adversaries. This fact makes it very difficult to come up with a singular security framework that adapts well to all scenarios.

Also, one of the key benefits of IoT is the massive amounts of data that its various sensors generate, which in turn can be processed offline using big data analytics. However, these steps often involve the use of third-party cloud hosting platforms, raising further issues of user privacy and data integrity (both while the data is in transit and when it is stored on remote servers). Thus, it becomes imperative to ensure that proper security mechanisms are implemented to help maintain the integrity of IoT data, as well as protect private user information from falling into the hands of malicious actors. As a result, the traditional security triad of confidentiality, integrity and authentication (CIA) is no longer sufficient and needs to include other security attributes, e.g., such as access control and availability, given the dynamic nature of IoT networks. Finally, since a significant portion of IoT device communication occurs over the wireless medium, this has further potential to lead to attacks such as relay, man-in-the-middle (MITM), etc.

Overall, the above-noted vulnerabilities of IoT devices have manifested themselves in the form of various malware and DDoS attacks. The most prominent of these was the Mirai botnet [2] and the attack on Dyn DNS servers using an army of infected IoT cameras to cause large scale disruption of Internet services. Similarly, other such attacks have also been carried out using infected IoT devices to cause Internet-scale disruptions, see [3] [4].

## 1.2 Motivations

One critical challenge in developing security frameworks for IoT devices is the lack of empirical data that can aid in the analysis of attacker behavior. This deficiency can be attributed to multiple reasons, primary among them being logistics and privacy concerns. The situation is further exacerbated by the variety of existing IoT devices in terms of their types, versions, operating system, etc. Hence, inferring and characterizing unsolicited IoT

behavior will directly benefit researchers in understanding critical weaknesses in related protocols and system design. Ultimately these steps can lead to faster remediation strategies.

Nevertheless, attackers are also adapting to post-Mirai defense mechanisms installed by various organizations. In essence, Mirai was basically a primitive malware that brute-forced factory default username and passwords to gain access to insecure IoT devices. Hence this modus operandi could be identified by looking for traffic directed towards or transmitted by the telnet port, i.e., since telnet was the service used by Mirai to exploit IoT devices. However, newer IoT malware variants are now evolving to remote code and other exploits directed towards a specific device. These attacks are much harder to detect and can go unnoticed by the aforementioned defense mechanism. This problem was clearly displayed in the case of Reaper, Owari [5] and other Mirai variants which employed a much more complex and discreet attack strategy.

Now the current body of research work on IoT security has primarily focused on context-aware or behavioral models. The objective of these studies is to create and enforce collaborative models keeping in mind the heterogeneous nature and complex interactions of IoT devices. Other endeavors have analyzed protocol weaknesses in specific IoT devices, as well identifying access control and monitoring mechanisms in some of them. Meanwhile researchers working in the area of wireless sensor networks (WSN) have focused their efforts on MITM, denial of service (DoS), replay and similar attacks on the networking layer. Efforts have also been made to leverage ML techniques and develop anomaly detection and signature-based algorithms to identify unsolicited IoT behaviors.

However, as noted later, many of these solutions suffer from the lack of sufficient data points to help build effective models to cover the entire spectrum of possible attacks. Yet in other cases, proposed defense mechanisms are only effective for a certain type of device, reducing their effectiveness in real world scenarios where many different types of devices and communication protocols are used, i.e., a higher level of heterogeneity. It is also impor-

tant to remember that attacker strategies are constantly evolving to defeat current security mechanisms (as was pointed out earlier). This fact implies that simple signature-based algorithms will prove inefficient in securing IoT setups. Thus the need of present and future IoT systems is a dynamic and flexible security framework that can identify both simple and complex threats to the network. In addition to this, such a framework should be able to actively identify malicious IoT devices present within a network.

### 1.3 Problem Statement

This dissertation addresses the above challenges and develops a novel set of methodologies to secure IoT networks deployed in residential and enterprise environments. To this end, network telescopes, i.e., darknet data, is leveraged to infer and characterize Internet scale notions of maliciousness. A taxonomy of darknet data is then generated to categorize unsolicited IoT data. This labelled data is then used in a security framework that employs ML-based ensemble learners to identify malicious IoT traffic directed towards a network, infected IoT devices within a network, as well as DoS activities conducted by IoT devices.

Finally, a generative adversarial network (GAN) based framework is also proposed to identify zero-day attacks along with known exploits against IoT devices. The unique advantage of this model is that it does not rely on any malicious data for training purposes and thus overcomes one of the key hindrances of securing IoT networks. This methodology (when used in conjunction with the previous ensemble learning methodology) can potentially identify a wide spectrum of threats. Innovative concepts such as software defined networking (SDN) and network function virtualization (NFV) are also incorporated to build dynamic security strategies that can be used with the aforementioned models to quickly mitigate a threat once it has been identified.



## 1.4 Proposed Work and Contributions

This dissertation studies the problem of securing IoT devices found in residential and business enterprise networks. The key contributions of this effort include the following:

- A unique approach that leverages network telescopes (darknet data) to infer and characterize malicious IoT devices seen on the Internet by correlating active and passive empirical measurements (along with generating IoT-specific signatures using a rolling hash algorithm).
- Characterization and labelling of different unsolicited IoT behaviors as well as training and testing of ensemble learners to classify IoT devices based on network traffic characteristics.
- Design of a novel framework using a GAN-based neural network to identify known and zero-day attacks on IoT devices based on network flow characteristics.

The remainder of this dissertation is organized as follows. First, Chapter 2 presents a detailed survey of the IoT ecosystem along with some of the security models proposed to defend IoT networks. Next, Chapter 3 details an empirical approach that leverages data from the CAIDA darknet repository for characterizing unsolicited IoT devices seen on the Internet. A rolling hash algorithm is also proposed for generating signatures that can be used alongside intrusion detection systems (IDS) to help organizations identify attacks against their IoT infrastructures.

Chapter 4 then extends this work by developing a taxonomy-based methodology for categorizing malicious IoT activities. This framework is then used to train a set of ensemble learners to capture the unique characteristics of each type of malicious activity. This setup can also be used as a stand-alone system for detecting malicious IoT activities. Meanwhile Chapter 5 presents a GAN-based neural network for identifying various seen and unseen

attacks on actual IoT devices. An advanced implementation framework for quickly mitigating these threats is also proposed. Finally, conclusions and directions for future work are thereafter presented in Chapter 6 to conclude this dissertation.

## 2 Background and Related Work

The overall area of IoT security has received a lot of attention in the last few years, especially in the light of recent large scale DDoS attacks and crypto-mining bots. In response, researchers have proposed various methodologies and frameworks to strengthen defenses for IoT devices. Along these lines, this chapter overviews some of the latest developments in the IoT domain and details key security mechanisms, with a particular focus on empirical analysis and ML-based methods. Several SDN and NFV-based solutions are also surveyed.

### 2.1 Survey on the Internet of Things

The legacy Internet has rapidly evolved in the past few decades from a simple platform for static web page transfers to a dynamic network consisting of multiple entities interacting and exchanging information. Sensor networks and near field wireless communication technologies have also seen a significant surge recently. Hence, the overall goal of IoT is to essentially combine the advantages of these two technology domains and create an efficient system where machines can operate autonomously in a self-organizing fashion. Indeed, the global ubiquitous computing model along with the context-aware nature of IoT devices is bound to provide many benefits for both individuals and organizations [6].

However the current standards and protocols developed for the Internet are not sufficient for the IoT sector owing to its heterogeneous nature and the low resource constraints of its tiny embedded devices. Thus, the research community and various standardization bodies have concentrated their efforts on developing solutions that are better-suited for the IoT ecosystem.

In particular, the following is a brief summary of some of the encapsulation, routing and session layer protocols that have been developed (or improved upon) to suit better the diverse nature of IoT:

- 6LowPAN: Protocol to enable IPv6 packets to work effectively in low-powered and lossy environments (IEEE 802.15.4)
- IEEE 802.15.4-206: Layer 1 or physical layer specifications for lossy networks
- Constrained Application Protocol (CoAP): Representation State Transfer (REST) based protocol for web-based communication with low-powered IoT devices
- Zigbee: Communication protocol built on the IEEE 802.15.4-2003 standard to enable the creation of low-rate ad-hoc wireless networks
- MQTT: A publish/subscribe architecture for asynchronous transfer of telemetry messages for low power and resource-constrained networks
- RPL: Distance vector routing protocol for lossy IoT networks
- IPv6 over Bluetooth: Low energy bluetooth-based solution using 6LowPAN compression techniques
- BACnet: Communication protocols for use in supervisory control and data acquisition (SCADA) and industrial control systems (ICS) systems

As seen above, the efforts in this domain have primarily focused on improving communication methodologies in light of the resource requirements of an IoT network. However, middleware is another important aspect of IoT communications. Namely, middleware, as defined in [7], is a layer of abstraction between the devices and the applications that use them. The primary motivation for using middleware has been the fact that creating customized applications for each device is a challenging task owing to the sheer number and variety of

IoT devices. Hence, middleware enables much easier integration with legacy technologies as well as rapid development of new ones. Along these lines, [7] discusses the individual components that collectively form the IoT middleware, a brief summary of which is provided below:

- Service Composition: Top layer that enables composition of individual services for creating specific applications
- Service Management: Focuses on the management of services and devices in the IoT ecosystem
- Object Abstraction: Enables standard web-based communication in an IoT environment by wrapping individual objects under a common framework to simplify accessibility

Akin to [7], a number of other studies have explored various trends and developments in the IoT domain. For example, [8] provides a comprehensive survey of IoT technologies in which the authors discuss various advances in hardware and application development. This work also points out the various challenges encountered when integrating IoT devices with conventional systems. However, this survey fails to provide any in-depth analysis of security concerns for IoT devices (although the authors do touch upon issues of data privacy and authentication). Meanwhile, [9] analyzes some important wireless communication protocols and also discusses some applications for IoT, e.g., smart parking, augmented maps, logistics, etc.

Overall, the general trend in IoT research has been to focus on issues related to the interoperability between various devices, protocols, and the applications built on top of them. By contrast, security in this domain has only now started to gain traction in the past few years, i.e., owing to the increasing severity of DDoS attacks as well as privacy concerns raised by the vulnerability of these pervasive devices. However, the number of IoT devices is only

expected to increase with the move to a more digital and automated future. Furthermore, the integration of legacy systems with IoT devices will also give rise to hitherto unforeseen security vulnerabilities.

Hence, in light of the above, conventional security standards and policies need to be reinvented. Furthermore, new developments in data analytics and networking also need to be adopted to provide more pro-active identification of IoT threats and rapid remediation strategies.

## 2.2 Survey on IoT Security Threats

Although the market for IoT devices has been expanding rapidly, in the rush to realize the vision of connected things, security concerns have largely been neglected. This is similar to the initial days of the Internet, where the main efforts were focused on improving connectivity and scalability. However, as noted earlier, the traditional security triad of confidentiality, integrity and availability (which was sufficient for the Internet) fails to meet new challenges in the IoT sector. Particularly, issues of privacy and authentication, along with access control, also need to be taken into account due to the rising popularity of service models such as Infrastructure-as-a Service (IAAS) and Software-as-a-Service (SAAS) for provisioning IoT devices. Specifically, in these frameworks personal user data is stored and managed by a separate entity. Furthermore, most traditional security policies and methodologies need to be re-evaluated to factor in the new and emerging IoT protocols and standards discussed in Section 2.1, as well as the constrained computational resources of IoT devices. Now the IoT itself is traditionally divided into 3 different layers, i.e., termed as perception, network and application [10]. As such, each layer has its own set of technological and security challenges, which are now further discussed.

The perception layer is primarily composed of sensing elements which monitor values such as the temperature, motion, humidity, etc. This acquired data can then be processed (if the

sensing element has adequate computational capability) and appropriately forwarded to the network layer. Meanwhile, the network layer focuses on routing collected data (from the perception layer) to various other devices, and in some cases, cloud service providers on the Internet. Finally, the application layer is concerned with the applications and services built on top of IoT devices and serves as the primary interface between the user and the device. Bearing the above setup in mind, some recent literature analyzing IoT security principles, technologies and possible countermeasures is now reviewed.

The authors in [11] provide an in depth analysis of the security requirements in an IoT environment. Most importantly, they discuss the need to consider other challenges to IoT security in the form of authentication, heterogeneity, policy and key management systems, etc. Furthermore, they also analyze inherent security issues associated with each layer in the IoT framework, namely perception, network and application. The authors then provide a high-level overview of some potential countermeasures to the various security concerns identified in recent literature. However these discussions focus on a few lightweight authentication and access control measures, as well as key management issues for trust establishment. The authors also emphasize the need for universal standards and policies, as well as strict enforcement to ensure optimum performance and security in a heterogeneous IoT network.

Similarly, [12] discusses and analyzes the various layers, sub-layers and related functionalities in the IoT architecture. The authors also highlight some potential attacks on each of these layers. For example, perception layer threats can include attacks on availability, authentication and service integrity as seen in WSNs. RFID security issues, such as tag disabling, cloning and tracking, are also mentioned. Similarly, attacks on the network and application layer (such as jamming, flooding, MITM and DoS) are also discussed. However, further measures to secure IoT devices from the aforementioned threats are not covered in this paper.

Meanwhile, [13] provides a unique analysis of security in the IoT domain. Namely, this paper proposes to use the unique features of IoT devices to identify potential security and privacy concerns in the network. The argument here is that since IoT devices and the environments in which they operate are significantly different from traditional computer networks, any security analysis needs to incorporate these features. For example, the features being analyzed here include interdependence, diversity, intimacy, ubiquitousness, mobility, unattended access, constrained environments, and heterogeneity. The authors then provide a detailed description of each of these features, along with a discussion of the associated threats and security challenges for each. In addition, they also review recent research literature that addresses some of the issues highlighted above and point to promising areas for future research.

Other recent studies have also analyzed security issues for specific IoT applications. For example, [14] and [15] consider security requirements in IoT-enabled smart grids. Namely, [14] provides a comprehensive discussion of several security issues seen in smart grids such as spoofing, eavesdropping, DoS and privacy issues. Similarly, [15] presents a survey of existing methodologies being proposed to improve security and privacy in such infrastructures. Additionally, the authors also try to identify some challenges across different domains in the smart grid architecture, primarily focusing on privacy issues and network layer attacks. Meanwhile, [16] presents a case study of two IoT devices that are commonly used in consumer and industrial IoT systems, i.e., a smart care hub for home automation and a smart meter for smart grids. For each device, the authors examine the underlying hardware being used and thereafter proceed to identify a few attack vectors. They also discuss the safety, security and privacy implications of compromising such devices.

Overall, the above studies reaffirm the need to study IoT security from a fresh perspective, i.e., to take into account the unique security challenges encountered in such settings. A few potential solutions here include frequent and secure update mechanisms, anonymizing user



identification data, as well as secure hardware and design principles. In addition, a proactive strategy to identify various IoT threats also needs to be developed. These studies also highlight the need to carefully address privacy considerations for users in light of the ubiquitous and pervasive nature of IoT devices.

### **2.3 Survey on Machine/ Deep Learning Applications in IoT**

Machine learning (ML) is a subset of artificial intelligence (AI) that learns a predictive model without being explicitly programmed to do so. Specifically, the goal of ML-based algorithms is to optimize a certain objective function, often called the error or loss function, in an iterative manner using a set of training samples. Such models also consist of various hyper-parameters that can be fine-tuned to improve prediction accuracy. Now in the past, ML applications were somewhat limited due to the high computation and storage requirements of the data needed to train and test related models. However, today this field is seeing much renewed interest from applications in various industries such as finance, health care, web search engines, etc. Again, this renewed focus can be attributed to improvements in enabling hardware and software technologies (that are also responsible for the growth of the IoT technology sector and market).

Furthermore, deep learning (DL) is a form of ML that mimics the functioning of the human brain. Specifically, this approach uses multiple hidden layers of neurons and can excel at learning from complex high-dimensional data sets to produce highly accurate prediction models. The uses of ML have been further broadened by advances in graphical processor unit (GPU) technologies. Namely, GPUs have evolved from supporting mostly video games into more generalized computing-type roles. These devices differ from CPUs in that they support a large number of simple cores and a significantly large number of concurrent hardware threads, thereby enabling much more efficient and faster computations. As such, these devices are very amenable to DL models (neural networks with a large number of nodes)

that need to perform large amounts of calculations during the learning phase. Finally, the massive amounts of generated data these days (also known as big data) has enabled researchers to overcome the issue of overfitting, which is a common limitation arising due to smaller sample sizes.

Overall, ML algorithms can be broadly classified into two key categories, namely unsupervised and supervised learning. A brief description of the two is provided below:

- Unsupervised learning: The main objective of these techniques is to identify natural patterns or structures in unlabelled data, e.g., or in other words, the expected outcome value is unknown. Unsupervised learning can also be used for dimensionality reduction, which is especially useful when the number of features for a given data set may be too large.
- Supervised learning: The main objective of these techniques is to identify relationships or patterns that help predict the correct labels for a given data set. Hence these learning models are supplemented with labels to help transfer some domain knowledge to the data sets. These algorithms are mainly used for classification and regression problems.

In general, a review of the core literature in ML and DL reveals that most studies have focused on improving performance for applications such as image classification, linguistics and audio tasks. However, recent efforts have also started to explore the applications of such techniques in other domains as well, including cybersecurity. For example, [17] presents a detailed survey on the applications of data mining and ML methods in cybersecurity. Namely, the paper reviews the various steps involved here, which include data pre-processing, feature extraction, class labelling as well as training and testing. The paper also discusses some of the typical metrics used to ascertain the predicting capabilities of the model, e.g., such as accuracy, sensitivity, false accuracy rate (FAR), etc. The authors also review some

of the common data types and data sets being used by researchers in the cybersecurity domain and provide a detailed analysis of the more popular ML and DL algorithms. Finally, important observations and challenges in applying such techniques to cybersecurity problems are discussed and recommendations are made for selecting the most appropriate algorithms for a specific task.

Meanwhile, in [18] the authors analyze ML applications for WSNs and present a brief overview of three main types of models, i.e., supervised, unsupervised and reinforcement learning. Related ML literature that addresses some of the functional challenges seen in WSNs is also reviewed, e.g., such as activity recognition, query optimization, medium access control (MAC) protocol design, broadcast scheduling, etc. The paper also highlights the challenges in applying ML algorithms to the IoT domain and identifies areas for future research. In [19] the authors present a statistical model that tries to characterize IoT devices based upon their network traffic. Essentially, this paper proposes a behavioral model using network traffic generated by IoT devices that are typically seen in smart cities and campuses. In particular the authors collect network data from 28 unique IoT devices and generate a statistical model that clusters data using the K-means algorithm for identification.

Meanwhile, [20] proposes a cloud-based offloading mechanism for malware detection. This work demonstrates the tradeoff between transmission cost and detection performance as seen in the radio channels of mobile devices. Additionally, the authors also propose three reinforcement learning techniques using Q-learning algorithms and related variants to improve detection accuracy and bandwidth utilization (by offloading application traces to cloud-based servers). Similarly, in [21] the authors analyze various security issues in WSNs such as DoS, spoofing, eavesdropping, etc. Furthermore, they also propose a multi-layered computational intelligence based architecture to be used as a wireless intrusion detection system (IDS). The advantages of this design are also outlined, e.g., such as robustness, adaptability, faster detection times, etc. Finally, in [22] the authors identify physical layer

spoofing attacks as a zero sum game of authentication between the adversary and WSN nodes. The Nash-Equilibrium (NE) of this game is then derived and akin to [20], Q-learning techniques are used to identify attacks based upon the uniqueness of the channel state of the transmitter. Overall, the above works attempt to address security concerns for WSNs and also present some unique solutions for certain functional challenges as well. However, security vulnerabilities of other communication technologies and protocols that are typically seen in IoT environments are not addressed here.

Meanwhile, in [23] the authors propose a classification algorithm that can distinguish between IoT and non-IoT devices using ML and network traffic characteristics. Namely three popular ensemble learners are used here (i.e., gradient boosting, random forest and XGBoost) in a multistage process. The authors also test their algorithms on 9 commercial IoT devices. Overall, this solution demonstrates high accuracy in identifying IoT devices from non-IoT data in a supervised learning environment. However, the lack of sufficient non-IoT data and the reliance on just the hyper-text transfer protocol (HTTP) protocol can potentially lead to overfitting. Along these lines, the authors acknowledge the need to include other protocols and test system scalability in future research. Furthermore, [24] presents another example of an application of Q-learning for WSN security. Namely, this model proposes a Markov-based framework that models a 2 player game between the sensor and the attacker, and results show high accuracy in identifying signal-to-interference-plus noise ratio (SINR) based DoS attacks on WSN nodes.

## 2.4 Applications in Intrusion Detection System

Network traffic characteristics are extensively used by various intrusion detection and prevention systems (IDS/IPS) to identify threats to general computing systems. Specifically, these detection systems are generally divided into the following two major categories.

- Signature Matching: This mechanism identifies the key features of a malware and generates a signature based upon these features to identify its future presence.
- Anomaly Detection: This mechanism generates a “normal” profile for the devices in a network and flags any deviations from this behavior as an anomaly.

As expected, a signature-based IDS needs to constantly adapt to an evolving threat landscape. However, such a methodology can potentially fail against zero-day or never-before-seen attacks. On the other hand, anomaly-based mechanisms identify deviations in the “normal behavior” of a device and hence are better at identifying previously-unseen attacks. However, such methods tend to produce a higher number of false positives, i.e., they can be very sensitive to changes in behavior profiles.

Accordingly, the authors in [25] use ML algorithms to classify malware on the Internet by observing network data packet payloads. Specifically, they use the Random Forest, Naive Bayes’ and J48 learning algorithms to identify malware activities using multiple network protocols and different resolution of network activities. However, these schemes can suffer from high operational overheads and fail to identify new threats (whose characteristics deviate from the family of malware used for training). Additionally this research does not include any provisions for IoT devices. Meanwhile, [26] presents a selection methodology using the chi-square method to identify the best features in a given feature set for a payload-based IDS. These selected features are then used in a multi-class support vector machine (SVM) classifier, and the results show improved performance as compared to other payload-based classifiers (on the NSL-KDD and KDDCup 1999 datasets).

Additionally, [27] proposes the use of recurrent neural networks (RNN) to identify anomalies in network activity. Namely, the authors liken network communication between two computers to human speech, wherein grammar rules are represented by the underlying protocols and services being supported. This interaction is then modeled using a long short-term memory (LSTM) RNN that uses flow sequences derived for every pair of IP addresses com-

municating in the network. The model is then trained using benign traffic, and flows are generated based upon sequences of port and protocol numbers. Note that two different datasets are used here, one which contains no attack traffic and another which includes some attack traffic. The overall results show that ML or DL models can provide an efficient tool to detect anomalous device behavior. In [28] the authors also explore the use of artificial immune systems to mitigate DoS attacks. Namely, the proposed solution is modeled after the behavior of human biological defense mechanisms, and a distributed sensor network is used to monitor the environment and coordinate and improve defense capabilities against DoS attacks.

Furthermore, [29] presents a study on the behavioral analysis of darknet data to identify malware activities of infected hosts. A data mining approach is then used for early identification of DDoS attacks by extracting features based upon network traffic characteristics. This work shows how darknet data, when combined with data mining techniques, can help identify Internet-scale malicious activities. Also, in [30] the authors use association rule extraction to detect anomalies and identify recurrent packets in the data with features similar to NetFlow. However, this solution does not provide automatic traffic classification and requires human expertise to analyze data and draw useful conclusions. Furthermore, no IoT-specific concerns are addressed in this work either.

In [31] the authors propose an IDS called SVELTE (literally meaning elegantly slim) for IoT networks. However, this system is only applicable to 6LoWPAN IoT networks using the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL). The authors also evaluate their approach using the Contiki operating system. Meanwhile [32] presents an anomaly detection framework for IoT devices. The algorithm essentially generates a normal profile for devices by using bit pattern matching of payload data. Thereafter, any deviations are flagged as anomalous. Similarly, the work in [21] tables a computational intelligence framework to generate behavior profiles for devices in a wireless system environment. Meanwhile, the

authors in [33] also use ML methods to detect suspicious IoT devices connected to a network through the use of static whitelists within the enterprise (which define allowable devices). However this scheme can easily be bypassed given (unauthorized) access to the whitelist. Furthermore, the authors do not consider other malicious activities that may target IoT devices from outside the enterprise domain.

An artificial neural network (ANN) based IDS for IoT devices is also presented in [34]. However, this work only focuses on the identification of a single type of attack i.e., DoS and DDoS attack. Furthermore, this scheme is not very computationally-efficient and may not be suitable for online learning. Similarly, [35] also uses ANNs to identify malicious IoT activity. The experimental setup here consists of an IoT testbed emulating node devices and a Raspberry Pi device which serves as a gateway for these nodes. However, this methodology requires a sufficient number of data points to validate its efficacy. Although the authors introduce invalid data points to define anomalous activity, they do not provide a solid explanation of what constitutes an anomaly for devices used in their testbed. Overall, the lack of sufficient data points has been known to cause overfitting in the case of ANNs with many tunable parameters. Furthermore, [34] uses an ANN to detect unsolicited IoT behavior. However, this approach uses simulated data which again is not necessarily a very accurate representation of real world attack patterns. Clearly the lack of realistic empirical IoT data can be a significant hindrance in testing the efficacy of any proposed intrusion and anomaly detection systems.

To address some of the above concerns, the authors in [36] propose a DL model that is trained and tested on the NSL-KDD dataset. The algorithm is primarily meant for fog networks, i.e., using a distributed attack detection architecture consisting of multiple fog nodes running the DL algorithm. Namely, these nodes are managed by a centralized master node to share model parameters and optimize performance. However, even though the model exhibits good accuracy, the dataset used here is quite old and does not include any IoT-

specific provisions. Furthermore, as shown in [37], this particular dataset is not necessarily a good representation of real world network traffic.

Finally, [38] uses autoencoders to identify malware, such as Mirai and Bashlite, which exploit port scanning and brute force authentication to infect IoT devices and launch large scale DDoS attacks. Namely, this work uses auto-encoders, which are known to be good inference machines, and trains them using network traffic statistics for detecting such attacks. Specifically, an auto-encoder is trained for each device in the network and a *normal* profile is also generated for these devices. Thus, when a device is infected, the previously trained auto-encoder will flag it as an anomaly. However, this approach has the added operational overhead of training an auto-encoder for every possible device in the network (and hence may not scale in large networks).

Overall, the above studies show the efficiency of ML and DL models for detecting security vulnerabilities in IoT devices. However it is also quite apparent that a lack of sufficient malicious training data is a major shortcoming when developing these models and this can potentially lead to over-fitting. As such, many of the proposed methodologies will be ineffective against zero-day attacks. Furthermore, some of the proposed solutions also carry additional processing and storage overheads making them unsuitable for real-time detection.

## 2.5 Survey on SDN/NFV Solutions for IoT Security

The large amount of data generated by millions of IoT devices and their varying communication needs are bound to stress current networking infrastructures. Furthermore, the move by many cloud providers to offer SAAS offerings for IoT users further complicates matters. However, SDN and NFV are two novel networking paradigms that have been developed to address many of the scalability, management and security issues encountered in traditional networks. As a result, these frameworks enable network administrators to implement policies that can dynamically adapt to network requirements. Furthermore, SDN



also separates the control and data planes, thereby providing a global view of the network with routing/switching elements that can be programmatically configured using a centralized controller. Meanwhile, NFV on the other hand virtualizes conventional networking functions such as firewall, load balancers, etc. As such, NFV-enabled setups allow dynamic service provisioning with greatly-reduced operational costs, i.e., as specialized hardware is no longer required (and a single server can host multiple virtual functions).

In light of the above, researchers have also explored the potential applications of SDN and NFV concepts in various networking environments and analyzed the security and performance implications of these software-centric strategies. Accordingly, [39] enumerates the various advantages of this framework, e.g., such as scalability, granularity, service chaining and improved security capabilities. The authors also detail multiple approaches that leverage these technologies along with ML classification methods for implementing a stateful firewall in an enterprise network. Finally, they also discuss various challenges in implementing a SDN/NFV-based firewall design with a particular focus on security issues at the control plane and performance limitations of flow-based policy enforcement. Meanwhile, [40] proposes a priority-based algorithm to handle DoS attacks launched against the SDN controller. Akin to an anomaly detection scheme, this solution defines normal user behaviors and assigns priorities to new flows by using a ranking algorithm that compares them to benign profiles (generated earlier). Hence in the case of a DoS attack, the algorithm follows a differential policy by assigning lower priorities to attack traffic and higher priorities to legitimate users.

Meanwhile, numerous other studies have also tried to leverage SDN and NFV solutions to improve the performance and security of IoT networks. For example, [41] addresses potential “new-flow” attacks in SDN-based IoT networks. Specifically, this attack is a particular type of DoS attack that attempts to exhaust the resources in a switching element by sending many packets that have no flow-rules associated with them. Hence a new flow attack can exhaust the caching and bandwidth capacity of switching elements, ultimately leading to connectivity

issues for legitimate IoT devices in the network. The paper also proposes a smart security mechanism which proactively mitigates such attacks by redirecting anomalous traffic to a security middleware.

Meanwhile, [42] proposes a Black SDN security architecture for IoT where the controller serves as a trusted third party to manage network security and performance optimization. Essentially, the framework encrypts the header and payload data of all packets, thereby making it very difficult to infer any important information from user traffic in case of MITM or inference attacks. Furthermore, any routing issues arising from encrypted headers are handled by using a relatively simple broadcasting scheme. However, this work is primarily focused only on the IEEE 802.15.4 protocol and imposes additional encryption overheads which may affect performance in large scale networks.

The authors in [43] also leverage/apply emerging block chain technology within the SDN framework to secure IoT devices. Namely, a DistBlockNet scheme is introduced, comprising of multiple modules (such as OrchApp, Shelter, etc.) that dynamically adapt to network changes and quickly identify and remediate attacks. The proposed architecture also uses a distributed block chain network to securely update flow tables for all IoT forwarding devices. Similarly, [44] proposes a distributed SDN architecture for IoT security. The concept is similar to the grid of security network and utilizes a border (or security) controller that is responsible for secure intra-SDN domain communication.

Finally, [45] uses a role-based security mechanism to improve security in IoT networks. Namely, this framework implements different control functionalities (such as access control, key management, intrusion detection, etc.) on multiple controllers, thereby eliminating the problem of a single-point-of-failure and improving bandwidth utilization. However, the authors admit that the proposed approach imposes consistency concerns and has high communications overhead, which may affect its overall performance. Other works like [46] and [47] have also explored SDN/NFV frameworks to improve network security and address

scalability issues. Overall, these studies affirm the importance of a centralized management philosophy with software implementations to improve efficiency and security in computer and IoT networks.

## 2.6 Open Challenges

Overall, the above literature review clearly indicates the growing need to secure IoT devices. Indeed, the heterogeneity and massive number of such devices makes it very difficult to provide a one-fits-all solution. Furthermore, the lack of sufficient data (due to logistic and privacy concerns) makes it even more challenging to infer large scale malicious activities of IoT devices (and identify attacker trends and behavior). This lack of data also makes it very hard to build effective ML and DL models, i.e., since these algorithms generally require a sufficiently large amount of data to improve prediction accuracy and avoid generalization. Furthermore many of the above works do not include popular and disruptive malware which use brute force and other covert mechanisms to infect devices and launch large scale DDoS attacks, e.g., such as Mirai, Reaper, etc. Therefore any security framework needs to take into account not only the dynamic nature of IoT, but also evolving attacker behaviors which are critical for identifying zero-day threats.

Finally as noted, SDN and NFV technology paradigms have also emerged as promising security solutions and are rapidly challenging conventional networking practices. As a result, several recent research studies have also tried to address various security challenges in IoT devices by leveraging these concepts. However, further challenges relating to overhead and latency are critical areas that need to be addressed in such cases in order to ensure efficient and optimized IoT operation.

### 3 Correlating Active and Passive Measurements to Infer and Characterize Unsolicited IoT Devices<sup>1</sup>

The seamless interconnectivity of IoT devices with the physical world is delivering many improvements across diverse application domains. However, compromised IoT devices pose a significant and growing threat to user data integrity and privacy. Furthermore, as noted in Chapter 2.2, when coupled with the highly-interconnected nature of IoT setups, even a few poorly secured Internet-connected devices could potentially affect the security and the resilience of the Internet at a global scale, not just locally. For example, an unprotected camera or smart home hub infected by malware might send thousands of harmful spam emails to worldwide recipients using the owner’s home Wi-Fi Internet connection [48]. Alternatively, it may also try to infect other devices with the same malware. Additionally, exploited Internet-scale IoT devices can be leveraged by an attacker to orchestrate malicious botnets, causing immense damage to corporate and Internet services via DDoS attacks or other typical misdemeanors [48]. In fact, examples of such Internet-scale attacks already exist as evidenced by malware such as Mirai and its variants.

In light of the above, the imperative tasks of quantifying, characterizing and attributing such vulnerable IoT devices will deliver a vital first step towards uncovering their inherent vulnerabilities and understanding their malicious behavior [48]. Hence this chapter leverages active measurements through (Internet-wide scanning) in conjunction with passive measurements (in the context of darknet traffic analysis) to shed further light on compromised de-

---

<sup>1</sup>Parts of this chapter were published in F. Shaikh et al. Internet of Malicious Things: Correlating Active and Passive Measurements For Inferring and Characterizing Internet-scale Unsolicited IoT Devices. International Wireless Communication and Mobile Computing Conference, 2018. Permission included in Appendix B

vices and analyze their unsolicited network traffic characteristics. Specifically the following contributions are made here:

- An innovative approach to infer, characterize and attribute unsolicited Internet-scale IoT devices by correlating passive and active empirical measurements.
- Generate IoT-specific malicious signatures by scrutinizing passive measurements. These signatures, which are based upon fuzzy hashing techniques, can then be deployed in local IoT realms for effective mitigation as well as for inferring other Internet-wide unsolicited IoT devices.
- Analysis of close to half a million compromised IoT devices in smart home appliances, critical infrastructures and automated control sector environments.

In a nutshell, as noted in [48], the proposed approach endeavors to generate actionable cyber threat intelligence related to Internet-scale IoT devices by applying several data-driven methodologies. These schemes mainly operate by scrutinizing passive empirical measurements. Overall, the garnered insights and inferences can be distributed, in an operational cyber-security fashion, to IoT stakeholders (i.e., operators, manufacturers, etc.) to assist with prompt remediation and thus mitigation support. Therefore, an additional artifact of the envisioned approach is also a repository which aims to index malicious IoT empirical threat information to be shared with the research and operational communities at large. Indeed such a facility can help provision advanced IoT threat analytics as well as support further forensic investigations.

### 3.1 Notation Overview

Before presenting the details of the proposed scheme, the requisite notation is introduced first. Namely,  $D$  is a list of IP addresses obtained from the CAIDA network telescope. Meanwhile,  $A$  is defined as a list of IP addresses obtained from active measurements in the

Shodan and Censys databases. Furthermore,  $H$  denotes the hash table which stores the value of the hash function for IP addresses obtained from the Shodan and Censys databases. Finally,  $h(x)$  is the hash function, and  $C$  is a list of all correlated IP addresses.

Table 1: List of variables

Variable	Description
$D$	List of passive measurement IP addresses
$A$	List of active measurement IP addresses
$ip$	IP address in $D \cup A$
$H$	Hash table
$h(x)$	Hash function
$C$	Correlated IP addresses

### 3.2 Inferring and Characterizing Internet-Scale Compromised IoT Devices

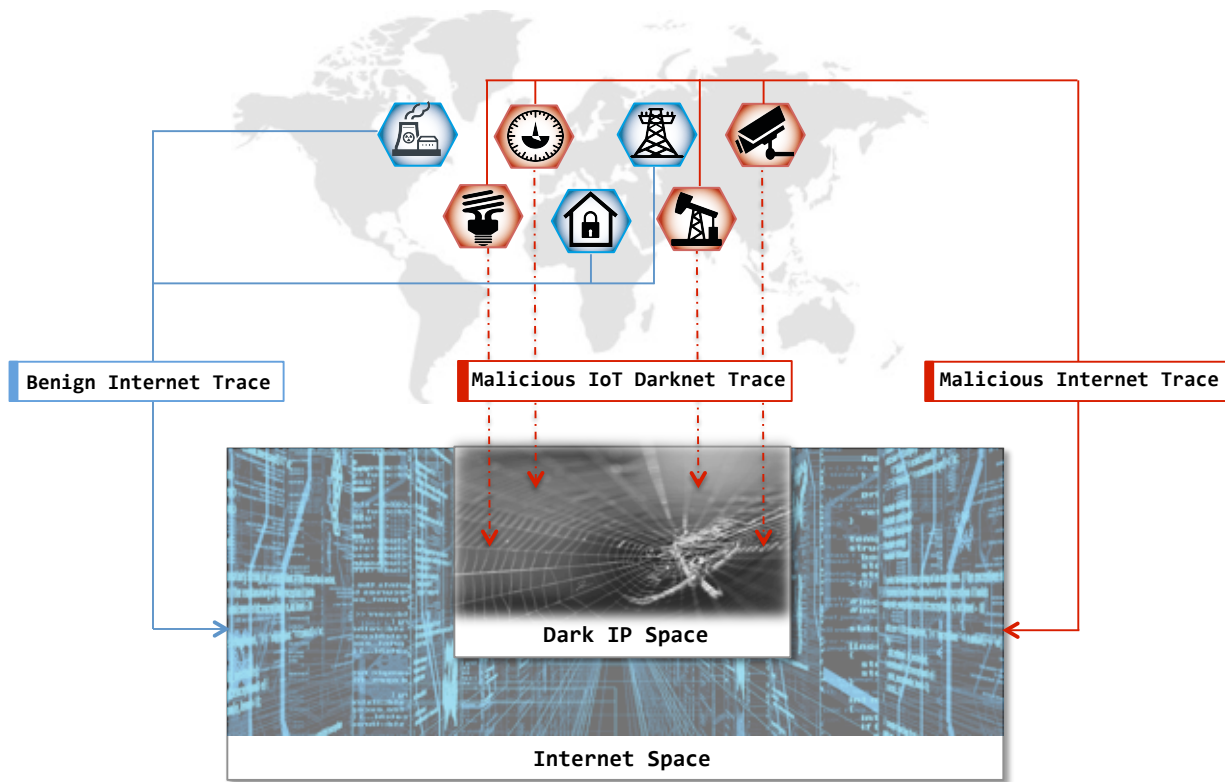
The heterogenous nature of IoT devices and their wide-scale deployment in different environments (in addition to lack of visibility into local IoT realms) makes it very difficult to generate substantial insights to characterize unsolicited IoT behaviors [48]. Therefore any attempts to infer and characterize such IoT devices can be highly impactful and beneficial, as they can effectively pinpoint infected devices to improve mitigation strategies. Overall, the final objective here is to thwart malicious activity by adversaries and prevent them from recruiting infected IoT devices as part of a bot network. In this context, a macroscopic approach to infer Internet-wide malicious IoT devices is further realized by correlating passive empirical passive measurements as well as active Internet measurements. This macroscopic approach enables inferring large scale malicious IoT behaviors as well as evolving attacker strategies.

### 3.2.1 Exploiting Network Telescopes

Network telescopes, also known as darknets, constitute a set of routable and allocated, but unused Internet Protocol (IP) addresses. Indeed darknet IP addresses provide a means to passively measure Internet-scale maliciousness of devices, i.e., due to the fact that there are no legitimate services or real devices that are actually using these addresses. Therefore, any traffic with a destination IP address in the darknet is surely considered to be unsolicited. Now darknets are commonly distributed on specific Internet IP address subspaces, e.g., usually operated by Internet Service Providers (ISPs), educational and research entities, and corporate and backbone networks. However, darknet IP addresses are, by nature, indistinguishable from other routable addresses, rendering them as an effective means to amalgamate Internet-wide, one-way unsolicited network traffic. Along those lines, network telescopes are used to identify network traffic originating from unsolicited IoT devices.

The proposed solution is illustrated in Figure 3.1. The overall rationale here is rendered by initial empirical observations which confirm the hypothesis that, as compared to typical Internet hosts, exploited IoT devices will attempt to propagate and infect other online IoT devices. Specifically, related actions can include launching scanning activities towards the Internet space or initiating DDoS attacks. In fact, most recently, malware such as Mirai, Bashlite etc., have exhibited similar behaviors, wherein a central command and control (C&C) setup is used to recruit infected devices in a bot network. These devices can then be used to launch large scale DDoS attacks against targets, and these bots can also continue to scan the Internet to infect other vulnerable devices.

In either case, depending upon the vantage points of the employed network telescopes, a varying portion of such malicious activities will indeed target darknet IP address spaces. To this end, well-established algorithms, methods, and techniques can be leveraged to scrutinize darknet data, also termed as Internet Background Radiation (IBR), i.e., to fingerprint such activities and infer sources of unsolicited IoT devices. In addition, corresponding darknet



Copyright © 2018, IEEE

Figure 3.1: Network telescopes capturing Internet-scale IoT unsolicited traffic

traffic traces [48] can also be extracted here. For instance, such algorithms could leverage threshold analysis to infer IoT-generated scanning activities or exploit backscattered packet analysis (i.e., by analyzing reply packets originating from victims of DDoS attacks that have been targeted by spoofed attackers to pinpoint targeted IoT devices).

In order to successfully recognize DDoS activities targeting IoT devices, this work analyzes close to 1.5 TB of real darknet data obtained from a  $/8$  network telescope, as provided by the CAIDA facility [49]. Carefully note that the proposed approach can only characterize malicious IoT devices whose traffic targets the darknet, a limitation typically encountered in network telescope analysis.



### 3.3 Leveraging Internet-Wide Scanning

Although empirical darknet data can help characterize Internet-scale malicious activities, it still may not yield any insights into unsolicited IoT behaviors. Therefore, it is imperative to also filter darknet sessions that originate from IoT devices. Now the task of fingerprinting many different types of IoT devices by analyzing IBR is an open research challenge. Accordingly, this issue is addressed here by correlating packet information from the CAIDA darknet data repository with data measurements obtained from Internet-wide scanning of active IoT devices. Note that this correlation is performed mainly using source IP addresses and transport layer protocol information (of the darknet sessions). Furthermore, the task of identifying active IoT devices involves continuous probing of the Internet space for all active devices, including IoT devices. Clearly, this approach is less efficient and will be unable to provide real-time cyber operational capabilities which are crucial for mitigating large scale orchestrated attacks.

In light of the above, instead of performing continuous Internet probing, measurement archives from the Censys and Shodan facilities are used for correlation with darknet data. Namely, both of these facilities probe the Internet for connected IoT devices and provide this information in comma separated value (CSV) files. Specifically, the Shodan database is built by scanning the entire IPv4 address space and usually provides IP addresses as well as the type of device, operating system, etc. Meanwhile the Censys facility was developed by a team at the University of Michigan and also supports another search engine to expand the number of IP addresses and help consolidate results. Overall, 275,478 online IoT IP addresses related to various device types, including, home automation devices, IoT cameras and industrial control devices are gathered as noted in [48]

### 3.4 Correlating Active and Passive Measurements

In order to identify darknet sessions which belong to IoT devices, a novel correlation algorithm is designed and implemented using Python, the overall pseudocode of which is presented in Figure 3.2. Now given a list of source IP addresses in the darknet space and a list of IoT-specific IP addresses from the IoT databases, a rudimentary linear search can be performed i.e., to compare each IP address obtained from the passive measurements to those from the active measurements. However, the time complexity of such an algorithm is of order  $O(n^2)$ , and this is clearly unscalable for most high-volume scenarios. Hence in order to address this concern, a hash table approach is developed based upon the active IP addresses from Shodan. The passive measurements are then correlated with the active measurements by searching the hash table for their associated entries, Figure 3.2. As a result, the resultant run-time complexity is now reduced from quadratic to linear, i.e.,  $O(n)$  since hash lookup is an expected constant time operation.

Furthermore, another key challenge in correlating the two measurements is the need to properly sanitize darknet data and filter out misconfiguration traffic. Namely traffic can often be erroneously directed to the darknet due to software, hardware or routing errors. Therefore such sessions need to be proactively identified and removed before any correlation search is performed. Another concern relating to the Shodan and Censys databases is how to identify relevant IoT device information. This identification can include device type, IP address, geographical location, etc. Furthermore, there is also a need to properly download, sanitize and store the obtained information. In general, the actual design of the correlation algorithm is optimized for efficiency, especially since it is envisioned that such information will be made available to the general research community and industry partners.

```

1: INPUT: D,A
2: OUTPUT: C
3: for all  $ip \in A$  do                                ▷ Convert list of active IoT IP addresses to a hash table
4:    $H \leftarrow h(ip)$ 
5: end for
6: for all  $ip \in D$  do                                ▷ Correlate active and passive measurements
7:   if  $ip$  in  $H$ :
8:      $C \leftarrow ip$ 
9: end for
10: return  $C$                                           ▷ Return list of IoT IP addresses in darknet

```

Figure 3.2: Correlation algorithm for unsolicited IoT device identification

### 3.5 Generating IoT-Specific Malicious Signatures

Overall there is a notable lack of tangible malicious indicators that are derived from empirical data in the context of IoT devices [50]. This shortcoming is also noted in Chapter 2 and can be attributed to various physical and logistical constraints that are often strictly enforced by IoT operators. Additionally, analytic efforts in this domain (which aim at addressing the distributed and heterogenous nature of the IoT ecosystem) are not yet mature enough. In light of the above, there is an urgent need to address IoT security research and development concerns. Accordingly, as noted in [48], the proposed methodology here attempts to facilitate the generation of such artifacts. Specifically, performing network traffic correlation between passive measurements (from CAIDA) and active IoT IP addresses (from Shodan and Censys) can provide rare insights into Internet-scale activity of unsolicited IoT devices. In turn, new insights into compromised device behaviors can be used to develop signatures to help strengthen perimeter defenses for IoT networks. These signatures can also be employed on new darknet sessions to fingerprint devices that have not yet been indexed by databases such as Shodan and Censys. Finally, signatures can also be distributed to local IoT realms to be used as part of IDS or IPS frameworks.

To further support the aforementioned objectives, this effort uses the concept of fuzzy hashing by tailoring and applying the Context Triggered Piecewise Hashing (CTPH) algorithm [48] on darknet traces (that are generated by inferred unsolicited IoT devices). Overall, the CTPH algorithm only operates in the current context of the input, i.e., maintaining its state solely based upon the last few bytes of the data file. This operation ultimately produces a pseudo-random value as output. The algorithm also generates discrete hashes by dividing the file into multiple segments/blocks and computes hashes for these segments, i.e., instead of computing a single hash for the entire file.

Essentially, the modified CPTH scheme uses a pre-determined block size, and the data length is used to generate an initial block size. Subsequently, the Fowler-Noll-Vo (FNV) algorithm is used to generate hashes on each byte of data, i.e., this algorithm produces a 32-bit output for any input. Then upon reaching a specified trigger value, further base 64 encoding is done for the last 6 bytes and added to the final signature [51]. This process is repeated once again with twice the block size to improve the robustness of the algorithm. Once the algorithm has run the entire length of data, a unique signature is generated for the file. The algorithm then uses dynamic programming to calculate the edit distance, which is essentially the benchmark used to identify similarity. Namely, the edit distance can be any value between 0 and 100, with anything above 0 indicating correlation and 100 indicating almost identical files. By contrast, most of the traditional hashing algorithms (such as md5, SHA1 etc.) only generate binary outputs, i.e., even a single bit variation can provide two uncorrected hashes. Therefore, the CTPH algorithm is much more robust to localized changes and thus can provide a relative degree of similarity to better compare two files.

In the context of this work, the CPTH algorithm is used to generate signatures for IoT devices based upon their IP header information. Specifically, the network layer information of IoT devices found in the darknet (from the CAIDA data repository) is used here. The primary motivation here is the fact that Layer 3 information from infected devices can be

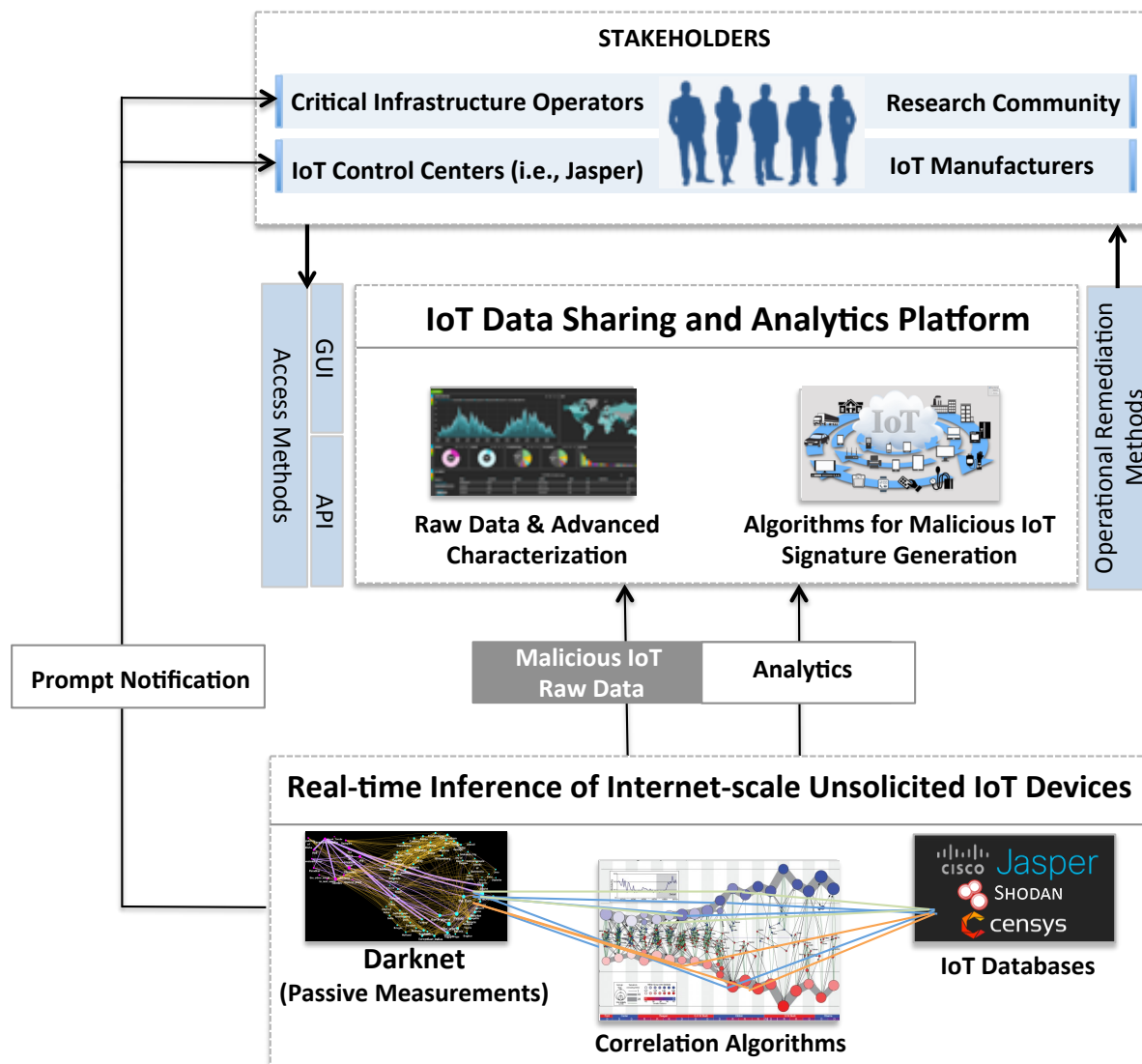
used to characterize them, i.e., since devices infected with (nearly) identical malware, viruses etc. will exhibit similar characteristics (with the obvious exclusion of source and destination IP addresses, checksum values, etc.).

Note that the time complexity of the basic CPTH algorithm is significantly higher than other traditional algorithms such as Whirpool or md5 [52]. Hence given the huge amount of darknet data involved, it is important to optimize the approach to increase efficiency and reduce comparison time. To accomplish this objective, a tool called ssdeep [53] is also used. Namely, this tool uses techniques (such as clustering and IntegerDB) to provide an optimum method to compare the hash generated by ssdeep. The overall benefits here include reduced memory requirements and faster processing times. Namely, the algorithm divides the hash values generated by the ssdeep tool into chunks of varying sizes [53]. This approach gives a reduced dataset since only files of the aforementioned chunk size need to be compared. The tool also goes one step further by identifying overlapping chunks to further reduce the number of comparisons, i.e., by only comparing the files that return a value greater than zero. Overall, a number of benchmark tests are done with large datasets and the results confirm the effectiveness of this proposed method.

### 3.6 Sharing of IoT Unsolicited Empirical Threat Information

In general the lack of real empirical threat information is a major hindrance when developing IoT security solutions. As a result it is proposed to publicly share the intelligence extracted from darknet data with all stakeholders in the research and development communities. Specifically, such a facility will provide:

- Raw unsolicited IoT traffic traces to support large-scale data analytics by leveraging rare empirical data
- Generated signatures to allow for further forensic IoT device investigations and use in local realms for proactive inference and mitigation.



Copyright © 2018, IEEE

Figure 3.3: Inferring and mitigating Internet-scale unsolicited IoT devices: A network telescope approach

Accordingly, the insights generated here are indexed in an accessible database hosted at the Florida Atlantic University. This setup includes near real-time information related to Internet-scale compromised IoT devices coupled with their geolocation information and generated signatures.

### 3.7 Empirical Evaluation

This section elaborates upon the extracted inferences and insights derived by employing the proposed methodology in Sections 3.2.1 and 3.3. Namely, passive darknet data and correlated active measurements are used to identify IoT devices that have been compromised on a global scale. In particular, IP addresses from the two databases (Shodan, CAIDA) are first extracted for the month of December 2016. Similarly, darknet data is also analyzed for the first 4 days of November 2016, December 2016, January 2017 and February 2017. On each of these days, the data retrieved is for a six hour duration.

Now in order to enable effective characterization of IoT devices, it is important to understand the type of devices that are most commonly found in the darknet (and if possible, the protocols and operating systems they are using). Fortunately, both Shodan and Censys provide a range of statistics for IoT devices, including the type of device, which can range from webcams, SCADA and Linux universal plug-and-play (UPnP) devices to modems, routers, etc. Additionally, in many cases these search engines also provide information on the type of operating system in use, associated ports and protocols, as well as ISP and manufacturer information. Some detailed findings are presented now.

Overall, close to 165,000 IoT devices deployed in various SCADA environments are extracted, as show in Figure 3.4b. This category represents 55% of all extracted devices. Meanwhile, DVRs and Bluetooth-enabled IoT devices are less prevalent, totaling close to 26% and 11% of the aggregate number of devices, respectively. The remaining 8% of investigated devices are rendered as IoT webcams and thermostats. As noted, darknet data from the CAIDA repository is extracted for the months of November and December 2016 as well as January and February 2017. However, only 30 hours of data is extracted and analyzed over the duration of these 4 months since CAIDA monitors a very large network telescope.

Upon executing the correlation algorithm between the two measurements, the results confirm the presence of almost 14,000 Internet-scale infected IoT devices. Foremost, this

figure validates the initial hypothesis that a significant number of malicious IoT devices exist in the wild, and that network telescopes provide an effective methodology to shed light on Internet-scale unsolicited device behaviors. Furthermore, the IoT devices in Figure 3.5a are also characterized in Figure 3.5b, and the results indicate that a significant portion of compromised IoT devices are in fact DVRs, i.e., 64.3%. This value corroborates with the rising exploitation of webcams and routers by the Mirai malware and its variants, i.e., to launch crippling attacks as seen on the Dyn DNS server which ultimately affected wider Internet connectivity [2]. Meanwhile, IoT devices deployed within SCADA realms (mostly belonging to building automation systems, power utilities, and manufacturing plants) also represent 28.4% of the total share of unsolicited devices found in the darknet. Again, this is a very notable figure since such devices are often deployed in vital roles. Indeed, the exploitation of such IoT devices may cause large scale economic losses and even endanger human lives.

Finally, IoT webcams, thermostats and other Bluetooth-enabled devices are also found to be compromised. Overall, the above inferences can be leveraged by the operational cybersecurity community (including IoT operators and manufacturers, cyber situational response teams, etc.) to aid in the rapid notification, and thus mitigation, of such attacks.

Finally, the hosting environments of the exploited devices are also characterized by identifying their geographical location, see Figure 3.6. This analysis reveals that most of the infected IoT devices are located in Asia, followed by Europe and the Americas. In particular countries in Asia seem to have a significant portion of infected video recorders, whereas SCADA-based IoT devices are most prevalent in Europe and the Americas. Furthermore, carefully note that more precise information, such as the hosting and ISP organizations, is also extracted. However these results are not revealed for privacy reasons.

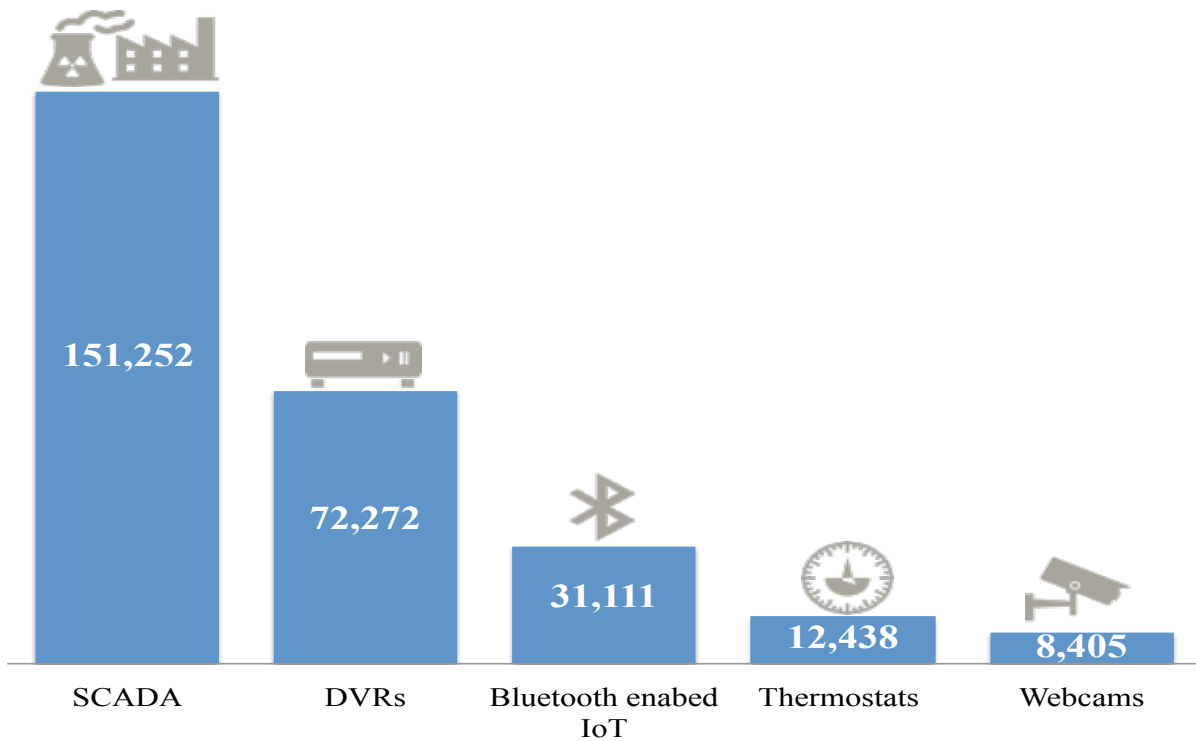
As detailed in Section 3.5, IoT-specific malicious signatures are also generated by employing an open source implementation of the CTPH algorithm, namely the ssdeep utility.



For proof-of-concept purposes, some samples signatures relating to 4 different IoT devices are also shown in Fig 3.4a. Overall, these signatures represent one of the first known attempts to capture notions of IoT maliciousness by scrutinizing empirical data. Also these signatures can be deployed on new incoming darknet traces to fingerprint newly-exploited IoT devices that have not yet been indexed by certain databases, e.g., such as Shodan and Censys. Moreover, these signatures can also be employed by local IoT realms to aid in mitigation, and thus remediation strategies.

Device Type	Signature
DVR	12288:wslGM8PFc6fXPWW4cVsBK0GT5gkLX15aurnz9k/Kk6:wslP8i6fPWW4cmBKrT5gkzlxrnz9+Kk6
Webcam	3072:6OA062aJtmzOTYfpTYJ7JaZgVx3BAaTZQzTwcht79+8R+TMWs9Zm2g0ivLJ1p/jR:rgFmQyTEJaQmzTwMl982g0YF1llYaJ
Printer	24576:XH9m8fEgLoZ7EqC0kf7tzH2uF/SD6dcZwEmGOqzH9m8fEgLf:4
Thermostat	6144:cMKa4Umz8VNPTg80mL4STGDs3+5FlwnVTF3gjGzTkpb/JkmBIDRkXY574OM231PL:0V8OfJow3glAJkmQ23l8eme/X4AMG6Bb\

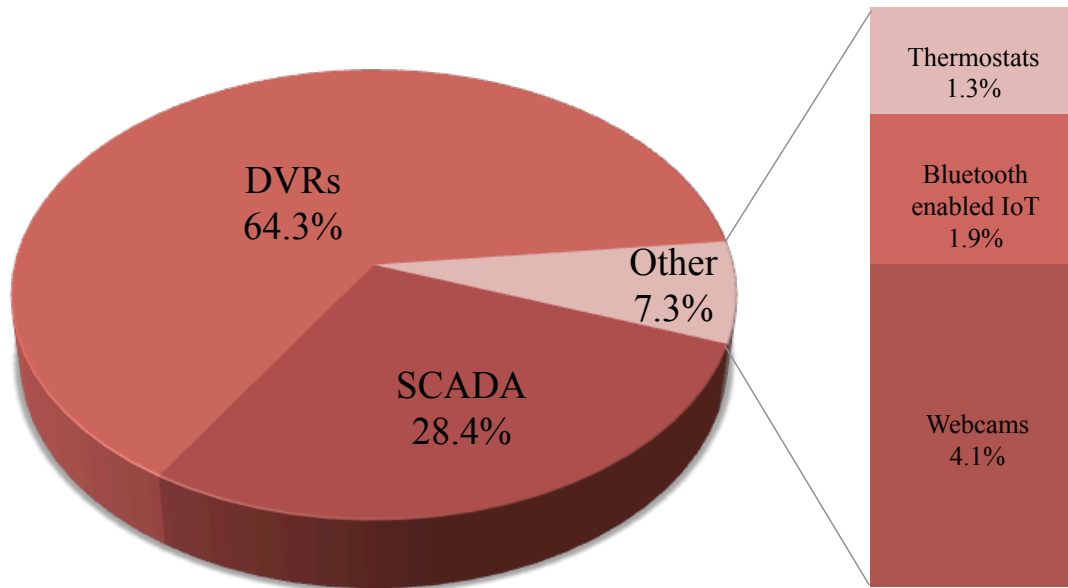
(a)



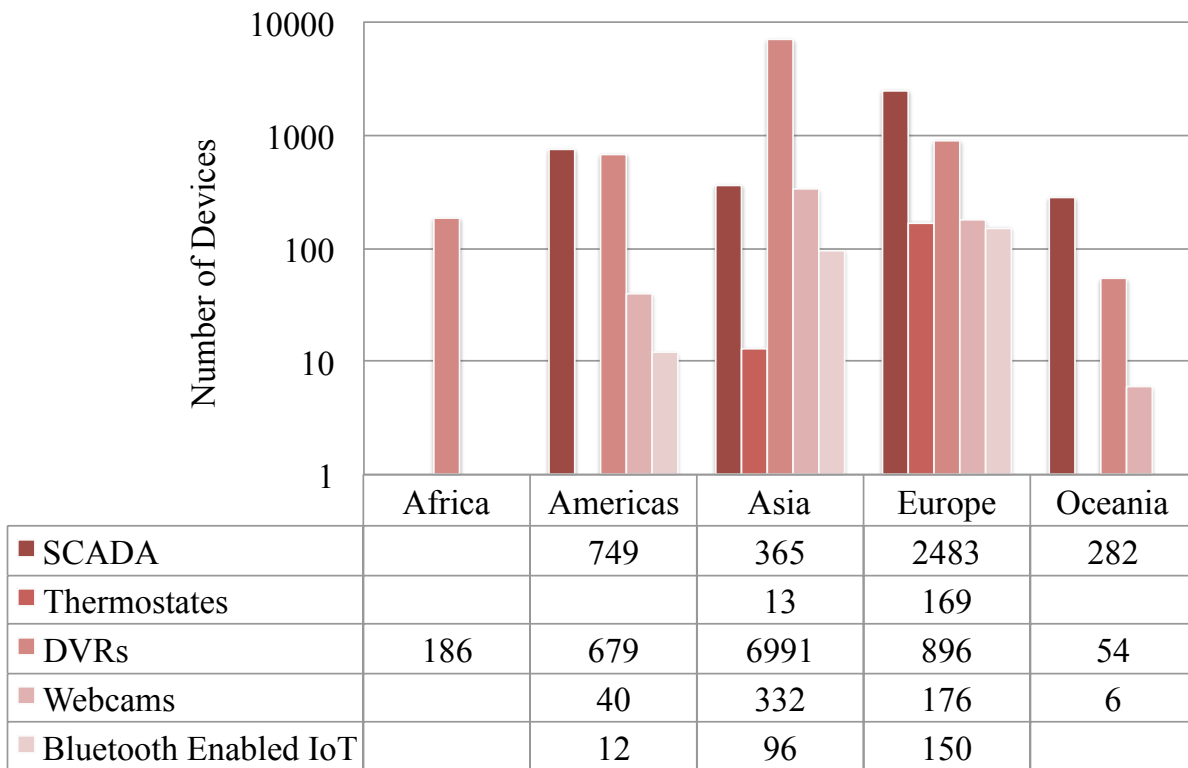
(b)

Copyright © 2018, IEEE

Figure 3.4: IoT devices: a) signatures and b) distributed by types



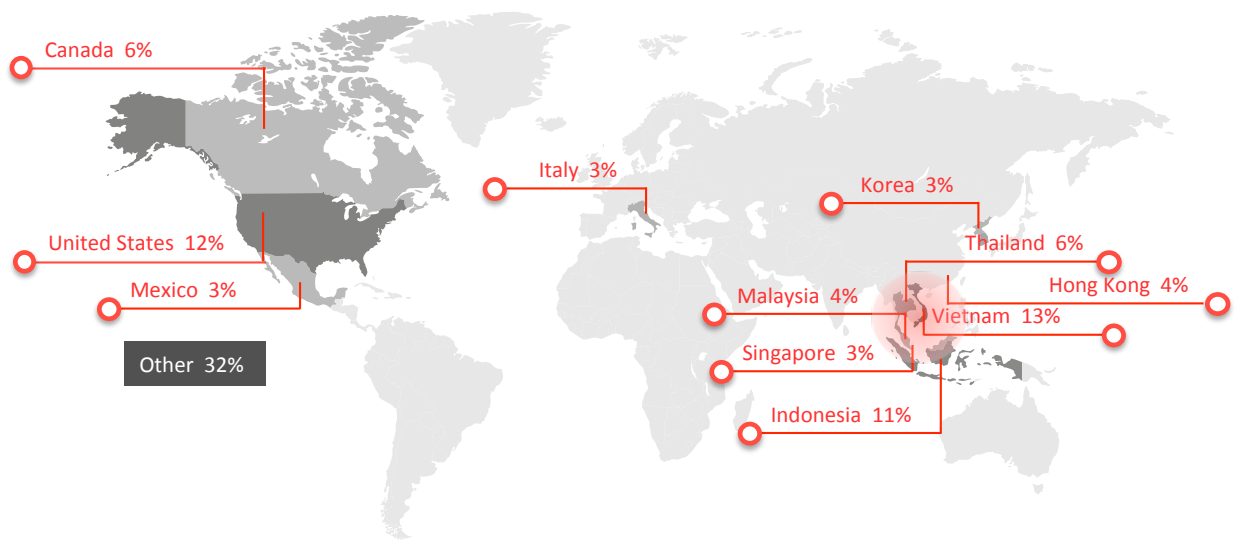
(a)



(b)

Copyright © 2018, IEEE

Figure 3.5: Exploited IoT devices: a) by types and b) by type and region



Copyright © 2018, IEEE

Figure 3.6: Global distribution of exploited IoT devices

## 4 Malicious IoT Device Classification using Machine Learning<sup>2</sup>

The use of network telescopes in conjunction with active measurements in Chapter 3 presents a solid foundation for empirical IoT threat analysis. Also, the proposed correlation algorithm along with its envisioned signature generation capabilities can help further infer and characterize large scale malicious IoT behavior. However, as noted in Chapter 2, signature-based schemes are not necessarily effective in identifying zero-day or new variants of attacks. There is also the added overhead of maintaining a database to store signatures for each IoT device in the network (and comparing them against darknet signatures). Furthermore, most attacker behaviors generally evolve in parallel with defense methodologies, as was observed in the case of Reaper, Hajime and Owari. Thus static signature generation techniques by themselves are insufficient for addressing the rich threat landscapes facing IoT devices.

Now, Chapter 2 also discussed the effectiveness of various ML and DL algorithms in solving practical cybersecurity problems across diverse domains, including IoT security. These learning algorithms present a new perspective for analyzing system vulnerabilities by identifying patterns and/or anomalies in data. These methods are particularly useful in the context of IoT security since related devices generate massive amounts of data (owing to the “always on” philosophy) which is essential for building robust and accurate models. Finally, as detailed in Chapter 3, Internet “sinks” (or blackholes) represent unused IP addresses with no devices associated with them. Therefore any traffic directed towards these IP addresses is not legitimate.

---

<sup>2</sup>Parts of this chapter were published in F. Shaikh et al. A Machine Learning Model for Unsolicited IoT Devices by Observing Network Telescopes. In *International Wireless Communications Mobile Computing Conference*, 2018. Permission included in Appendix B

In light of the above, a proper characterization of such traffic will also offer invaluable insights into attacker techniques and may also provide a stepping stone to build highly robust and flexible security models. Hence, this chapter presents a characterization and labelling taxonomy for unsolicited IoT devices found in the darknet, as detailed in [54]. Thereafter, multiple supervised ML algorithms are trained and tested on this data to identify the malicious activities performed by IoT devices. The high accuracy of ensemble models demonstrates the efficacy of the proposed approach in leveraging network telescopes and using ML schemes to classify malicious IoT activity. Overall, the proposed taxonomy schema (in conjunction with the ML models) will help further improve IoT cybersecurity situational awareness.

#### 4.1 Notation Overview

The requisite notation is presented first, see also Table 4.1. Foremost,  $header_{csv}$  is used to denote a csv file containing the header information for malicious IoT devices found in the darknet. Meanwhile,  $database_{sql}$  denotes a SQL implementation in Python that stores the same header information for faster processing. Additionally,  $feature_{csv}$  is another csv file containing the statistical features of IoT traffic from network telescopes. Similarly,  $feature$  is defined as a vector of explanatory variables, and  $label$  is the assigned label for each instance based on the taxonomy defined earlier.

The total number of trees used by the ensemble learners is denoted by  $T$ , and  $s_i$  and  $r_i$  represent the random subset of samples and the decision of a subset of predictors, respectively. Also,  $M$  denotes the number of stages in gradient boosting, and  $F_m(x)$  is the function updated at the  $m^{th}$  stage. Finally,  $weights$  is a vector of weights used to maintain a distribution over the dataset, and  $\alpha$  decides the step size of the Gradient Boosting algorithm, while  $\beta$  updates weights in the SAMME AdaBoost.

Table 2: List of variables

Variable	Description
$header_{csv}$	IP header information of unsolicited IoT devices in a csv file
$database_{sql}$	SQL database containing unsolicited IoT device information
$uniq_{ip}$	Unique IP in csv file containing header information
$feature_{csv}$	Features for training ML models in csv file
$feature$	Feature vector of samples for training ML model
$label$	Label vector of samples for training ML models
$Trees$	Total trees in a Random Forest algorithm
$s_t$	Subset of samples used by weak classifiers in Random Forest
$f_i$	Subset of features used by weak classifiers in Random Forest
$R$	Learned tree by Random Forest
$M$	Number of stages in Gradient Boosting & Samme AdaBoost
$L$	Loss function optimized in Gradient Boosting
$F_m(x)$	Function learnt at $m^{th}$ stage in Gradient Boosting
$\alpha$	Multiplier to decide step size in Gradient Boosting
$\beta$	Variable to update weights in SAMME AdaBoost
$C(x)$	Approximate Bayes classifier
$weights$	Vector of weights for SAMME AdaBoost

## 4.2 Exploiting Network Telescopes to Infer Malicious IoT Behavior

The lack of empirical threat data for IoT devices is a major roadblock in applying ML methods to IoT security problems, as noted in Section 3.3. This is due to the fact that ML methods must be trained on sufficient amounts of data in order to make reliable decisions. Although a few studies have attempted to address this concern by using simulation techniques, the resultant traffic patterns do not necessarily reflect true attacker behaviors seen on a global scale. However, the CAIDA repository monitors a large network telescope and makes the measurement archives available for research and analysis. This massive IBR repository contains various forms of illegitimate traffic, including but not limited to, backscatter from DDoS attacks, scanning attempts by unsolicited devices, misconfiguration traffic, etc., [55]. Specifically, CAIDA provides IP header information such as source/destination IP address,

source/destination port, protocol type, etc. Hence this network information can be used to derive key insights into attacker scanning and exploitation strategies.

Along those lines, akin to [48], the network characteristics of IoT devices are extracted from the CAIDA repository. The main motivation here is the fact that compromised IoT devices will likely form part of a bot army and will typically exhibit similar traffic characteristics as they scan the Internet for other vulnerable devices [2]. Now Chapter 3 has shown that network telescopes can help infer and characterize Internet-scale malicious IoT behaviors. Additionally, curating, processing and storing IP header information is computationally more efficient than payload-based analysis. Hence IP header information of IoT devices found in network telescopes is utilized for training the ML algorithms here. This information includes their source/destination IP addresses, source/destination port numbers, protocol numbers, etc. In general it is reasonable to expect infected devices to have similar ports and applications actively engaging in scanning or DDoS attacks.

### 4.3 Darknet Data Categorization

Once the darknet data for unsolicited IoT devices is extracted from CAIDA, it becomes imperative to categorize malicious IoT activities. Specifically, such a taxonomical scheme will enable security administrators to label their own traffic and build models to identify unsolicited device activity. However, darknet traffic is unique in a sense that there are no legitimate devices associated with the IP addresses. This means that packets sent to these addresses will not receive any replies. Hence, traditional classification-based mechanisms that rely on both source and destination IP addresses are not suitable in this situation.

To address the above challenge, various efforts have analyzed network telescope traffic and proposed data labeling schemes. Notably, [56] proposes a solution to detect compositional changes in darknet data. The authors also develop a tool called *iatmon* that groups one way traffic into two separate subnets based upon either the type of scanning being performed or



on inter-arrival time distributions of packets from each source IP addresses. Similarly, [57] proposes a classification scheme for detecting unsolicited darknet activity. Here, the authors outline session models based upon 3 protocols, i.e., Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP). Thereafter, the authors identify patterns in these darknet sessions to categorize malicious activities e.g., such as scanning, DoS, backscatter, etc. Finally, [58] and [59] also analyze network telescope data to identify patterns and trends in Internet-scale malicious activities. Namely, [58] proposes a taxonomy scheme that relies on traffic statistics for every source IP address seen in the darknet. Specifically, the authors identify various types of scanning and backscatter activities based on various metrics, e.g., such as the number of packets sent to a host, the number of ports targeted, etc.

Overall, the above studies reaffirm the importance of network telescopes as a rich source of Internet-scale malicious activity. However, these contributions do not account for IoT devices or the rise of botnet malware such as Mirai, Reaper, etc. Hence, this Chapter tries to address this concern by developing a taxonomy scheme for IoT IP addresses found in the darknet. The rationale behind this labelling methodology, as noted in [54], is now presented.

### 4.3.1 Scanning

Unsolicited IoT devices are constantly trying to infect other Internet devices with similar vulnerabilities. This is a common strategy used by malware such as Mirai to increase the size of its bot network and thereby orchestrate larger scale DDoS attacks. As noted in [58], darknet traffic patterns indicate the presence of various types of scanning activities. For example, in port scanning the attacker identifies active ports on a host by sending it requests packets. The motivation for this approach is to exploit a particular service or application using that specific port. Therefore, if a specific source IP address is sending such packet requests to multiple ports at unique darknet IP addresses, then it is identified as performing

a port scan. Note that in case of TCP traffic, the percentage of total packets sent by source IP addresses with either the SYN, FIN-ACK or NULL flags set must be greater than 50%.

Meanwhile, if an attacker targets the same port on multiple services, it is identified as a network scan. This scanning activity is primarily meant to exploit a specific host vulnerability. Again, this is apparent in the case of Mirai, which exploited open telnet ports on various IoT devices such as webcams, DVRs etc., (which were recruited as part of a bot network). Thus, if the presence of such activity is detected from a source IP address, then the device is classified as performing a network scan. Additionally, akin to the case of port scanning, the number of TCP packets with either the SYN, FIN-ACK or NULL flags set must be greater than 50%.

Finally, many attackers often perform “stealth” scanning to evade detection by IDS/IPS setups. In such an attack, a small number of TCP or UDP packets are sent to a small number of hosts in a given time period. These attacks yield low memory footprints as they target a small number of hosts, thereby evading detection. Thus, in the case of TCP, if a source IP address sends less than 15 SYN packets to less than 15 hosts targeting less than 5 ports in a given time frame, it is identified as performing a stealth scan.

### **4.3.2 Backscatter**

Backscatter packets observed in darknet data arise from a popular strategy used by attackers known as IP spoofing. This technique is mainly exploited to hide the true identity of the attacking device and involves spoofing the source IP address. IP spoofing is particularly useful in the context of DoS or DDoS attacks, i.e., since the reply packets are sent to fake IP addresses instead of the real attacker IP addresses. Now clearly there will be no response to these packets if the spoofed IP addresses reside in the darknet [54]. Therefore any such packets seen in the darknet are a clear indication of DDoS attacks and are labelled accordingly

in the taxonomy scheme. This includes TCP packets with either the SYN-ACK, RST-ACK or RST flags set.

Carefully note that any packet from an IoT device which does not meet either of the above 2 criterion is labelled as misconfiguration traffic [54]. Such traffic can occur due to various reasons, including incorrectly configured network address translation (NAT) rules, routing tables errors, etc. However, this traffic generally constitutes a very small portion of the overall total traffic and is treated as negligible here.

#### 4.4 Feature Extraction from Network Telescopes

As noted in Chapter 2, ML techniques analyze data and learn patterns to generate useful inferences. In general, these methods constitute a set of explanatory variables or features, i.e.,  $X = x_1 + x_2 + \dots + x_n$ , that are used to predict values in case of regression or probabilities in case of classification. Assuming that these outputs are termed as  $Y$  [60], the ML algorithm estimates a function  $f$  that satisfies the following expression:

$$Y = f(X) + \epsilon \quad (1)$$

where  $\epsilon$  is an irreducible error term with zero mean. Therefore the selection of explanatory variables that yield the most accurate prediction or inference is one of the most important goals of efficient ML algorithm design. Accordingly, [17] discusses the importance of selecting relevant features to improve the performance of supervised learning algorithms. Indeed, the presence of redundant or irrelevant features can have an adverse effect on the prediction accuracy of the algorithm. Furthermore, [17] also notes that the number of features is ultimately related to the number of samples, i.e., a larger number of features requires a larger number of samples to maintain the same accuracy. Therefore in many cases the number of samples needed can rise exponentially with every added feature.

Now previous work in [61] and [62] has explored the application of ML techniques for network traffic classification. For example, the authors in [61] discuss the impact of feature selection in correctly classifying network traffic. More specifically, they argue that statistical properties or flows are more efficient in distinguishing network traffic than simple port or payload-based mechanisms. Similarly, [62] highlights the importance of relevant feature selection for network traffic classification and also proposes a feature selection methodology based on “concept drift”.

Overall traffic statistics generated from IP flow data have yielded very good results for both supervised and unsupervised learning setups. Hence, a similar strategy is also adopted here where features are derived from network traffic statistics of IoT IP addresses found in the darknet. However, a key challenge here with regards to darknet traffic is that the traffic is one-way and hence these conventional methods cannot be used to derive the necessary features for ML. To address this concern, akin to [29] and [63], flows are now identified by only using the source IP addresses of IoT devices found in the darknet. Essentially, the important characteristics derived from the labelling methodology itself are used to derive the features here, as summarized in Table 3. A primary motivation for selecting these features is the fact that statistical values (such as the number of destination IP addresses and ports) are closely related to both scanning and DDoS activity. Another important feature to consider is also the total number of packets that each unsolicited IoT device sends to darknet IP addresses. This count provides a good indication of the scanning strategy of an attacker.

#### 4.5 Data Processing

The CAIDA measurement archives contain hour long compressed files of pcap data. Therefore as per the methodology utilized in Chapter 2, IP header information is extracted from this repository (along with relevant IoT device data) and placed in a CSV file. In addition to the data obtained and processed in Chapter 2, added data from entire days in

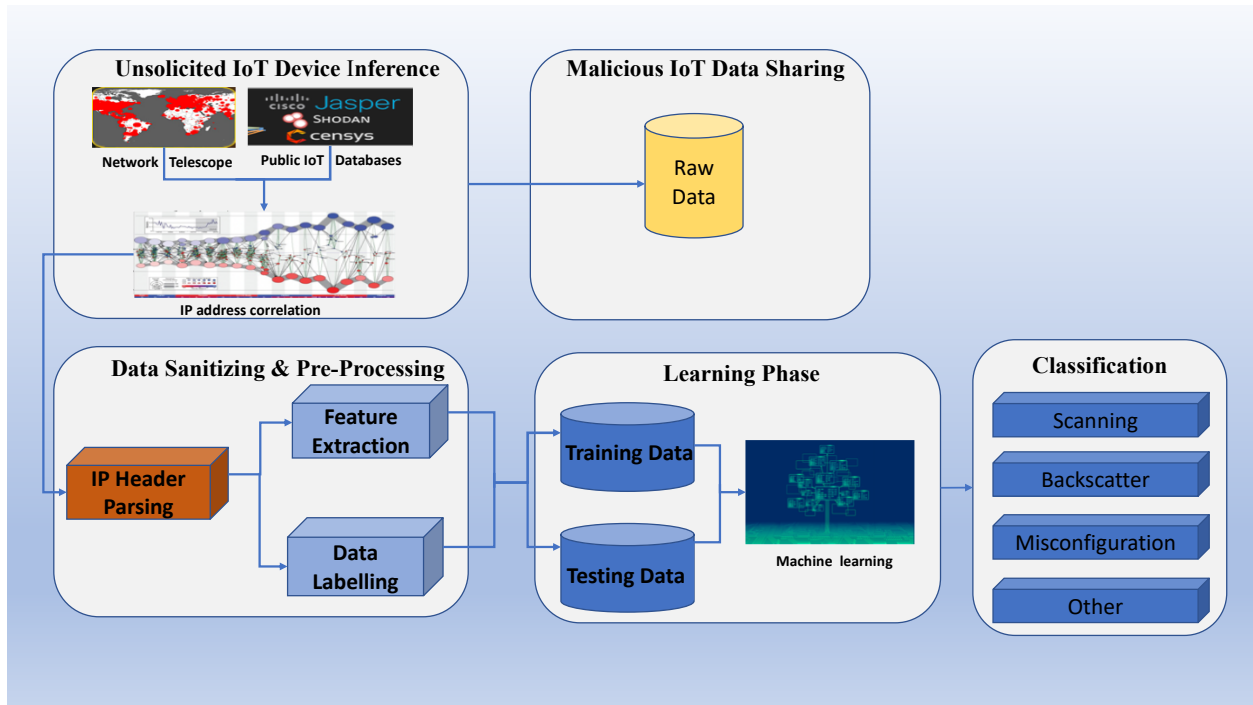
Table 3: Features for training ML algorithms

Feature	Description
avg ttl	Average ttl value for each src IP address
total packet count	Total # of packets sent by each src IP address
num of unique destination IPs	Total # of unique destination IP address targeted by each src IP address
avg of packets sent to port	Average # of packets sent to each dest port by each src IP address
avg packets sent to IP	Average # of packets sent to every dest IP by each src IP address
inter-arrival time	Average inter-arrival time of packets sent by each source IP
scan flags	# of packets sent with scan flags [SYN, FIN, FIN-ACK, NULL]
udp packets	# of UDP packets sent by each src IP address
number of unique dest ports	Total number of unique dest ports targeted by each src IP address

February and March 2017 is also extracted. This increased amount of information helps improve the prediction capabilities of the ML models, i.e., by comparing their performance on expanded datasets from different time periods.

A high level overview of the proposed malicious IoT activity classification methodology is show in Figure 4.1. This solution involves labelling the above data using the taxonomy proposed in Section 4.3. However the large size of the CSV files make it difficult to effectively retrieve the required statistics. Hence, the above issue is addressed by first converting the CSV files to a Structured Query Language (SQL) database, which significantly improves the time required to label the network telescope data. Namely, this database is implemented in Python and the labelled data is then stored in a CSV file. Furthermore, this database is also used to retrieve the features required for training the ML algorithms.

Overall, the use of hash tables (coupled with the database implementation) significantly reduces the time required to process the raw data retrieved from the CAIDA repository. The overall pseudocode of the data taxonomy and feature selection scheme is presented in Figure 4.2.



Copyright © 2018, IEEE

Figure 4.1: Proposed IoT device classification methodology

```

1: INPUT:  $header_{csv}$ 
2: OUTPUT:  $database_{sql}$ 
3:
4: for all  $row \in header_{csv}$  do
5:    $convert(database_{sql} \leftarrow row)$  ▷ Convert CSV file to SQL database
6:    $return(database_{sql})$ 
7: end for
8:  $feature_{csv} = \{\}$ 
9: for all  $uniq_{ip} \in header_{csv}$  do ▷ Extract features for each unique source IP
10:   $feature_{csv} \leftarrow (query\ SELECT\ feature\ for\ uniq_{ip}\ from\ data)$ 
11:   $return(feature_{csv})$  ▷ Return CSV file containing feature vectors
12: end for

```

Figure 4.2: Data labelling and feature selection algorithm

#### 4.6 Machine Learning Models

Overall, the strong pattern recognition capabilities of ML-based models can be used to derive unseen intelligence from data. This is clearly demonstrated by their highly successful

application in various domains. Now Section 2.3 has already presented some recent literature with regards to ML applications in IoT security, including the motivations behind using ML models for darknet data analysis. Therefore, in this Chapter, popular ensemble models such as Random Forest, Gradient Boosting and AdaBoost, which use decision trees as the weak learners, are leveraged to classify malicious IoT activity. Namely ensemble learners are ML algorithms that use multiple weak or base learners to solve either classification or regression problems. The outputs from these learners are aggregated by either using majority voting (classification) or weighted averaging (regression) [64]. Overall, the collective performance of an ensemble learner is generally better than that of the individual learners as shown in [65]. Note, that the performance of the individual weak learners themselves is only slightly better than random guessing, i.e., they are right about 50% of the time.

In general, decision trees represent a simple method to divide the predictor space into smaller sub-spaces. Although these trees are easy to interpret and have been modeled somewhat after human decision making processes, these algorithms by themselves perform poorly when compared to other supervised ML solutions [60]. Nevertheless, when coupled with techniques such as *boosting* or *bagging*, the performance of decision trees can improve considerably. Along these lines a brief description of these two techniques is provided here:

- Bagging: Averages the prediction of multiple *weak* learners trained on a random subset (with replacement) of input data. This method also offers the advantage of parallel operations since each model is trained independently.
- Boosting: Sequential operation to learn from residual errors of previous predictors. This method is more resilient to overfitting. However, the sequential nature of operations makes it difficult to parallelize this scheme for runtime efficiency.

As can be seen in Figure 4.3, for the case of boosting, a random subset of samples  $S = S_1, S_2, \dots, S_n$  is selected to train each weak learner. The misclassified samples are then

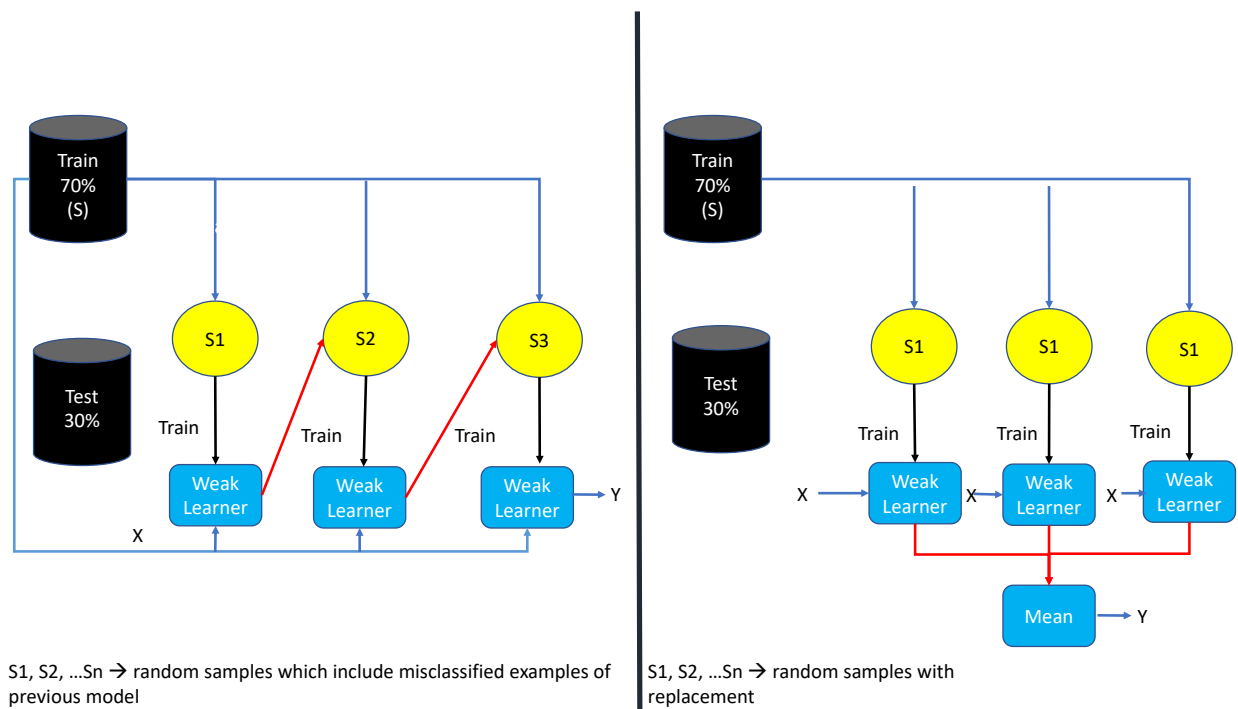


Figure 4.3: Boosting vs Bagging

included in the sub-samples used by the next model, i.e., the algorithm focuses on samples it has misclassified to iteratively improve its performance.

On the other hand, bagging uses a random subset of samples (with replacement) to train each weak learner and then simply computes the mean of the output of each model to give a final prediction, see Figure 4.3 as well.

In light of the above, the Random Forest, AdaBoost and Gradient Boosting schemes are trained on the data (which is labelled using the methodology detailed in Section 3.1). Note that these ensemble methods have also been successfully applied in previous literature to perform classification of non-malicious network traffic [66], as well as for detecting intrusion activities by using them as part of an IDS [67]. Consider some further details.



```

1: INPUT:  $feature = (x_1, x_2, x_3, \dots, x_n), label = (y_1, y_2, y_3, \dots, y_n)$ , number of trees =  $T$ 
2: OUTPUT:  $LearnedtreeR$ 
3: Function RandomForest{feature, label}
4:  $R = \{\}$ 
5: for all  $t \in Trees$  do
6:    $s_t \leftarrow$  random subset of sample
7:    $r_t \leftarrow$  LearnTree( $s_t$ , feature) ▷ Learn tree for bagged samples
8:    $R \leftarrow r_t$ 
9: end for
10:  $return R$ 
11: Function LearnTree{feature, label}
12: At each split:
13:  $f_i \leftarrow$  random subset of feature ▷ Improve variance by random selection of features
14: split on subset of features
15:  $r_i \leftarrow$  LearnTree( $f_i$ , feature)
16: return  $r_i$ 

```

Figure 4.4: Random Forest algorithm

#### 4.6.1 Random Forest

The Random Forest approach improves upon the bagging methodology by selecting a subset of learners at each tree split, see pseudocode in Figure 4.4. For this approach, all learners are used on the bootstrapped samples to make a decision [60]. However, as noted earlier, this can lead to overfitting especially if the trees are highly correlated due to the dominance of one or more strong predictors. Accordingly, the Random Forest scheme addresses this issue by only selecting a subset of models at each split instance. This strategy (also known as bootstrap aggregating or bagging) ultimately helps reduce variance, since each subset of predictors has its own variance, which is then averaged out in the final output. The prediction accuracy is further improved by selecting only a subset of features at each split to further reduce the variance of the model.

```

1: INPUT:  $feature = (x_1, x_2, x_3, \dots, x_n), label = (y_1, y_2, y_3, \dots, y_n)$ , stages =  $S$ , loss function =  $L(label, label_{estimated})$ , weak learner =  $w(feature, param)$ , multiplier =  $\alpha$ 
2: OUTPUT: Function optimized over parameter space:  $F_x$ 
3: Function Gradient Boost{feature, label}
4:  $F_0 = \sum_{i=1}^n L(y_i, \alpha)$ 
5: for  $i = 1, 2, \dots, S$  do
6:   for  $k = 1, 2, \dots, n$  do ▷ Perform gradient descent optimization
7:      $\hat{y}_k \leftarrow$  -(partial derivative of  $L$  w.r.t Predictor  $F$  for  $k^{th}$  sample)
8:   end for
9:   update model parameters
10:  update  $\alpha$ 
11:   $F_m(x) = F_{m-1}(x) + \alpha.w(feature, param)$  ▷ Update sequentially
12: end for
13: return  $F_m(x)$ 

```

Figure 4.5: Gradient Boosting algorithm

#### 4.6.2 Gradient Boosting

The original Gradient Boosting machine algorithm proposed in [68] uses a stage-wise additive process to estimate the best predictor/function (in the parameter space) and performs numerical optimization using the steepest descent algorithm. This approach differs from the Random Forest scheme in one very important way, i.e., Gradient Boosting relies upon a sequential operation that learns from past weak learners, whereas Random Forest averages out the prediction of the subset of weak learners. Hence this approach allows the scheme to avoid the problem of overfitting, i.e., low variance model. However, as compared to the Random Forest approach, the number of tunable parameters in the Gradient Boosting scheme is also higher. The pseudocode for the general implementation of this algorithm is also presented in Figure 4.5.

```

1: INPUT:  $feature = (x_1, x_2, x_3, \dots, x_n), label = (y_1, y_2, y_3, \dots, y_n), stages = S, error, weights = (w_1, w_2, \dots, w_n), w(feature, param), \beta$ 
2: OUTPUT: Approximate Bayes Classifier:  $C_x$ 
3: for  $i = 1, 2, \dots, n$  do ▷ Normalize weights
4:    $w_i = 1/n$ 
5: end for
6: for  $i = 1, 2, \dots, S$  do ▷ For every epoch
7:   Fit weak classifier  $F(x)$  to training data using weights
8:    $err^k = \sum_{i=1}^n w_i \cdot Pr(c_i \neq w(feature, param))$ 
9:   Update  $\beta$  ▷ Update  $\beta$  based on classifier error
10:  Update & normalize  $w$ 
11: end for
12:  $C(x) = \operatorname{argmax}_j \sum_{i=1}^S \beta^i \cdot Pr(F_x^{(i)} j)$ 
13: return  $C(x)$ 

```

Figure 4.6: SAMME AdaBoost algorithm

### 4.6.3 AdaBoost

Another variation of boosting known as AdaBoost is also implemented to classify IoT darknet data. AdaBoost is very similar to Gradient Boosting in the sense that both solutions try to iteratively improve the performance of a single weak learner. However, whereas Gradient Boosting uses residual errors to perform optimization, AdaBoost modifies a set of weights or distribution associated with the input instances [69] to train the weak learners. Hence the objective is now defined as reducing the error over this distribution rather than performing optimization over some residual function space. Along these lines, the AdaBoost implementation in [70] for multi-class classification is used here, i.e., termed as the Stagewise Additive Modeling using a Multi-class Exponential loss function (SAMME). Accordingly, the pseudocode for this algorithm is also presented in Figure 4.6.

Finally as per [54], a Naive Bayes scheme is also chosen for comparison purposes. This algorithm fits a Gaussian distribution over the input data by assuming feature independence. However, this assumptions may not hold in practice, especially in light of ever-mutating

Internet traffic patterns. Despite these concerns, the Naive Bayes scheme can handle complex inputs and its overall tractability makes it an ideal choice for a number of ML problem applications [71].

## 4.7 Performance Evaluation

The above schemes are implemented using the Python scikit library, which includes multiple supervised and unsupervised ML modules. In addition to this, diverse loss functions and accuracy metrics are also provided to test the performance of the models. Overall, these modules aid in rapid prototyping and testing of a wide array of ML models. Empirical findings from various datasets are now presented, some of which are also discussed in [54]. Namely, the IP addresses of about 3 million IoT devices are extracted from various databases including Censys, Shodan, etc. As noted in Chapter 3, these IoT devices belong to different system categories such as SCADA, webcams, thermostats, DVRs, etc. A database of IP addresses is then created and compared to the source IP addresses collected by the CAIDA network telescope for a one hour period on the first day of January 2017. Indeed, the vast size of this dataset makes it easier to train and test the proposed algorithms.

Overall, a total of slightly over 250,000 malicious IoT instances are found on the first day of January 2017 via the CAIDA network telescope. As noted in Section 4.5, the IP header information extracted here is also stored in a database for easier processing. The features described in Section 4.4 are also extracted and labelled according to the pseudocode algorithm in Figure 4.2. Based on this, a total of 208,647 IoT IP addresses are classified as performing scanning, 10,628 as sending backscatter traffic and 72,794 IPs are labelled as misconfiguration traffic. Now as noted in Section 4.6, the Random Forest, Gradient Boosting, Ada Boost and Naive Bayes schemes are used for classifying darknet data. Specifically, 70% of the data is used for training and the remaining 30% is used for testing. This partitioning is done using a random split function in the scikit library and a 5-fold cross-validation is

also done on the dataset. Also, precision and recall metrics are used to gauge the efficiency of the proposed schemes. Namely, precision is defined (in ML literature) as the percentage of a particular class  $C$  that are truly classified as class  $C$  among all instances which are classified as class  $C$ . Mathematically, precision can be defined as the ratio  $\frac{tp}{tp+fp}$ , where  $tp$  is the number of true positives and  $fp$  is the number of false positives. On the other hand recall is defined as the percentage of members of a class  $C$  that are correctly classified as belonging to class  $C$ . Mathematically, this value can also be represented as the ratio  $\frac{tp}{tp+fn}$  where  $fn$  stands is the number of false negatives.

Furthermore, two variants of the AdaBoost algorithm [70] are also implemented, i.e., depending on what parameter the algorithm uses to adapt in every iteration. Specifically, AdaBoost1 adapts based upon predicted class label errors, whereas AdaBoost2 adapts based upon class probabilities [72]. For the Random Forest scheme, the tree depths are also varied between 2 and 3, and these variants are denoted as Random Forest1 and Random Forest 2, respectively. Overall, findings show, that higher depth values do not provide significant improvements in performance. Finally, the Naive Bayes algorithm is run with both normalized and non-normalized data, and results show that the latter gives higher accuracy and recall scores. Hence only results for the non-normalized data are presented here. The overall results for all schemes are presented in Table 4.

To further validate the efficacy of the proposed ML models, tests are also done using data from different time periods. This approach helps avoid the problem of overfitting or high variance that might arise due to a large number of testing samples being drawn from the same time period. This strategy also helps assess how attack behavior changes over time. Accordingly, IP header information from IoT devices found in the CAIDA dataset on the first days of February and March 2017 is extracted. Based upon the detailed labeling methodology, it is determined that over 1.7 million IoT devices are performing scanning activities and over 100,000 devices are sending backscatter traffic. Furthermore a meagre

7,764 are classified as generating misconfiguration traffic in February 2017. Furthermore, in March 2017, the number of devices performing scanning drops to 1.2 million and the number sending backscatter drops to 80,000, see Figure 4.8. Note the parameters used here are the same as those used for the January 2017 dataset. For training purposes, the same methodologies proposed earlier are re-used and the detailed findings for these 2 months are presented in Tables 5 and 6

Overall, as can be seen in Figure 4.7a, the Gradient Boosting scheme outperforms all other algorithms, both in terms of accuracy and precision scores. This approach is followed closely by the Random Forest scheme, and finally the AdaBoost and Naive Bayes methods (with Naive Bayes performing the worst). The low performance of Naive Bayes method can be attributed to its independence assumption between all features, which is clearly not the case for network traffic. From Figures 4.7b and 4.8a, it is also seen that for the February and March 2017 datasets, the performance (recall and precision scores) only deteriorates slightly with the Gradient Boosting and Random Forest schemes, whereas it drops significantly for the Naive Bayes method. Overall, the very small drop in performance with Gradient Boosting and Random Forest indicates that these methodologies are quite effective for classifying malicious IoT traffic, i.e., the algorithms seem to have avoided overfitting.

The reduced performance of the Random Forest scheme can also be attributed to the fact that it is a high variance scheme and thus prone to overfitting. Although this algorithm does not perform as well as the Gradient Boosting approach, in many practical applications it is preferred due to its parallelized nature. Thus, practical constraints such as the large volume of data and requirement of real-time classification can result in the Random Forest scheme being selected over the Gradient Boosting scheme. However, the latter excels at handling outliers, selecting important variables and even performs well with missing data. Hence the trade off, as is often the case with any ML algorithm selection, lies between computation cost and prediction accuracy [72].

Table 4: Recall and precision values for the January 2017 dataset

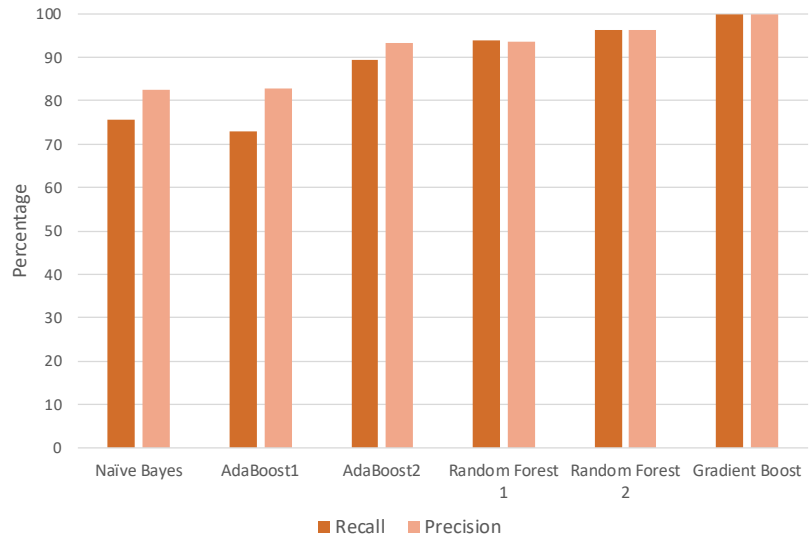
Metric	NaiveBayes	AdaBoost1	AdaBoost2	Random Forest1	Random Forest2	Gradient Boost
Recall	75.63	72.99	89.32	93.93	96.19	99.88
Precision	82.43	82.81	93.24	93.72	96.41	99.88

Table 5: Recall and precision values for the February 2017 dataset

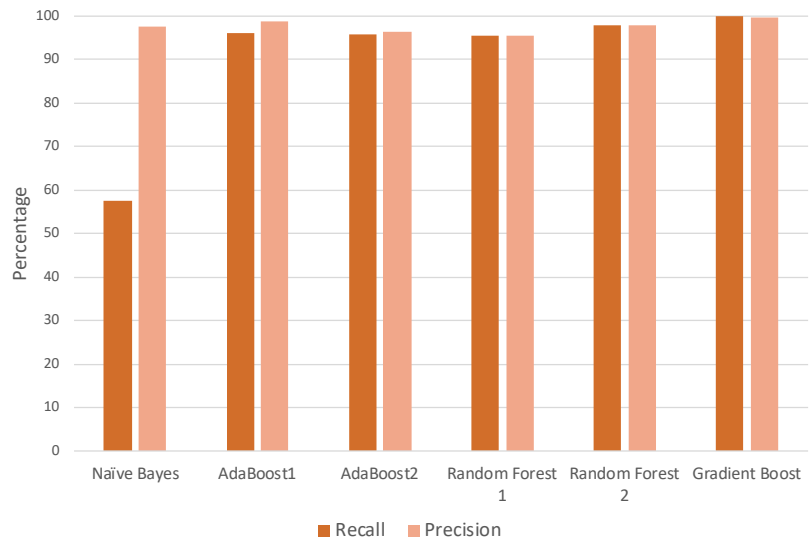
Metric	NaiveBayes	AdaBoost1	AdaBoost2	Random Forest1	Random Forest2	Gradient Boost
Recall	57.53	96.16	95.74	95.63	98.04	99.97
Precision	97.56	98.74	96.40	95.55	97.77	99.61

Table 6: Recall and precision values for the March 2017 dataset

Metric	NaiveBayes	AdaBoost1	AdaBoost2	Random Forest1	Random Forest2	Gradient Boost
Recall	60.14	97.17	96.60	93.32	98.33	99.90
Precision	95.54	99.01	95.59	93.67	96.72	99.43



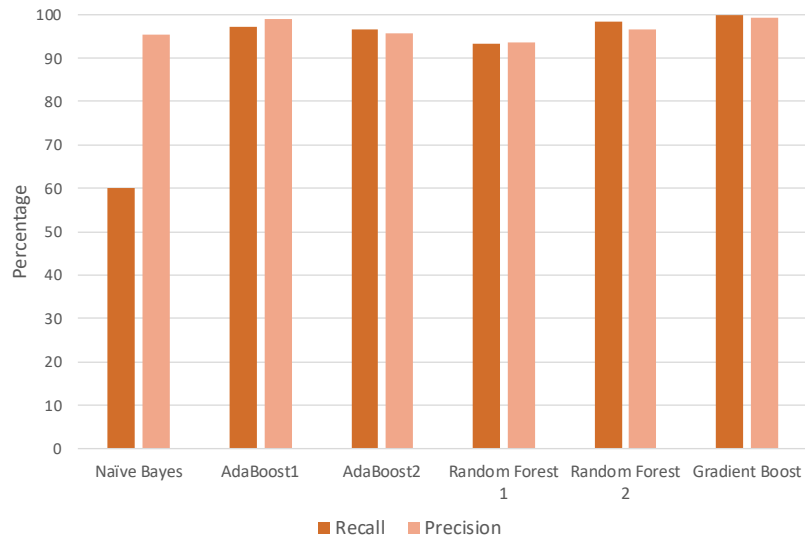
(a)



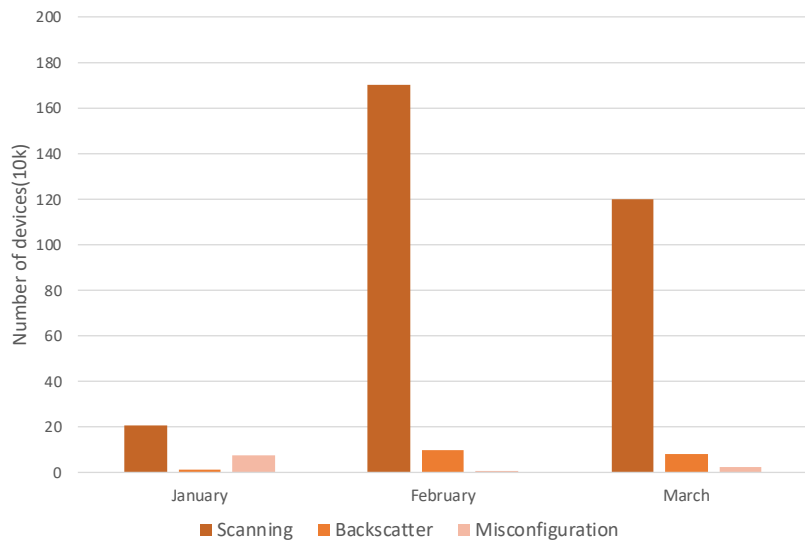
(b)

Figure 4.7: Recall and precision scores for: a) January 2017 and b) February 2017





(a)



(b)

Figure 4.8: a) Recall and precision scores for March 2017 and b) categorization based on activity

## 5 IoT Threat Detection Leveraging Network Statistics and GAN

The IoT device characterization methodology presented in Chapter 3 offers some novel insights into Internet-scale malicious behavior and can help in updating security defenses based on the type of activity being performed. Furthermore, the ML framework and unique taxonomy developed in Chapter 4 can help extract additional intelligence for identifying threats to and from IoT devices and is also highly effective in distinguishing between the various malicious behaviors of IoT devices, i.e., scanning, backscatter etc. However, as noted in Chapter 2, attacker behaviors are continually evolving from simple brute force techniques to more sophisticated methodologies. In fact, adversaries are now developing remote code exploits to target specific services or applications on a device. Indeed, the rise of Mirai-based malware variants with more advanced code bases and complex attack strategies is a clear indication of such new and emerging cyber threats.

In light of the above, traditional security mechanisms that rely on a database of malicious signatures or static methods to identify behavioral anomalies will prove insufficient. To address this concern, this Chapter presents a novel GAN-based framework that is trained to identify known and zero-day threats against IoT devices from both inside and outside the network perimeter. A use case for NFV is also presented that demonstrates how to apply this new networking paradigm to aid in IoT threat mitigation. In particular, the proposed solution leverages GANs (which excel at generating latent representations of high-dimensional complex datasets) to create a *normal* profile for IoT devices. Specifically, a GAN is trained to distinguish between normal and anomalous network traffic for 3 IoT devices in an experimental network lab testbed. Furthermore, to validate the efficacy of this solution, data from additional IoT devices (released by other researchers) is also utilized.

This is used to overcome potential overfitting issues which may occur due to the limited sample size of 3 IoT devices. Finally, well-established firewall and IDS software setups are also used to effectively mitigate threats from IoT devices once they have been identified by the GAN-based solution.

## 5.1 Experimental Setup

The proposed experimental network setup consists of 3 IoT devices that have been empirically shown to be vulnerable to popular malware (such as Mirai). A holistic overview of this setup is presented in Figure 5.1.

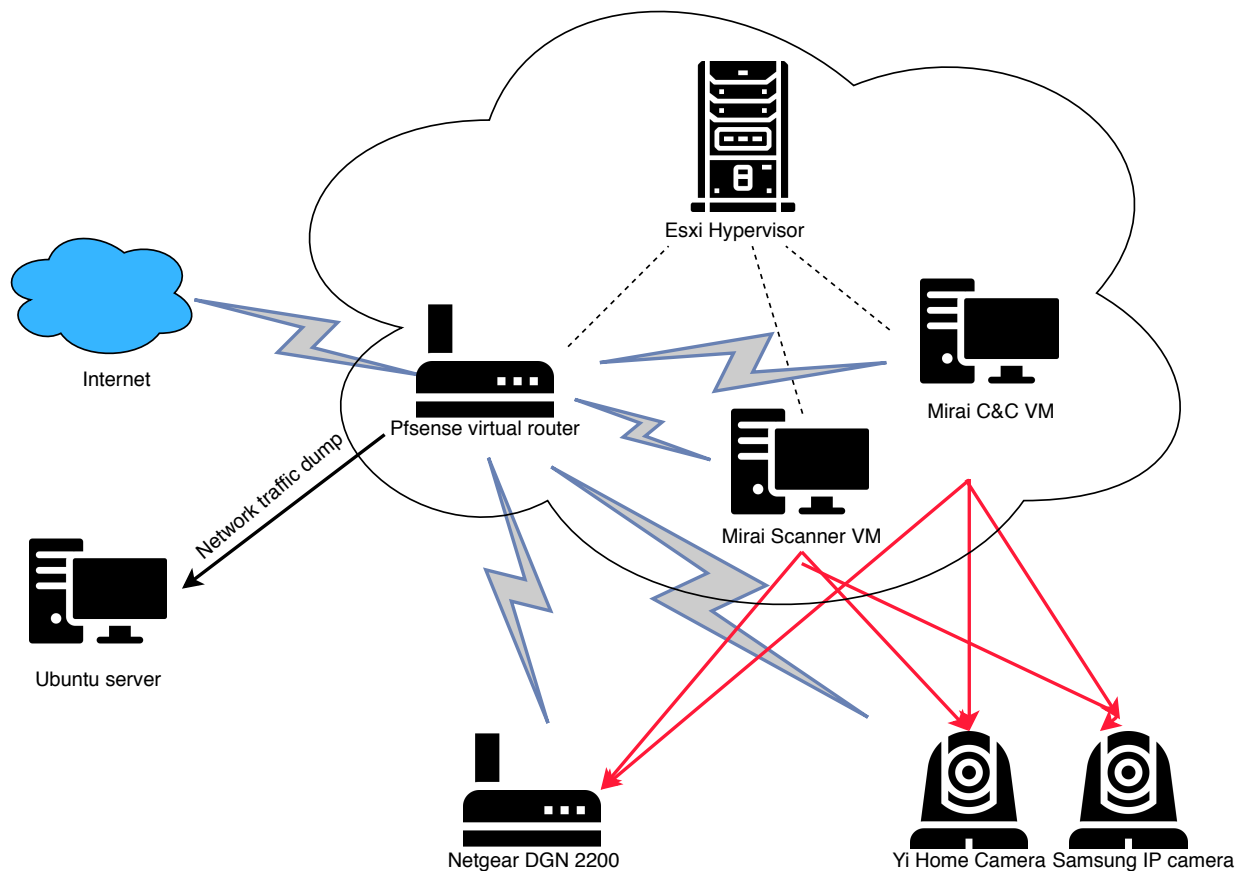


Figure 5.1: Holistic overview perspective of the experimental testbed attack setup

Overall, webcams and DVRs have seen a surge in popularity owing to their low cost and ease of use. Furthermore, many device manufacturers are also offering cloud-based services for storing and processing data collected by various sensors. A similar trend has also been observed in the Wifi access points/routers market, where manufacturers are providing users with very fine-grained bandwidth control capabilities in addition to various firewall and security features. However, these devices also present very lucrative targets for malicious actors since most users do not change their default username and passwords. As such, this makes it very easy for bad actors to use brute-force techniques (by exploiting manufacturer-specific credentials) on a large number of devices and recruit them as part of a bot network. Indeed, analyzing the Mirai malware reveals a list of default usernames and passwords that have been used to gain successful administrative privileges on IoT devices. Thereafter, depending upon the underlying architecture, binaries can be downloaded to these devices to communicate with the malware C&C setup and launch DDoS attacks or recruit other devices. In light of the above, several representative systems are used to build the experimental testbed of compromised IoT devices. In particular, the Netgear DGN 2200 router, Samsung IP camera and the Yi Home Camera 1080p are chosen to demonstrate the effectiveness of the proposed GAN frameworks, see Figure 5.1.

Now, the Netgear router has known remote code execution vulnerabilities which can be exploited to gain administrative privileges on the device. Essentially, these exploits include the HTTP POST command that can exploit the UPnP feature to allow cross-site forgery attacks. As noted earlier, bot attacker behaviors are evolving constantly and no longer rely on simple brute force techniques. Hence, to further test the efficiency of the proposed solutions, the aforementioned vulnerabilities are also exploited here.

Additionally, the Metasploit offensive security module (on the Kali Linux operating system) is also utilized to carry out additional attacks on the IoT devices. Namely, this version of Linux is a Debian-based operating system that contains numerous exploitation

tools/applications that are often used by professional penetration testers to identify weaknesses in a service, device, or network [73]. In particular the Metasploit framework in Kali Linux provides a number of exploit payloads as well as auxiliary and post exploitation modules for multiple architectures and operating systems.

Also, the experimental testbed setup uses the VMware ESXI hypervisor to help support VM instances. This virtualization software provides a simplified user interface for deploying multiple virtual machines that can run on most operating systems (with multiple use cases in traditional networks as well as the cloud). Now, a key advantage of the ESXI solution is its seamless separation of the guest operating system from the underlying hardware. This hypervisor also provides centralized management support along with performance improvements (as well robust storage and backup capabilities). As a result, multiple VMs can be established to host the C&C center for the Mirai and Bashlite malware. Furthermore, another VM instance is also created to run the Apache web server. Therefore once the attacker has gained administrative privileges on the IoT device, it can redirect it to this web server to download the relevant malware binaries.

Finally, the pfsense tool is also used as a virtual firewall and router to handle all networking tasks. This software offers increased flexibility and scalability, as well as a centralized management capability, i.e., key enablers of the NFV paradigm. Additionally, pfsense can also be extended to support packages such as Snort, OpenVPN, etc., through its package management system. Finally, this tool also provides a centralized view of all resources within a network along with a web graphical user interface for easy implementation of access control and firewall policies. This capability is particularly useful as when a device is infected and it needs to be isolated from the rest of the network (until appropriate remediation).

## 5.2 Attack Methodology

Overall, multiple attack strategies are adopted in the experiment to ensure that a sufficient breadth of malicious activities are covered. Along these lines, Table 8 presents a brief summary of all the attacks that are conducted. As noted in Section 5.1, the Metasploit module in Kali Linux is used to launch slightly more discrete attacks on the network, i.e., in addition to deploying the Mirai and Bashlite malware. Now the source code for both the Mirai and Bashlite malware contains a list of default usernames and passwords for brute-forcing. Hence, once a vulnerable device has been identified, the malware launches a dictionary attack using the aforementioned list to gain administrative access. Note that Mirai also has a list of IP address ranges that it is directed to avoid, e.g., including addresses belonging to the US Department of Defense (DoD), US Postal Service, as well as IP ranges belonging to internal networks. Since IoT devices are hosted in an internal network in the experimental setup of Figure 5.1, the relevant lines in the Mirai source code are appropriately edited. Now once infected, the application binaries are downloaded from the loader server. Thereafter, the infected device attempts to scan and recruit other vulnerable devices in the network. Finally, the adversary can launch a targeted DDoS attack using all devices as part of its bot network. Accordingly, a brief overview of Mirai's attack strategy is also shown in Figure 5.2. By contrast, Bashlite has a much simpler code base and hence the client-server model is used here instead of the original Internet Relay Chat (IRC) based version. Note that the 3 IoT devices and the attacker VMs are located in different sub-networks in order to emulate real-world attack scenarios.

In addition, the Nessus module in Kali Linux is also used here. Specifically, this module is capable of launching a variety of web attacks, e.g., such as checking for default credentials and web application scanning. Meanwhile, the Nmap module is also used to perform a scan of the entire network. Overall, Nmap provides a lot of potentially useful information, including the operating system of a device, the underlying architecture it is running on, as

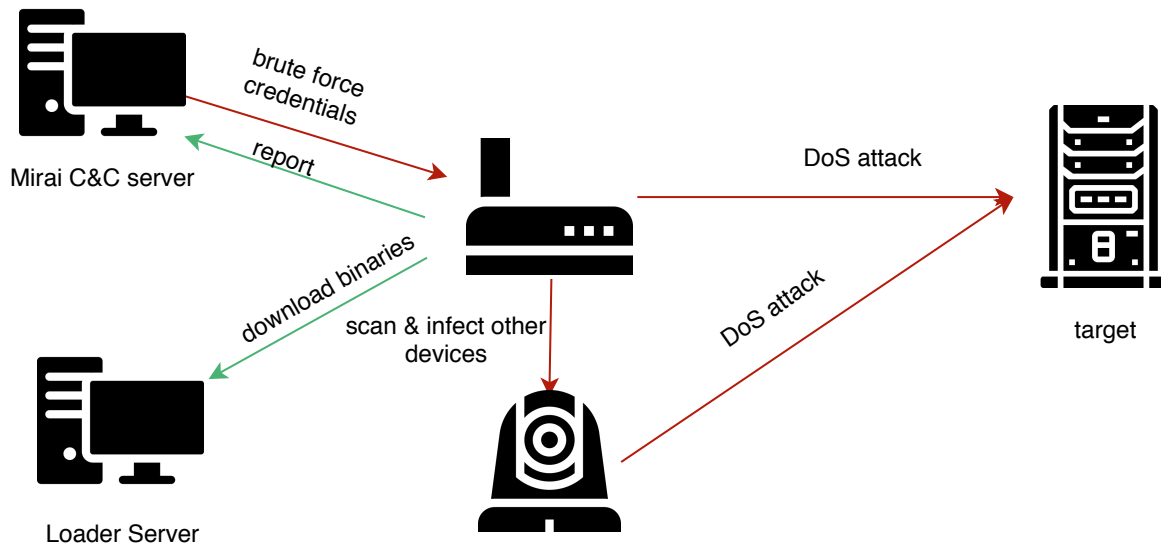


Figure 5.2: Mirai attack strategy

well as a list of all its open ports. Such a tool is especially useful during the reconnaissance phase of an attack, where the adversary tries to identify weaknesses in a system, ultimately enabling them to craft specific exploits to target a particular type of device.

As noted earlier, the Netgear DGN 2200 router used in the experimental testbed has multiple publicly-listed vulnerabilities, a brief description of which is also given in Table 7. Hence, the firmware of this router is appropriately downgraded to carry out the attacks noted earlier, i.e., since its latest firmware is patched and no longer vulnerable to the aforementioned attacks.

Table 7: Exploits against Netgear DGN 2200

Vulnerability	Description
Command Injection	Command injection using special post request
Remote Command Execution	Access to default user account
Cross-site request forgery	Unauthenticated remote code execution

Furthermore, in addition to the above-detailed attacks on the devices themselves, additional MITM attacks are also carried out on the 2 IoT cameras. These attacks are facilitated

by the fact that the cameras directly connect to the network over the WiFi network. Accordingly, the MITM attack is performed by using the address resolution protocol (ARP) caching technique. Namely, the adversary presents itself as a legitimate access point (AP), i.e., by broadcasting the MAC address of the legitimate gateway. This essentially deceives the devices into connecting to the rouge AP, thereby enabling the attacker to observe all communications with the real AP. In particular, a wireless adapter with packet injection capabilities is used to perform the MITM attack on the 2 IoT cameras.

Table 8: Attacks against IoT network

Attack	Netgear DGN2200	Yi camera 1080p	Samsung IP camera
Nmap scanning	Yes	Yes	Yes
Command injection	Yes	No	No
Nessus scanning	Yes	No	Yes
Remote code execution	Yes	No	No
Mirai	Yes	Yes	Yes
Bashlite	Yes	Yes	Yes
MITM	No	Yes	No

### 5.3 Overview of GAN

Deep generative models have recently become extremely popular due to their ability to learn underlying complex data patterns. Key examples of such methodologies include variational auto-encoders, GANs and autoregressive models. Overall, these models attempt to evaluate (explicitly or implicitly) the probability distribution of their input datasets. As a result, these DL models have found many applications in numerous and diverse areas such as image classification, natural language processing, etc [74]. Briefly consider the evolution of such generative models.

An early attempt to learn arbitrary dataset distributions was presented via the Boltzmann machine framework. This technique can be implemented using a multilayer perceptron (with maximum likelihood) in which case the learning rule becomes “local”, i.e., nodes update their



weights only based upon the output of their immediate neighbors, akin to human biology. Essentially this solution defines a joint probability distribution over the feature variables using an energy function which can be expressed as follows:

$$P(x) = \frac{e^{-f(x)}}{Z} \quad (2)$$

where  $P(x)$  is the joint probability distribution of the input variables,  $f(x)$  is the energy function, and  $Z$  is an intractable partition function that ensures that the sum of probabilities remains one. Similarly, the Restricted Boltzmann machine is another multilayer perceptron scheme which uses a single hidden layer and forms the basis for many DL models. The unique properties of this scheme yield conditional distributions which are simpler to compute, and the associated derivations are further detailed in [75].

As defined in [75] generative models, such as the Boltzmann machine, transform samples of latent (hidden) variables  $z$  (uniform distribution with zero mean and unit standard deviation) into data samples  $X$  (or a distribution over these samples using some differentiable function). Also, [75] notes that generating samples from more complex distributions requires the use of feedforward networks that represent a family of non-linear functions whose parameters are updated through training. The authors also note that two main approaches are used here, including a conditional distribution over the data or direct sampling. Overall, the key challenge here involves learning the generator mapping from  $Z$  to  $X$ .

Overall, GANs belong to the category of aforementioned models that are used to capture conditional distributions. Along these lines, [76] presents an adversarial-based framework for estimating a generative model. Essentially, this approach consists of two neural networks competing in a zero-sum game. Namely, the generator's distribution is first learned (over the data set) by mapping  $Z$  to the space manifold of  $X$  using a differentiable multilayer perceptron. Note, differentiability is an important condition for using the stochastic descent algorithm for backpropagation. Meanwhile, the second multilayer perceptron, termed as the

discriminator, tries to identify whether or not a given sample came from the actual data set or the mapped distribution. This overall scheme is shown in Figure 5.3 and the reader is referred to [76] for a more thorough treatment, i.e., including proofs of optimality and algorithm convergence.

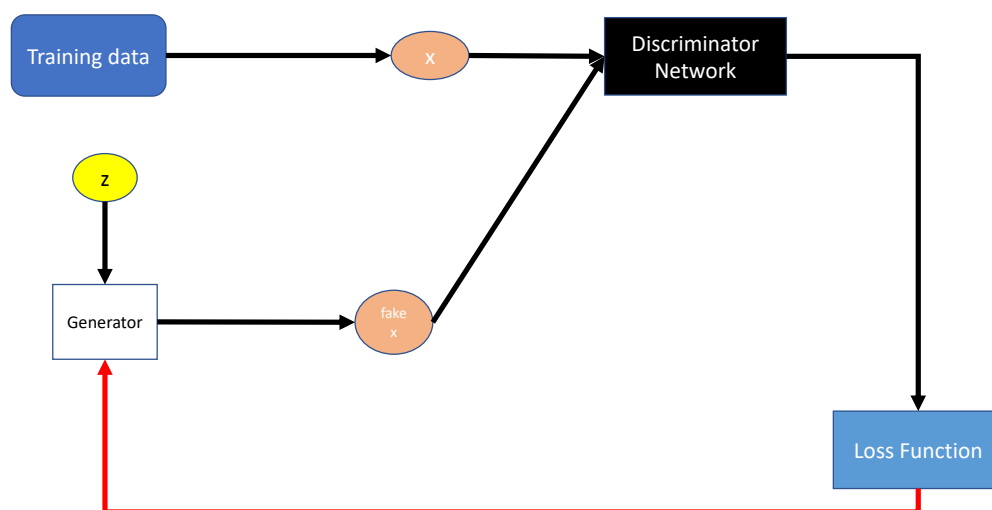


Figure 5.3: Overview of generative adversarial network (GAN)

Nevertheless, an important drawback with generative models (unlike variational auto-encoders) is that they cannot perform inference owing to the implicitly learned distribution. Hence, [77] attempts to solve the inference problem by requiring the discriminator to distinguish between the joint distribution over the data and latent space in a GAN framework. The joint learning over the two distributions enables the algorithm to identify whether a query image is from the same distribution as the data. This approach is also known as Adversarially Learned Inference (ALI) GAN and detailed further.

### 5.3.1 ALI GAN

The primary difference of the ALI GAN scheme versus the traditional GAN approach is that the generator now consist of two distinct components called the encoder and decoder. A zero sum game is then played between the discriminator and the generator, wherein the discriminator is now expected to identify data generated form both the encoder and the decoder. Accordingly, the value function for this game is defined as:

$$V(D, E, G) = \mathbb{E}_{q_x}[\log D(x, G_z(x))] + \mathbb{E}_{p_z}[1 - \log D(G_x(z), z)] \quad (3)$$

where  $q_x$  is the encoder data distribution,  $p_z$  is the latent distribution, and  $D$  and  $G$  are the generator and discriminator functions respectively. Once the encoder, decoder and discriminator networks have been learnt (trained), the model in [77] performs gradient descent on the discriminator network ,which is then propagated to both the encoder and decoder networks. As mentioned earlier, this approach differs from the traditional GAN in the sense that the discriminator attempts to learn a mapping from the latent space to the space manifold and vice-versa as shown in Figure 5.4.

In light of the above, the loss can now be represented as a convex multimodal function and is defined as follows:

$$L(x) = \alpha L_G(x) + (1 - \alpha)L_D(x) \quad (4)$$

where,  $L_G$  is the generator loss (which measures how similar a sample from the latent space is to the actual data) and  $L_D$  is the discriminator loss (generally cross-entropy which tries to improve the samples yielded by the generator). In particular  $L_D$  can also be defined as a loss based on a feature matching methodology where the features of an intermediate layer of the discriminator are used to compute the loss instead of the scalar output of the discriminator network [78].

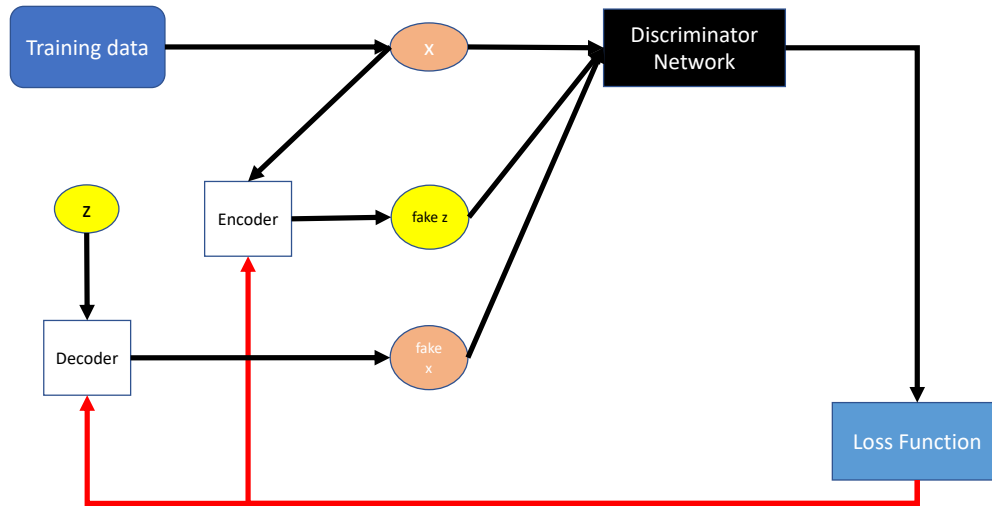


Figure 5.4: Overview of ALI GAN framework

Overall the proposed GAN-based solution adopts a similar strategy to [77], where the encoder, decoder strategy is used to train the discriminator. Namely, the generator produces samples from both latent and sample space distributions. The overall pseudocode for this procedure is shown in Figure 5.5.

### 5.3.2 AnoGAN

The GAN-based model presented in [78], termed as AnoGAN, is also implemented here. This model uses the feature matching methodology for computing the discriminator loss as mentioned earlier, although it does not use the decoder/encoder architecture mentioned in [77]. However, the authors do induce inference in the network by randomly sampling from the latent space ( $Z$ ) and passing it through the generator neural network ( $G(z)$ ). The resultant output is then compared with the query input ( $x$ ) and the difference is called the residual loss. Thus, the overall loss function in this case is defined as follows:

```

1: INPUT:  $feature = (x_1, x_2, x_3, \dots, x_n), label = (y_1, y_2, y_3, \dots, y_n)$ , Initialize network parameters
   for G, D, E
2: OUTPUT: Learned networks for G,D,E
3: while not converged do
4:    $q(x) \leftarrow x_1, x_2, \dots, x_n$ 
5:    $p(z) \leftarrow z_1, z_2, \dots, z_n$ 
6:   Sample  $z_i$  from  $q(z|x)$  ▷ Sample form data distribution
7:   Sample  $x_i$  from  $p(x|z)$  ▷ Sample form latent distribution
8:    $d1 \leftarrow$  probability match encoder ▷ Check discriminator output for encoder network
9:    $d2 \leftarrow$  probability match decoder ▷ Check discriminator output for decoder network
10:  Compute  $L_D$  ▷ Compute discriminator loss
11:  Computer  $L_G$  ▷ Compute generator loss
12:  Update parameters for G,D,E
13: end while

```

Figure 5.5: ALI GAN model for IoT threat detection

$$L(x) = (1 - \alpha)L_R(x) + \alpha L_D(x) \quad (5)$$

where,  $L_R$  is the residual loss (which measures how similar the generated data is to the query data) and  $L_D$  is the discriminator loss based on the feature matching methodology.

Overall, the ALI GAN architecture in [77] along with the aforementioned AnoGAN schemes are implemented here, akin to [62] (although the authors therein use the BiGAN architecture). A summary of the different GAN architectures implemented here is presented in Table 9. Overall, the neural networks used for the generator, discriminator and encoder are implemented using the Tensorflow library. Specifically, these networks are built using fully-connected or dense layers since non-image data is being processed. Namely, the encoder neural network consists of 3 layers with 128, 64 and 32 nodes/units. Similarly, the decoder network consists of two layers with 128 and 64 nodes each. The Adam optimizer is also used here for training the GANs, along with 500 iterations of gradient descent for AnoGAN scheme, akin to [78].

Table 9: Description of GAN models

Model	Description
AnoGAN_CE	AnoGAN with cross-entropy loss
AnoGAN_FM	AnoGAN with feature matching loss
ALIGAN_CE	ALI GAN with cross-entropy loss
ALIGAN_FM	ALI GAN with feature matching loss

#### 5.4 Data Processing and Feature Selection

Feature selection plays a vital role in the effectiveness of any DL model. Ideally, flow based statistics have performed well for network traffic anomaly detection, as noted in Chapter 2. Along those lines, pfsense is used to capture network traffic from all devices in the experimental test network in Figure 5.1. In particular, the feature selection methodology is applied for bidirectional flows over 4 windows that span the most recent 50, 100, 500, 1,000, and 2,000 packets. Overall, these packet windows provide varying resolutions of network activity and the driving intuition here is that malicious IoT devices will either attempt to scan, attack or infect another devices resulting in a change in network activity. Additionally, the mean and standard deviations of the number of packets, packet lengths, and packet inter-arrival times are also calculated. As a result, a total of 12 features are generated over a 5 window period yielding a total of 60 features which are indexed by their source IP addresses. A further description of these features is also presented in Table 10

Overall, 70,000 instances of benign data and 3,000 instances of malicious data are extracted from the experimental testbed. These samples include data from the 3 IP cameras and the Netgear router. The previously-detailed GAN models are also trained on this dataset and then tested using both the malicious and benign samples.

For comparison purposes, a different hypothesis is also evaluated here. Namely, the GAN network is also trained on the darknet data extracted in Chapter 3. As noted earlier, network telescopes (such as those at the CAIDA facility) provide passive measurements of Internet-

Table 10: Description of features selected for GAN learning

Feature	Description
Number of packets	Mean & std of total number of packets
Packet length	Mean & std of length of packets
Number of unique ports	Utilized ports
Packet inter arrival time	Mean & std of inter-arrival time
TCP	If using TCP as communication protocol, 0 if UDP
PSH flag	Number of packets with PSH flag set
URG flag	Number of packets with URG flag set
Idle time	Duration for which the connection was idle
Active	Duration for which the connection was active

scale unsolicited IoT behavior. However, the feature selection is done in a slightly different manner here. Specifically, the one-way features of the unsolicited IoT devices in the darknet data are extracted, as detailed in Chapter 4. Now a GAN trained on this darknet data should be capable of identifying anomalous IoT traffic as belonging to the same distribution as malicious IoT data found in the darknet. As a result any benign traffic generated by the IoT devices should be flagged as “anomalous”.

Furthermore, the network dataset from [19] is also analyzed here, which consists of network traffic data from 28 IoT devices. Namely, this study analyzes and identifies the statistical characteristics of multiple IoT devices. Now a key contribution of [19] is that the authors make a subset of the dataset publicly available for other researchers. Therefore, this data is used to train the proposed GAN model in addition to the benign data of the IoT devices detailed earlier. Overall, this diversified dataset allows one to better overcome potential overfitting concerns that might occur due to the limited number of devices in the initial setup and truly test the prediction powers of the model. Along these lines, close to 250,000 benign instances are extracted from the dataset in [19], in addition to the 70,000 instances already extracted earlier. However, the total number of malicious samples in this case remains unchanged at 3,000. Carefully note that even though two benign samples may

originate from different networks, the objective is still to capture the statistical characteristics (which are agnostic to specific variables like IP addresses, operating system type, etc.).

Finally, the performance of the Random Forest and the Gradient Boosting algorithms are also compared using the aforementioned augmented IoT dataset. As detailed in Chapter 4, these ensemble learners combine the outputs of multiple weak learners to provide highly accurate and low variance models. However, these supervised learning schemes required labelled data for training purposes, i.e., both benign and malicious samples are used to train the models.

## 5.5 Performance Evaluation

Detailed performance results for the various datasets are now presented. First of all, Figure 5.6a plots the overall precision and recall scores for all the schemes i.e., ALI GAN and AnoGAN as tested on the data collected from the 3 IoT devices. Here it is seen that the AnoGAN model using feature matching loss yields a precision score of 100% whereas the recall score is only 79.99% when applied to benign IoT samples from the experimental testbed network. These results indicate that the model is successfully able to distinguish between benign and malicious samples. However, the low recall score is a clear indication that the lack of samples restricts the learning effectiveness of the AnoGAN network, i.e., the network misclassified some benign samples as malicious. Similarly, the ALI GAN model using feature matching loss yields a precision score of 100% and a recall score of 84.56% when trained on the data of the 3 IoT devices in the experimental test network. However, the 100% precision scores for all the models might be an indication of overfitting as the number of samples provided to the models is very small.

Next, test results are presented using the datasets in [19]. Namely, the AnoGAN model trained on this augmented IoT dataset had a precision score of 99.30% and a recall score of 82.34%, as shown in Figure 5.6b. These results indicate that the AnoGAN model benefits



from the additional samples added to the distribution. A similar observation is also noted with the ALI GAN model, which gives a precision score of 100% and a recall score of 89.21%. Note that the performance of the ALI GAN model is considerably better than that of the AnoGAN model. These results indicate that the simultaneous learning of both latent and data distributions helps improve the inference power of the network.

Similarly, the performance of the GAN models trained on anomalous samples from the darknet are also presented in Figure 5.7a. Note, the precision scores in this case indicate how many relevant anomalous samples are selected, i.e., how efficient is the GAN in distinguishing anomalous and benign traffic. Meanwhile, the recall scores indicate how effective the GAN network is in learning the anomalous darknet data distribution. Once again, the ALI GAN model outperforms other GAN variants with precision and recall scores of 98.1% and 80.21% respectively. An important observation here is that changing the loss methodology from feature matching to cross-entropy does not result in any significant improvement in performance. In fact, both the ALI GAN and AnoGAN models have almost the same precision and recall scores for the two different loss methodologies.

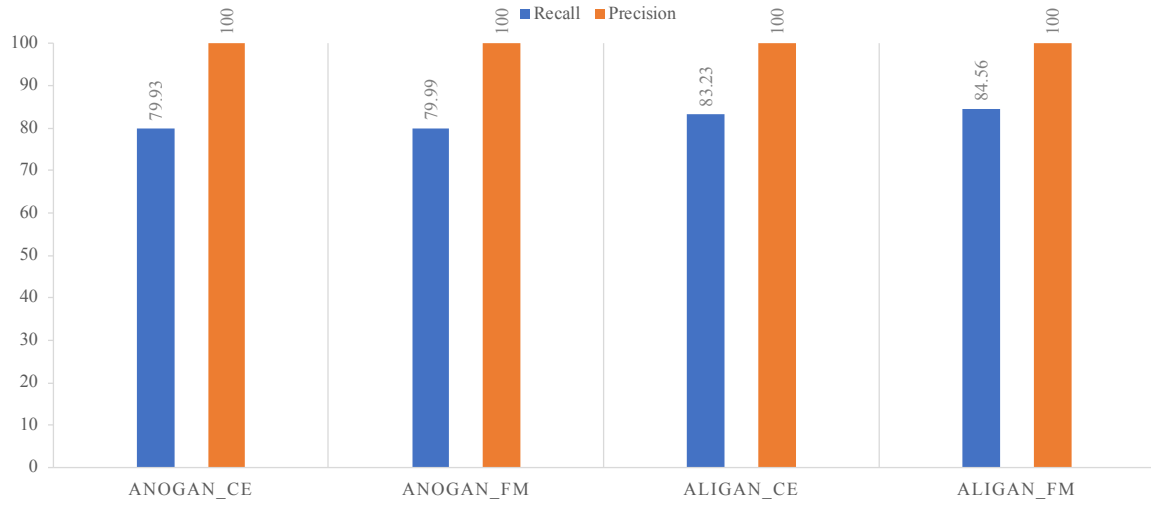
As noted earlier, the supervised learning algorithms used in Chapter 4, namely Random Forest and Gradient Boosting, are also evaluated here for comparison purposes. Overall, Random Forest gives a precision score of 97.29% and a recall score of 97.3%. The Gradient Boosting method also gives a precision score of 100% and a recall score of 98.90%. However, a major drawback with these algorithms is the fact that they require both benign and malicious data for training, whereas the GAN model only needs benign samples. This is a key distinction, since in local IoT realms, the GAN-based model would be much more preferable as it only requires data packets from devices directly connected to the actual network being safeguarded.

Finally, a comparison of the inference times of the various schemes is also presented in Figure 5.7b. In particular this is a measure of time it takes for the algorithm to predict

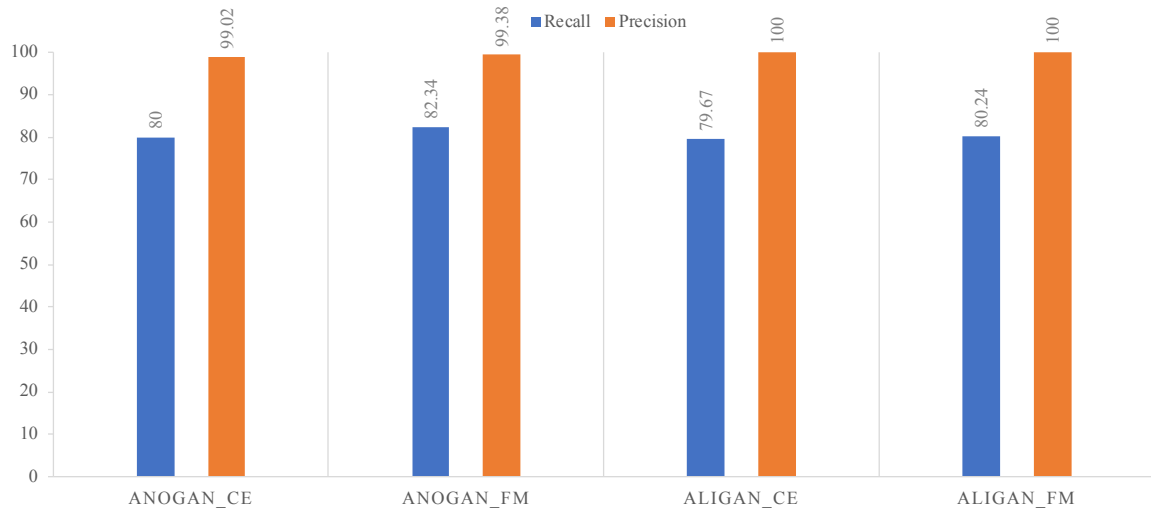
outputs for a given number of test samples. The overall findings indicate that the AnoGAN model takes the longest amount of time followed by the Gradient Boosting and Random Forest methods. Meanwhile, the ALI GAN network with both the encoder and the generator models yields the smallest inference time. In fact, the inference time is a magnitude of order difference compared to AnoGAN performance.

Note that once a device is identified as malicious or anomalous, an automated Python script is run to update an IP address list which is used by pfsense to block all communications from that device. This enables fast threat mitigation along with ease of automation and programming. As such, this approach mimics an SDN/NFV setup where NFV components and a centralized SDN philosophy can be used to simplify threat mitigation in complex IoT ecosystems. Furthermore, the aforementioned DL models can also be deployed as individual VM instances in various network segments, i.e., with each GAN model trained for the devices in that particular sub-network.

Overall, the results confirm that using a GAN model (to train the discriminator to distinguish between the latent and sample space) yields the best results. This solution also gives the smallest inference times. Although these findings are quite promising, the overall GAN-based methodology needs to be further evaluated using more diverse and expanded datasets to better gauge its effectiveness. Furthermore, such GAN models can also be applied to other layers of the communication network to help identify anomalous behaviors.

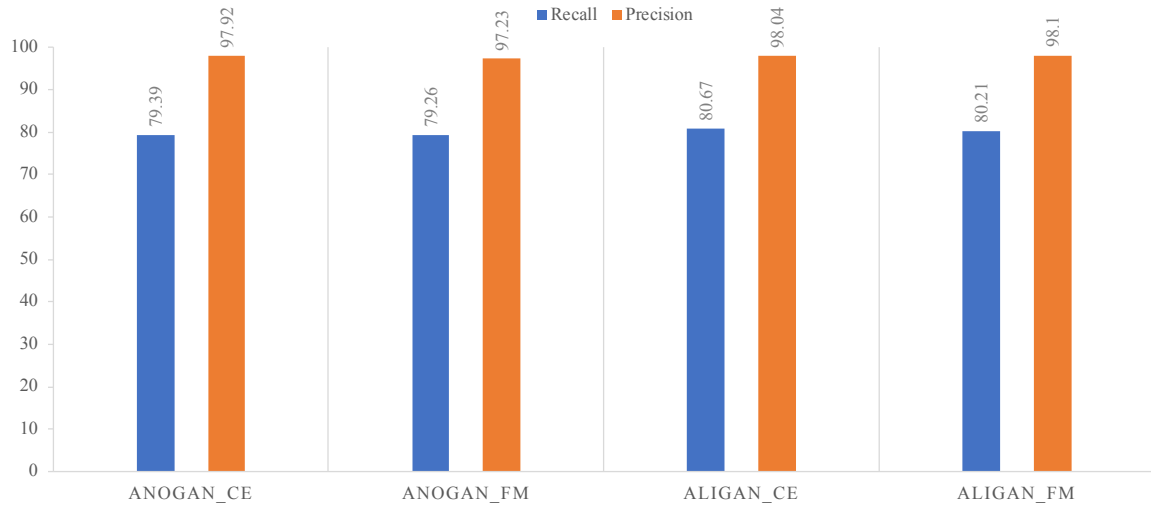


(a)

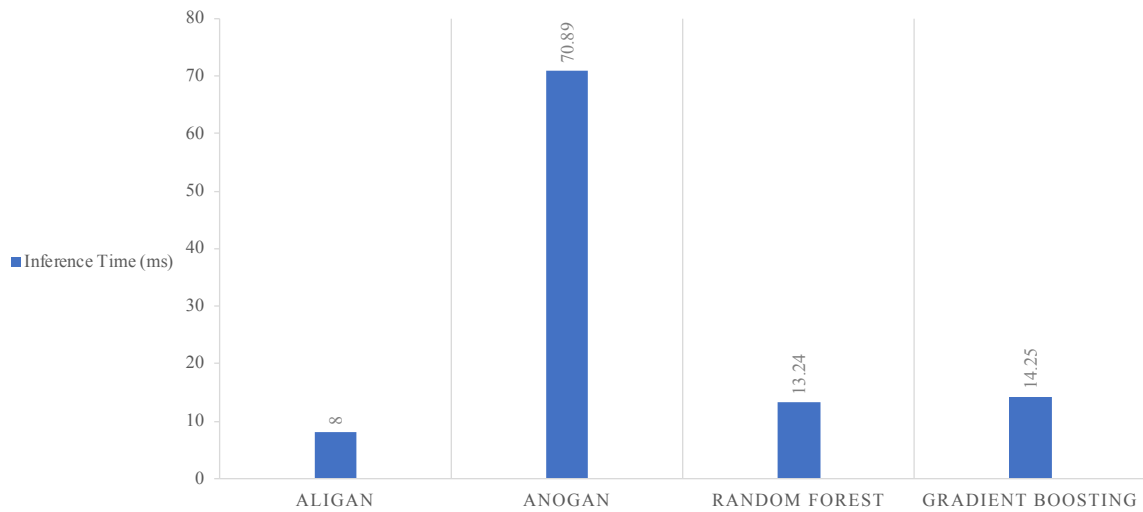


(b)

Figure 5.6: GAN performance on a) 3 IoT devices b) 31 IoT devices



(a)



(b)

Figure 5.7: a) GAN performance on darknet and b) inference time

## 6 Concluding Remarks

This research dissertation focuses on the IoT security problem and leverages empirical observations and network telescopes to develop defense frameworks to help identify threats to the IoT ecosystem. A brief summary of this dissertation is provided.

First, Chapter 2 begins by presenting a survey of the novel protocols and technologies being developed to accelerate the growth of IoT. This is followed by a discussion on the unique challenges associated with securing IoT devices. Finally, a review of ML and DL applications in the context of IoT security is also presented here. Thereafter, Chapter 3 proposes and evaluates a novel approach to infer, characterize and attribute unsolicited IoT devices by correlating active and passive measurements. In addition to this, IoT specific artifacts are also generated using the CPTH (hashing) algorithm. Subsequently, Chapter 4 proposes a unique taxonomy and labeling methodology for malicious IoT activities observed in network telescope data. Additionally, an ensemble learner framework is also proposed and evaluated to classify the malicious activities noted earlier. Finally, Chapter 5 proposes and evaluates a GAN-based framework that is capable of identifying known and zero-day threats from both internal and external sources to an IoT network.

Overall, the proposed methodologies present a novel data-driven approach (based on network traffic statistics) to defend against rising threats to IoT devices from malware such as Mirai and its variants. Indeed these exploits can potentially have debilitating effects on large scale Internet infrastructure and critical industries and also cause large financial loss.

## 6.1 Summary of Research Findings

This dissertation starts by addressing the lack of empirical threat data for unsolicited IoT devices due to various logistics and privacy concerns (noted in Chapter 2). Subsequently, Chapter 3 proposes a unique methodology for correlating active and passive measurements to identify the presence of malicious IoT data in network telescopes. This represents, to the best of the author's knowledge, a first-ever attempt to infer, characterize and attribute Internet-scale notions of malicious IoT behavior using darknet data. IoT-related signatures are also generated by using a unique hashing algorithm known as CPTH. Namely, this algorithm outputs a percentage of similarity between two files rather than a binary output generated by traditional hashes. These generated hashes (for malicious IoT devices) can then be used by IDS/IPS setups to further strengthen network defenses. Overall, the key observations from this chapter are summarized below:

- There are a large number of unsolicited IoT devices in the wild, and network telescopes or darknets provide a good empirical source to generate notions of maliciousness for such devices.
- DVRs and webcams are the most targeted IoT devices, comprising of more than 60% of the total discovered malicious devices. Meanwhile SCADA systems also represent a sizable fraction of compromised devices, i.e., totaling 28.4%. This observation is also validated by the fact that DVRs and webcams were the most common devices used in some recent DDoS attacks, as noted in Chapter 3.
- Generated signatures for each type of IoT device can be used to build effective security mechanisms that can quickly identify the presence of malicious IoT data in network traffic.

Chapter 4 extends the above work to classify the malicious activities of IoT devices observed in the darknet data by using ML techniques. This objective is accomplished by

proposing a novel taxonomy and labelling scheme that categorizes IoT activity as either scanning, backscatter, or misconfiguration. A number of ML algorithms, such as Random Forest, Gradient Boosting, Ada Boost and Naive Bayes are then trained to learn accurate representations of darknet IoT activity. Extensive testing is then conducted using data from different time periods to validate the efficacy of the proposed solutions. Overall, the results of the experiments indicate the following:

- Gradient Boosting yields the highest precision and recall scores, i.e., it can effectively identify whether a given device is infected (scanning), a victim of a DDoS attack (backscatter) or improperly configured. This observation is also validated by the highly successful applications of Gradient Boosting to other domains as well.
- Random Forest closely tracks the performance of Gradient Boosting in terms of accuracy metrics. However, the parallelizable nature of operations in Random Forest is very useful in situations which demand faster inference times.
- Ada Boost and Naive Bayes methods do not perform as well as the above two algorithms, with Naive Bayes being the worst at identifying malicious IoT activity.

Overall, the proposed schemes can help cybersecurity operators identify different types of malicious IoT activity directed toward or originating from IoT devices and adapt their defense mechanisms accordingly. For example, if a device is identified as performing scanning, it needs to be isolated from the rest of the network until the infection has been removed. Similarly, if a device is identified as sending backscatter packets, it is likely a victim of a DoS or DDoS attack and appropriate protection mechanisms can be implemented to mitigate the attack.

Finally, Chapter 5 presents a GAN-based framework which is further tested on real IoT traffic to identify known and zero-day threats. The main advantage of the approach is that it is trained by analyzing only benign IoT samples, i.e., no malicious data is required during

the training process. In particular, two different architectures tested here, i.e., ALI GAN and AnoGAN. These solutions vary in the design of the generator network and also how they perform inference. In addition to this, both cross-entropy and feature matching loss methodologies are also tested by using a variety of datasets. Overall the findings indicate:

- The ALI GAN architecture (with a feature matching methodology) performs the best on all 3 different datasets tested. These results clearly demonstrate the benefits of learning both the latent and the sample distributions to improve the inference power of the network.
- The feature matching methodology only gives a small improvement in overall accuracy metrics when trying to identify malicious IoT activity. In fact, in the case of the darknet dataset the performance using cross-entropy loss and feature matching loss is almost the same.
- The inference time of the ALI GAN architecture is significantly better than the AnoGAN framework and even beats the performance of ensemble frameworks (such as Gradient Boosting and Random Forest) when tested on the same sample size.

In conclusion, this dissertation proposes and analyzes multiple methodologies that leverage network telescopes and ML algorithms to identify threats in an IoT network. Most notably, network telescopes are used to characterize malicious IoT activity by correlating with active measurements to detect the presence of millions of IoT devices. This finding proves that network telescopes can offer hitherto unseen insights into large scale unsolicited IoT activity. These empirical observations are also used to train ML algorithms that can further identify whether or not a particular device is the victim or the perpetrator of an attack. Finally, Chapter 5 demonstrates how GAN-based frameworks can be used as anomaly identifiers without the need for malicious data samples. The proposed methodology is especially useful in the light of rising instances of zero-day threats.



## 6.2 Future Work

To the best of the author's knowledge, this work presents one of the first studies on leveraging network telescope data for building IoT security solutions. The proposed schemes also represent a solid foundation from which future research efforts in empirical IoT security can be planned and pursued. Namely, the feature selection methodology of the algorithms can be improved upon to capture the most relevant details for training algorithms. Research work can also be done in the area of GAN-based frameworks to capture benign notions across multiple layers of the communication model. Namely, instead of simply focusing on network statistics, one can also train GANs on a range of other data, e.g., such as application layer data, system data, etc. Future research efforts can also focus on training networks based on cross-layer features, i.e., multiple features are extracted from the different layer of the communication model to enable more accurate predictions. Finally, GAN networks trained on both benign and anomalous sample can be used collectively to identify anomalies. This framework is called *co-operative* GANs, in which the outputs of the two GAN networks can either be averaged in an ensemble-like framework or be combined by a bi-linear pooling operation to enable improved anomaly detection accuracy.

## References

- [1] N. Dragoni et al. “The Internet of Hackable Things”. In *arXiv:1707.08380 [cs]*, pages 129–140, 2018.
- [2] C. Koliadis et al. “DDoS in the IoT: Mirai and Other Botnets”. In *Computer*, volume 50, pages 80–84, 2017.
- [3] Z. Zorz. “Hajime IoT Worm Infects Devices to head off Mirai”, April 2017.
- [4] B. Kerbs. “Reaper: Calm Before the IoT Security Storm?”, October 2017.
- [5] T. Seals. “Wicked Botnet Uses Passel of Exploits to Target IoT”, May 2018.
- [6] S. Bandyopadhyay et al. “Internet of Things : Applications and Challenges in Technology and Standardization”. In *Wireless Personal Communication*, volume 58(1), pages 49–69, 2011.
- [7] L. Atzori et al. “The Internet of Things: A Survey”. *Computer Networks*, 54, 2010.
- [8] A. Whitmore et al. “The Internet of Things: A Survey of Topics and Trends”. *Information Systems Frontiers*, 17(2):261–274, 2015.
- [9] S.H. Shah and I. Yakoob. “A Survey: Internet of Things (IoT) Technologies, Applications and Challenges”. In *IEEE Smart Energy Grid Engineering (SEGE)*, ON, Canada, August 2016. IEEE.
- [10] K. Zhao and L. Ge. “A Survey on the Internet of Things Security”. In *Ninth International Conferene on Computational Intelligence and Security*, Leshan, China, December 2013. IEEE.

- [11] R. Mahmoud et al. “Internet of Things: Current Status, Challenges and Prospective Measures”. In *10th International Conference for Internet Technology and Secured Transactions*, London, UK, December 2015. IEEE.
- [12] C. Suchitra et al. “Internet of Things and Security Issues”. *International Journal of Computer Science and Mobile Computing*, 5(1):133–139, January 2016.
- [13] W. Zhou et al. “The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions and Challenges Yet to be Solved”. *IEEE Internet of Things Journal*(*Early Access*), pages 1–11, June 2018.
- [14] C. Bekara. “Security Issues and Challenges for IoT-based Smart Grid”. *Procedia Computer Science*, 34:532–537, 2014.
- [15] F. Dalipi et al. “Security and Privacy Considerations for IoT Applications on Smart Grids: Survey and Research Challenges”. In *4th International Conference on Future Internet of Things and Cloud Workshops*, Vienna, Austria, August 2016. IEEE.
- [16] J. Wurm et al. “Security Analysis on Consumer and Industrial IoT Devices”. In *21st Asia and South Pacific Design Automation Conference*, Macau, China, January 2016.
- [17] A.L. Buczack et al. “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection”. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2016.
- [18] M.A. Alsheikh et al. “Machine Learning in Wireless Sensor Networks: Algorithms, Strategies , and Applications”. *IEEE Communications Surveys & Tutorials*, 16(4), April 2014.

- [19] A. Sivanathan et al. “Characterizing and Classifying IoT traffic in Smart Cities and Campuses”. In *2017 IEEE Conference on Computer Communications Workshops*, pages 559–564, Atlanta, May 2017. IEEE.
- [20] L. Xiao et al. “Cloud-Based Malware Detection Game for Mobile Devices with Offloading”. *IEEE Transactions on Mobile Computing*, 16(10):2742–2750, October 2017.
- [21] A. Gupta et al. “Computational Intelligence based Intrusion Detection Systems for Wireless Communication and Pervasive Computing Networks”. In *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–7. IEEE, 2013.
- [22] L. Xiao et al. “IoT Security Techniques Based on Machine Learning”. *arXiv:1801.06275v1*, 2018.
- [23] Y. Meidan et al. “ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis”. In *Symposium on Applied Computing*, pages 506–509, Marrakech, Morocco, April 2017.
- [24] Y. Li and M. Chen. “Software-Defined Network Function Virtualization: A Survey”. *IEEE Access*, 3:2542–2553, December 2015.
- [25] D. Bekerman et al. “Unknown Malware Detection using Network Traffic Characteristics”. In *IEEE Conference on Communications and Network Security*, Florence, Italy, September 2015. IEEE.
- [26] S.T. Ikram and A.K. Cherukuri. “Intrusion Detection Model using Fusion of Chi-Square Feature Selection and multi class SVM”. *Journal of King Saud University - Computer and Information Sciences*, 29(4):462–472, October 2017.

- [27] B.J. Radford et al. “Network Traffic Anomaly Detection Using Recurrent Neural Networks”. *arXiv:1803.10769 [cs.CY]*, March 2018.
- [28] J. Vidal et al. “Adaptive Artificial Immune Networks for Mitigating DoS Flooding Attacks”. *Swarm and Evolutionary Computation*, 28:94–108, February 2018.
- [29] T. Ban. “3-3 Data mining applied to Darknet Traffic Analysis”. *Journal of National Institute of Information and Communications Technology*, 63(2):45–54, 2016.
- [30] D. Apiletti et al. “Characterizing Network Traffic by means of the NetMine Framework”. *Computer Networks*, 53(6):774–789, 2009.
- [31] R. Shahid et al. “SVELTE: Real-time intrusion detection in the Internet of Things”. *Ad Hoc Networks*, 11(8):2661–2674, November 2013.
- [32] S. Douglas H et al. “Ultra-lightweight Deep Packet Anomaly Detection for Internet of Things Devices”. In *34th International Performance Computing and Communications Conference*, Nanjing, China, December 2015. IEEE.
- [33] Y. Median et al. “Detection of Unauthorized Devices Using Machine Learning Techniques”. *arXiv:1709.04647v1*, 2017.
- [34] E. Hodo et al. “Threat Analysis of IoT Networks using Artificial Neural Network Intrusion Detection System”. In *International Symposium on Networks, Computers and Communications*. IEEE, May 2016.
- [35] J. Canedo and A. Skjellum. “Using Machine Learning to Secure IoT Systems”. In *14th Annual Conference on Privacy, Security and Trust*, Auckland, New Zealand, May 2016. IEEE.

- [36] A.A. Diro and N. Chilamkruit. “Distributed Attack Detection Scheme Using Deep Learning Approach for Internet of Things”. In *Future Generation Computer Systems*, volume 82, pages 761–768. Elsevier, May 2018.
- [37] L. Dhanabal and SP. Shantharajah. “A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms”. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6):446–452, June 2015.
- [38] M. Yair et al. “N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders”. *IEEE Pervasive Computing*, 13(3):12–22, September 2018.
- [39] L. Claas et al. “An SDN/NFV-Enabled Enterprise Network Architecture Offering Fine-Grained Security Policy Enforcement”. *IEEE Communications Magazine*, 55(3):217–223, March 2017.
- [40] L. Wei and C. Fung. “FlowRanger: A Request Prioritizing Algorithm for Controller DoS Attacks in Software Defined Networks”. In *IEEE International Conference on Communications*, London, UK, June 2015. IEEE.
- [41] T. Xu et al. “Defending Against New-Flow Attack in SDN-Based Internet of Things”. *IEEE Access*, 5:3431–3443, February 2017.
- [42] C. Shaibal et al. “Black SDN for the Internet of Things”. In *12th International Conference on Mobile Ad Hoc and Sensor Systems*, Dallas, October 2015. IEEE.
- [43] S. Pradip Kumar et al. “DistBlockNet: A Distributed Blockchains-Based Secure SDN Architecture for IoT Networks”. *IEEE Communications Magazine*, 55(9):78–85, September 2017.

- [44] F. Olivier et al. “SDN Based Architecture for IoT and Improvement of Security”. In *IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, Gwangju, South Korea, March 2015. IEEE.
- [45] K. Kalkan and S. Zeadally. “Securing Internet of Things with Software Defined Networking”. *IEEE Communications Magazine*, 56(9):186–192, September 2018.
- [46] I. Farris et al. “Towards provisioning of SDN/NFV-based Security Enablers for Integrated Protection of IoT Systems”. In *IEEE Conference on Standards for Communications and Networking*, Helsinki, Finland, September 2017. IEEE.
- [47] V. Sivaraman et al. “Network-level Security and Privacy Control for Smart-Home IoT Devices”. In *IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications*, Abu Dhabim UAE, October 2015.
- [48] F. Shaikh. “Internet of Malicious Things Correlating Active and Passive Measurements for Inferring and Characterizing Internet-scale Unsolicited IoT Devices”. *IEEE Communications Magazine*, 56(9):170–177, September 2018.
- [49] K. Claffy. “CAIDA:Visualizing the Internet”. *Internet Computing Online*, page 88, 2001.
- [50] Y. Tianlong et al. “Handlin a Trillion (unfixable) Flaws on a Billion devices: Rethinking Network Security for the Internet-of-Things”. In *14th ACM Workshop in Hot Topics in Networks*, Philadelphia, November 2015. ACM.
- [51] T. Andrew. “Spansum Readme”, 2002.
- [52] J. Kornblum. “Available from <http://ssdeep.sourceforge.net>”.
- [53] Brian Wallace. “Optimizing SSDEEP For Use At Scale”. *Virus BULLETIN: Covering the global threat landscape*, pages 1–9, 2015.

- [54] F. Shaikh et al. “A Machine Learning Model for Classifying Unsolicited IoT Devices by Observing Network Telescopes”. In *14th International Wireless Communication and Mobile Computing Conference*, Limassol, Cyprus, June 2018.
- [55] C. Fachkha and Debbabi M. “Darknet as a Source of Cyber Intelligence: Survey, Taxonomy, and Characterization”. *IEEE Communications Surveys & Tutorials*, 18(2):1197–1227, 2016.
- [56] N. Brownlee. “One-Way Traffic Monitoring using iatmon”. In *Passive and Active Network Measurement Workshop*, volume 7192, pages 179–188. Springer Berlin Heidelberg, March 2012.
- [57] J. Treurniet. “A Network Activity Classification Schema and Its Application to Scan Detection”. *IEEE/ACM Transactions on Networking*, 19(5):1396–1404, 2011.
- [58] J. Liu and K. Fukuda. “Towards a Taxonomy of Darknet Traffic”. In *International Wireless Communications and Mobile Computing Conference*, Nicosia, Cyprus, August 2014. IEEE.
- [59] E. Wustrow et al. “Internet Background Radiation Revisited”. In *10th Annual Conference on Internet Measurements*, page 62, 2010.
- [60] G. James et al. *An Introduction to Statistical Learning: With Applications in R*. 103. Springer, 2013.
- [61] T.T.T. Nguyes and G. Armitage. “A Survey of Techniques for Internet Traffic Classification using Machine Learning”. *IEEE Communications Surveys & Tutorials*, 10(4):56–76, 2008.
- [62] H. Zhang et al. “Feature Selection for Optimizing Traffic Classification”. *Computer Communications*, 35(2):1457–1471, July 2012.



- [63] N. Furutani et al. “Detection of DDoS Backscatter Based on Traffic Features of Darknet TCP Packets”. In *9th Asia Joint Conference on Information Security*, Wuhan, China, September 2014.
- [64] Z-H Zhou. “*Encyclopedia of Biometrics*”, chapter Ensemble Learning. Springer, 2009.
- [65] P. Salamon and L. Hansen. “Neural Network Ensemblers”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, October 1990.
- [66] J. Zhang et al. “Random Forest Based Network Intrusion Detection”. *IEEE Transactions on Systems, Man and Cybernetics*, 38(5):649–659, September 2008.
- [67] Farnaz Gharibian and A.A. Ghorbani. “Comparitive Study of Supervised Machine Learning Techniques for Intrusion Detection”. In *5th Annual Conference on Communication Networks and Services Research*, Fredericton, Canada, May 2007. IEEE.
- [68] J.H. Friedman. “Greedy Function Approximation: A Gradient Boosting Machine”. *The Annals of Statistics*, 2001.
- [69] Y. Freund and R.E. Schapire. “A Short Introduction to Boosting”. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.
- [70] T. Hastie et al. “Multi-class AdaBoost”. *Statistics and its Interface*, 2(3):349–360, 2009.
- [71] N.B. Amor et al. “Naive Bayes vs Decision Trees in Intrusion Detection Systems”. In *ACM Symposium on Applied Computing*, pages 420–424, Nicosia, Cyprus, March 2004. ACM.
- [72] T. Hastie et al. “*Elements of Statistical learning*”. Springer, 2 edition, 2009.
- [73] G. Singh and J. Singh. “Evaluation of Penetration Testing Tools of Kali Linux”. *International Journal of Innovations and Advancement in Computer Science*, 5(9):28–32, September 2016.

- [74] Y. Mirsky et al. “Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection”. In *Network and Distributed System Security*, San Diego, February 2018. arXiv.
- [75] I. Goodfellow et al. “*Deep Learning*”. MIT Press, 2016.
- [76] I. Goodfellow et al. “Generative Adversarial Nets”. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.
- [77] V. Dumoulin et al. “Adversarially Learned Inference”. In *International Conference on Learning Representations*, Toulon, France, April 2017. arXiv.
- [78] T. Schlegl et al. “Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery”. *Information Processing in Medical Imaging*, abs/1703.05921, August 2017.

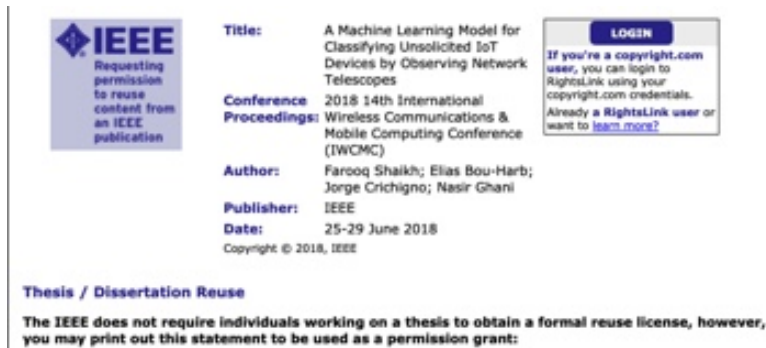
## Appendix A Proof of Copyright Permissions

The permission below is for the use of material in Chapter 3.



The screenshot shows the IEEE RightsLink interface. At the top left is the Copyright Clearance Center logo. The main header features the IEEE logo and the text "Requesting permission to reuse content from an IEEE publication". To the right, the "RightsLink" logo is displayed. Navigation buttons for "Home", "Create Account", "Help", and an email icon are visible. The central content area includes a "LOGIN" button and a text box: "If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already a RightsLink user or want to learn more?". The main text provides the following details:  
**Title:** Internet of Malicious Things: Correlating Active and Passive Measurements for Inferring and Characterizing Internet-Scale Unsolicited IoT Devices  
**Author:** Farooq Shaikh; Elias Bou-Harb; Natalia Neshenko; Andrea P. Wright; Nasir Ghani  
**Publication:** IEEE Communications Magazine  
**Publisher:** IEEE  
**Date:** Sept. 2018  
Copyright © 2018, IEEE  
Below this, a section titled "Thesis / Dissertation Reuse" states: "The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:"

The permission below is for the use of material in Chapter 4.



The screenshot shows the IEEE RightsLink interface for a different document. It includes the same IEEE logo and "Requesting permission to reuse content from an IEEE publication" text. The "RightsLink" logo and navigation buttons are also present. The central content area features a "LOGIN" button and the same text box: "If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already a RightsLink user or want to learn more?". The main text provides the following details:  
**Title:** A Machine Learning Model for Classifying Unsolicited IoT Devices by Observing Network Telescopes  
**Conference Proceedings:** 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)  
**Author:** Farooq Shaikh; Elias Bou-Harb; Jorge Crichigno; Nasir Ghani  
**Publisher:** IEEE  
**Date:** 25-29 June 2018  
Copyright © 2018, IEEE  
Below this, a section titled "Thesis / Dissertation Reuse" states: "The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:"

The permission below is for the use of material in Chapter 5 (Icons by Freepik & lineator from [www.flaticon.com](http://www.flaticon.com)).

## How must I insert the attribution?

### Related articles

When do I need to provide attribution to the author?

How can I use Flaticon's resources without attribution?

How many resources can I download?

Are Flaticon contents for free and where can I use them?

What are Flaticon's Premium resources?

In order to use an icon you must attribute it to its author, so we will be able to continue creating new graphic resources every day.

Icon made by [author link] from [www.flaticon.com](http://www.flaticon.com)

E.g.: Icon made by Freepik from [www.flaticon.com](http://www.flaticon.com)

### How to attribute it?

#### For websites:

Insert the attribution on the page of the icon (for example in the page footer) or on the imprint page.

#### For printing:

Paste this attribution on the final work so the authorship is known (for instance, in the acknowledgements chapter of a book)

#### For apps:

Place the attribution on the credits/description page of the application.

#### Social Networks:

If you desire to upload any of our resources to your social networks please remember you must insert the attribution "Designed by **Freepik** from [www.flaticon.com](http://www.flaticon.com)" on the uploaded image and on the description of the image.

## Appendix B Glossary

<i>ANN</i>	Artificial Neural Network
<i>AP</i>	Access Point
<i>ARP</i>	Address Resolution Protocol
<i>BACnet</i>	Building Automation and Control networks
<i>CAIDA</i>	Centre for Applied Internet Data Analysis
<i>C&amp;C</i>	Command and control
<i>CoAP</i>	Constrained Application Protocol
<i>CPU</i>	Central Processing Unit
<i>CRC</i>	Cyclic Redundancy Check
<i>CSV</i>	Comma Separated Value
<i>DDoS</i>	Distributed Denial of Service
<i>DL</i>	Deep Learning
<i>DNS</i>	Domain Name Service
<i>DoS</i>	Denial of Service
<i>DVR</i>	Digital Video Recorder
<i>FAR</i>	False Accuracy Rate
<i>FNV</i>	Fowler-Noll-Vo
<i>GAN</i>	Generative Adversarial Network
<i>GPU</i>	Graphics Processing Unit
<i>HTTP</i>	Hyper Text Transfer Protocol
<i>ICMP</i>	Internet Control Message Protocol
<i>IAAS</i>	Infrastructure-as-a-Service

<i>IBR</i>	Internet Background Radiation
<i>ICMP</i>	Internet Control Message Protocol
<i>IDS</i>	Intrusion Detection System
<i>IoT</i>	Internet of Things
<i>IPS</i>	Intrusion Prevention System
<i>IRC</i>	Internet Relay Chat
<i>ISP</i>	Internet Service Provider
<i>kNN</i>	k Nearest Neighbor
<i>LSTM</i>	Long Short Term Memory
<i>MAC</i>	Media Access Control
<i>ML</i>	Machine Learning
<i>MITM</i>	Man-in-the-middle
<i>MQTT</i>	Message Queuing Telemetry Transport
<i>NE</i>	Nash Equilibrium
<i>NFV</i>	Network Function Virtualization
<i>NAT</i>	Network Address Translation
<i>NFV</i>	Network Function Virtualization
<i>REST</i>	Representational State Transfer
<i>RFID</i>	Radio Frequency Identification
<i>RNN</i>	Recurrent Neural Network
<i>RPL</i>	Routing Protocol for Lossy networks
<i>SAAS</i>	Software-as-a-Service
<i>SCADA</i>	Supervisory Control and Data Acquisition
<i>SDN</i>	Software Defined Network
<i>SINR</i>	Signal-to-interference-plus-noise-ratio
<i>SVM</i>	Support Vector Machine

<i>TCP</i>	Transmission Control Protocol
<i>UDP</i>	User Datagram Protocol
<i>UPnP</i>	Universal Plug and Play
<i>WSN</i>	Wireless Sensor Network