IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 19, 2019, accepted May 7, 2019, date of publication May 14, 2019, date of current version May 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2916717

Forensics and Deep Learning Mechanisms for Botnets in Internet of Things: A Survey of Challenges and Solutions

NICKOLAOS KORONIOTIS, NOUR MOUSTAFA, AND ELENA SITNIKOVA
UNSW Canberra Cyber, the School of Engineering and Information Technology (SEIT), University of New South Wales Canberra, Canberra, ACT 2612, Australia

Corresponding author: Nickolaos Koroniotis (n.koroniotis@student.adfa.edu.au)

The author N. Koroniotis would like to thank the Commonwealth's support, which is provided to the aforementioned researcher in the form of an Australian Government Research Training Program Scholarship.

ABSTRACT The constant miniaturization of hardware and an increase in power efficiency, have made possible the integration of intelligence into ordinary devices. This trend of augmenting so-called non-intelligent everyday devices with computational capabilities has led to the emergence of the Internet of Things (IoT) domain. With a wide variety of applications, such as home automation, smart grids/cities, and critical infrastructure management, the IoT systems make compelling targets for cyber-attacks. In order to effectively compromise these systems, adversaries employ different advanced persistent threat (APT) methods, with one such sophisticated method, being botnets. By employing a plethora of infected machines (bots), attackers manage to compromise the IoT systems and exploit them. Prior to the appearance of the IoT domain, specialized digital forensics mechanisms were developed, in order to investigate Botnet activities in small-scale systems. Since IoT enabled botnets are scalable, technologically diverse and make use of current high-speed networks, developing forensic mechanisms capable of investigating the IoT Botnet activities has become an important challenge in the cyber-security field. Various studies have proposed, deep learning as a viable solution for handling the IoT generated data, as it was designed to handle diverse data in large volumes, requiring near real-time processing. In this study, we provide a review of forensics and deep learning mechanisms employed to investigate botnets and their applicability in the IoT environments. We provide a new definition for the IoT, in addition to a taxonomy of network forensic solutions, that were developed for both conventional, as well as, the IoT settings. Furthermore, we investigate the applicability of deep learning in network forensics, the inherent challenges of applying network forensics techniques to the IoT, and provide future direction for research in this field.

INDEX TERMS Internet of Things, IoT, nework forensics, botnets, deep learning.

I. INTRODUCTION

The Internet of Things (IoT) has exhibited a dramatic growth over the years. With Gartner reporting that the number of deployed IoT devices around the world are expected to reach about 20.4 billion in 2020 [108], displaying an increase of 145% from 2017, it is becoming evident that this new diverse domain will continue to grow as companies discover the benefits of IoT services.

As these numbers are on the rise, a growing concern for IoT systems is their security and privacy. In a study by Hewlett Packard in 2015 [85], it was shown that out of a number of IoT devices that were investigated, 80% raised privacy concerns, with 60% lacking any mechanisms that verify the authenticity of security updates or even their integrity, allowing an adversary to modify the firmware without being noticed. Another example of IoT vulnerability, is the study by Ling *et al.* [63] with its focus being a smart plug, a device that provides automation to mundane electronic equipment (fans, heaters). They were able to compromise the device and perform a number of attacks, one of which was a firmware attack, where an attacker can modify the devices firmware, gaining the ability to install malicious code to the device.

Seeing as IoT devices are manufactured with various pre-existing inherent limitations and vulnerabilities, it should come as no surprise that they have been targeted and recruited by botnets. Having the advantage of being designed to function 24/7, botmasters lately have shown their preference for using these devices instead of the better protected, not so reliable PCs and Laptops [59]. In their quarterly report on

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

the state on the Internet security for the 4th quarter of 2016, Akamai highlighted that we experienced the third wave of botnets, with the emergence of IoT-based botnets, with the first two harnessing PCs and servers respectively as bots [69]. One such prominent example of this new wave of botnets, is the Mirai botnet. It was first observed performing DDoS attacks against journalist Brian Krebs' blog, with the first DDoS peaking at 623 Gbps (77.9 GBps) and latter attacks targeting French web-host and cloud service provider OVH reaching 1.1 Tbps [59], [69].

Such attacks, apart from a discrediting factor that can affect the ability of a company to be perceived as trustworthy and reliable, can also have more immediate monetary repercussions, as most companies targeted by DDoS attacks rely greatly on their Internet connection to provide their services. Alternatively, some botnets have been designed to launch several types of diverse cyber-attacks such as identity/data theft (data exfiltration), where the Bot-code infecting a machine gathers sensitive user information and sends it to the botmaster, e-mail spamming, where infected machines are used to produce and send fake e-mail, key logging, where the user's input is logged and transfered to the botmaster and maleware propagation, through which a bot is used to further propagate a malware to its network neighbors and/or other Internet nodes.

With such destructive attacks on the rise, it is clear that security and forensics in the IoT should become a priority for research. In this paper, we provide a comprehensive background for the IoT, botnets and forensics, followed by a taxonomy of recent methods for botnet identification and tracking. We provide a new definition for the IoT, which ABBA.We investigate the applicability of deep learning in network forensics, and the inherent challenges that appear when network forensics techniques are applied to the IoT. In addition, we determine future directions for research related to performing forensics investigations of IoT powered botnets.

The rest of the paper is organized as follows. Section II gives background information on the IoT, some adopted architectures and underline technologies. Section III discusses the origin of botnets, their topological and propagation architectures and their activities. Section V discusses Digital Forensics, some of its expanding sub-domains, and then gives an overview of Network Forensic methods for investigating botnets, first in non-IoT environments and then in the IoT. Section VI discusses deep learning models and its role in network forensics. Section VII lists challenges in performing Network Forensic investigations of IoT environments. In Section VIII, future directions for research in the domain of Network Forensics of botnets in the IoT are depicted. Finally, in Section IX the conclusion is given.

II. COMPARISON WITH OTHER STUDIES

To the best of our knowledge, this is the first survey about forensic investigation techniques focusing on botnets in the IoT. However previous studies have investigated the three aforementioned areas separately. In this section, these aforementioned studies are presented. Additional, Table 1 depicts an overview of their characteristics.

- IoT Security/Botnets: These studies focused on botnets which primarily target IoT things, and their impact [10], [66], [113], [119].

  In their work, Angrishi [10] studied the composition of IoT botnets. The researchers outline several characteristics of IoT malware which set them apart from conventional malware, for example: (IoT) malware mostly does not affect the performance of the infected device, it resides in the RAM and generates DDoS attacks which are mostly volumetric or use unconventional patterns. The study then presented a number of IoT-targeting malware between the years 2008 to 2016, followed by lists of both major and minor IoT security incidents. Additionally, the researchers provide an abstract anatomy of an IoT botnet comprised of *bots*, *C&C*, *Scanners*, *Reporting server*, *Loaders* and *Malware distribution servers* as-well-as a typical life-cycle of IoT malware proliferation. Finally, best practices are presented such as changing default passwords and updating firmware, while an argument is made for Cyber Insurance [10]. This research provides a fair amount of information about IoT botnets their characteristics, operations, existing attacks and mitigating techniques. As such, it focuses more on how IoT botnets operate, and thus no information is given about how security incidents are forensically investigated, which is something that we cover in this study.

  Ransomware and its effects on IoT were discussed by Yaqoob *et al.* [113]. Based on the target of a ransomware, three categories are identified, *crypto, locker* and *hybrid*. Several penetration methods were then discussed, including botnets, social engineering and the use of IoT things as an attack vector, to spread the malware in an otherwise protected building. Finally, after presenting current research in IoT security, the work is organized in a proposed taxonomy, based on several metrics like *threats, requirements, IEEE standards, deployment levels, technologies*.

  A study on the impact of IoT in cyber security was conducted by Zhou *et al.* [119]. In their work, they proposed a new term, 'IoT features', which encapsulates a group of characteristics, unique to the IoT, which separates it from other computing systems. These proposed features, could help developers improve security and privacy shortcomings of IoT systems. Next, the researchers provided a statistical overview of research conducted in IoT Security, providing suggestions with regards to future endeavors in this discipline.

  A study by MacDermott *et al.* [66] addressed the changes that an "Internet of Anything "brings to the forensic field. They juxtaposed conventional types of crime and their investigative methods, to cyber-crime and discussed ways in which the forensic process can

TABLE 1. Comparison with other studies.

Studies	Botnets		IoT		Network Forensics			Rec
	Structure	IoT	Architecture	Security	Network Flow	IDS	Honeypots	
Angrishi et al. [10]	T	T	F	F	F	F	F	T
Khan et al. [57]	F	F	F	F	T	T	T	F
Nisioti et al. [81]	F	F	F	F	T	T	F	T
Garcia et al. [31]	F	F	F	F	T	T	T	T
Yaqoob et al. [113]	F	T	F	T	F	F	F	T
Hyslip et al. [44]	T	F	F	F	T	T	T	F
Singh et al. [102]	T	F	F	F	T	F	T	F
Ismail et al [47]	F	F	F	F	T	T	T	T
Zhou et al. [119]	F	F	F	T	F	F	F	T
MacDermott et al. [66]	F	F	F	T	F	F	F	T
Ghafir et al. [32]	F	F	F	F	T	T	F	T

be adapted to detect it. One key point that is discussed is that due to the ubiquitous nature of the IoT, the lines that separate networks from one another become blurry, hindering forensic investigations. As such, they identified several challenged that appear when investigating an IoT crime scene, such as jurisdictional issues. This study investigates the rising problems of investigating IoT-related cyber-crime from a high level of abstraction, focusing more on the investigative procedures and less on the techniques used.

- Botnet Detection/Network Forensics: These studies focused on network forensics techniques [31], [32], [44], [57], [102].

  Numerous Network Forensic Techniques (NFT) were surveyed by Khan *et al.* [57]. This study provided a critical investigation of NFT and their characteristics. One main contribution, was the proposed thematic taxonomy of NFT, based on eight different metrics. They then reviewed several NFT, each specialized to function in different scenarios, with the categories being: *traceback based NFT, converge network based NFT, attack graphs based NFT, distributed based NFT* and *NFT using intrusion detection systems*. Further classification of NFT was provided based on the different stages of the *forensics process*, the *data utilized*, the *time when NFT are invoked* (relative to the attack) and the *objective of the NFT*. Finally, the study listed challenges that affect network forensics procedures, some examples being: high connection speeds which overwhelm the collection process, privacy issues, data volume and integrity and more. Although this work provides a detailed and comprehensive taxonomy of NFT, it lacked any reference to the IoT and its impact in network forensics, which we address in our work.

  Network-based botnet detection methods were the focus of a study by Garcia *et al.* [31]. The researchers provided a well-structured taxonomy of previous work. Various detection techniques were identified in literature and analyzed in detail. Several issues were presented by this study, with relation to the published work that was reviewed. Amongst the issues raised, were a lack of information for reproducibility of experiments, problems with the dataset in use and clash of terminology between papers.

  Hyslip and Pittman [44] investigated the progress made in the field of botnet detection. Initially, the researchers provided an overview of detection techniques between 2005 and 2014, separating the solutions into 'old research'(between 2005-2010) and 'modern research'(2011-2014). They further categorized the detection techniques into four groups based on the C&C infrastructure, *IRC, HTTP, P2P, Encrypted* and further compared the solution of the two aforementioned time-groups. They concluded that detection techniques which relied on machine learning and flow data outperformed deep packet inspection solutions. Considering that botnets constantly evolve out of necessity, new C&C infrastructures have appeared over the years, which are not discussed in this survey, rendering this survey to be outdated.

  In a survey by Singh and Bijalwan [102], different forms of malware and botnets and their detection techniques were studied. The study initially provided an overview of several types of malware, such as *viruses, worms, trojans, rootkits and keyloggers*, and the main components of botnets. Next, the life-cycle of a botnet was discussed, followed by major groups based on C&C communication model and protocols in use. The detection methodologies that were presented, were ones that primarily target botnet traffic, namely *honeynets* and *passive traffic monitoring*. This study provided the general characteristics of a subgroup of the botnet detection methods available today, missing some popular solution, such as network flow analysis, and deep packet inspection. Additionally, honeynet solutions are underestimated by the research as being ineffective, even though they can acquire bot binaries, along with useful information about the botnets communication patterns.

  Similarly, Ismail and Jantan [47] reviewed botnet detection methods, that incorporated machine learning techniques. Initially detection methods were organized into two main categories: *Honeynet* and *IDS*, with IDS

further categorized. The research identified considerable work which employed machine learning to discover botnet traffic, indicating that it is a viable solution for processing large network-generate traffic, with some real-time solutions already proposed. Emphasis was given on IDS and specifically anomaly detection.
Ghafir *et al.* [32] developed BotDet, a novel method for detecting C&C communication in critical infrastructure. BotDet is dichotomized into two components. In the first component, four modules are tasked each with the identification of a single C&C type of traffic, these modules include a domain-flux and tor connection detection model. The second component is tasked with reducing the false positive rate by forwarding the activations of the previous four modules, in a correlation framework. The correlation framework works once per day and raises alerts, per user, based on how many modules detected an attack. Evaluation tests indicated that the proposed system displayed the best performance, when the correlation framework used two modules as input, achieving a detection rate of 82.3 % and a false alarm rate of 13.6 %. As mentioned by the authors, the system in its current form can be extended to detect more C&C traffic types, which will make it adequate for real-world application.

- Intrusion Detection Systems: These studies focused on the security techniques, called Intrusion Detection Systems [81].
The effectiveness of existing IDS against current network attacks, was reviewed by Nisioti *et al.* [81]. The study provided a classification of IDS based on the *Implementation*, *Architecture* and *Detection*. Additionally, the importance of feature selection when training the core model fo an IDS was emphasized. One of the conclusions of this study, was that a possible optimal choice for the core model, would involve a combination of supervised and unsupervised models. Clustering methods with irregularly-shaped groups outperform circular ones. The researchers stated that IDS should evolve to include correlation and attribution mechanics, to assist in the forensics process. Finally, they proposed that IDS should include three new classes of traffic, *data exfiltration*, *C&C communication* and *ransomware*, in order to be more effective at recognizing malicious activities.

Table 1 depicts the studies mentioned in this section, and their focus, split into the three major categories: Botnets, IoT and Network Forensics. The tags 'T' and 'F' are used to indicate references to the categories and subcategories for each study.

Overall, to the best of our ability, we were unable to identify any work that directly combined *IoT, Botnets* and *Network Forensics*. As such, we provide this study in order to address this gap.

III. INTERNET OF THINGS (IOT)

This section explains IoT concepts, growth of the IoT and its areas of application, as well as IoT Models and systems.

A. IOT CONCEPTS AND DEFINITIONS

Over the years, the IoT has evolved in complexity and functionality, maturing and becoming an integral part of society, spanning multiple fields of application. The concept of IoT has existed for quite a while, sometimes under a different name, like 'ubiquitous computing', 'embedded intelligence', 'web of things', 'Internet of objects' and 'ambient intelligence' [67]. The term 'Internet of Things' was introduced by Ashton *et al.* during a presentation in 1999 [11], where he explained the value of having computers that gather and utilize data in an automated and contextual fashion. In literature, IoT has various definitions, which we summarize in Table 2, along with their individual advantages and disadvantages.

Interestingly, the first time the "essence"of IoT was embodied, was around the beginning of the 1980s at the Carnegie Mellon University where a soft drink dispenser was coded to allow users to remotely view the availability of certain drinks [67], followed by Cambridge's Trojan coffee room, where a similar logic was applied to a camera which was used to check the amount of coffee remaining in a pot [25]. In 2000, Suresh *et al.* [105] produced a white paper depicting their views of the new MIT Auto-ID Center, where they described a world filled with objects connected to one another and tagged with relevant information, a vision similar to the RFID technology. From then onward, a number of events occurred which shaped the IoT into its current form. The first major adoption of this new idea was in 2000 when LG announced their plans to launch the first 'smart' refrigerator which could determine if the stored supplies were running low [25], [105], while a more formal introduction to the IoT was given by the International Telecommunication Union in 2005, through a report titled 'the Internet of Things' [89], [105].

In 2008, IPSO alliance was formed to promote the adoption of the Internet Protocol (IP) for the communication of "things", in what appeared to be the first step to start setting up common practices among the many vendors. Although work had been underway to develop the IoT, in one way, it was the creation of IPv6 that truly enabled its rapid development, as it allowed for virtually unlimited number of devices to be connected [25], [105]. Finally, in 2014 the Open Interconnection Consortium was promoted as an open framework for the Internet of Things by Intel and other firms [25]. Even so, there is still no standard framework for the IoT commonly adopted in industry, which forces vendors to decide on their own how to implement their devices, hindering somewhat the interoperability between differing IoT implementation.

The current diverse technologies involved, the fact that multiple communication protocols could be in use in a single infrastructure, and the mobility that characterizes the IoT, make it polymorphic in nature and contribute to the difficulties of pinpointing a single definition that best describes it in its entirety. Consequently, we provide a comprehensive definition for the IoT, which is *"the IoT is a network of networks comprised of devices, small and large named 'things',*

TABLE 2. **Overview of IoT definitions.**

Definition	Merits	Demerits
Ashton et al. [11]: A set of systems in which the Internet is connected to the physical devices using several sensors and actuators.	This definition shows that IoT links the physical and virtual world. It is a direction of executing tasks quickly.	The definition does not consider the scalability and compatibility of IoT's appliances.
Haller et al. [37]: The seamless interconnection of the information network and physical objects, called 'smart objects', with these object being active participants in business processes, being accessed through network services, with security and privacy in mind.	This definition shows the important role of the IoT for businesses, as a source of services to be provided. It also suggests that security and privacy need to be considered, when these devices are accessed.	This definition provides a business point of view, excluding home IoT systems. It does not consider the security of individual IoT device or the system as a whole. Scalability issues also not mentioned.
CERP-IoT report [18]: A dynamic global network, capable of self-configuration, based on interoperable standards and protocols, with 'things' having both physical and virtual aspects, built in a way to be seamlessly interoperable. Things are able to interact with themselves and their surroundings, and can be manipulated securely by users.	This definition acknowledges the duality of IoT devices (physical and virtual 'things'), mentioning also the concepts of interoperability and self-organization.	The definition does not explain how the IoT will be a self-sustainable system.
Gubbi et al. [35]: A cross platform interconnection of sensors and actuators, capable of sharing information, and achieved through a combination of ubiquitous sensing, data analytics, visualization techniques and Cloud as a unifying framework.	This definition describes the interoperability and interconnection, which make up the IoT, emphasizing more on the technologies that can be used for an implementation.	Asserts that the unifying framework, should be Cloud computing, other technologies are not considered. Scalability and security are not considered. IoT enabled services not in the scope of the definition.
Madakam et al. [67]: A network of intelligent, self-organizing objects, capable of sharing information and resources, sensing the environment and reacting to changes.	This definition shows that the IoT should be scalable, and that IoT devices should be able to react to sensory input.	The definition does not describe any user interaction with the devices, also security is not considered.

that have been imbued with finite amounts of processing power and communication capabilities providing services, including software, platforms and infrastructures to a remote user/organization on-demand, with lower cost than purchasing physical systems ". In other words, IoT is the creation of networks where machine-to-machine communication is used between geographical locations, industry/business sectors and other entities, whereby there is no direct communication. This can either enable software applications through the sharing of data, or allow for direct intervention of the environment where these IoT devices have been deployed. Such devices could be as complex as smartphones, which having multiple sensors and significant processing power or as simple as smart lightbulbs, that enable control of lighting conditions in a large environment such as universities [93].

B. GROWTH OF IOT AND ITS AREAS OF APPLICATION

Promising innovation, automation and optimization of industrial and commercial systems, no one should be surprised by the worldwide growth that the IoT market has experienced, with multiple studies and predictions made for it. A study by Gartner for instance calculated that in 2017 IoT deployed devices will reach 8.3 billion and project that the numbers will skyrocket to 20,4 billion in 2020 [108]. The predictions made for the economic growth of the IoT are better understood, if one considers the fact that the IoT does not cater to only a single portion of the world market, but instead slowly becomes an integral part for most fields in today's society.

In a white paper by the Internet of Things Alliance Australia (IoTAA), a segmentation of the IoT field into domains each of which describes a specific market was provided, with the domains being: Consumer, Industrial, Healthcare, Smart City, Automation, Agriculture, Critical Infrastructure [46]. Applications of the IoT for these domains appear constantly and in inventive ways. In the Agriculture domain, sensors for monitoring environmental data, such as levels of moisture in crops, air speed and temperature, placed the underpinnings for a future system [106]. In the Automotive domain, a monitoring system transmits and displays location and diagnostic data through a Cloud provider, improving the driving experience and assisting in determining the optimal time for mechanical services [43]. Finally, in the Healthcare domain, the incorporation of IoT devices has been shown to benefit patients, as vital information can be gathered from the comfort of their home, contributing thus to detecting quickly any deterioration in their condition [2].

C. IOT MODELS

As previously mentioned, the IoT has no single commonly accepted standard framework or set of standards. Instead, vendors are free to implement their systems by using the technologies they prefer, resulting in an heterogeneous IoT environment. The IoT is by design vast, spanning multiple technologies, which need to coexist in perfect harmony for the whole system to be functional. There are multiple ways of the IoT's designs and models.

First of all, from a communications point of view, the Internet Architecture Board (IAB) in a guiding architectural document released in 2015 described four communication IoT models [94]. First, the Device-to-Device Communication model, where devices communicate directly with one another. Such type of communication is primarily present in home automation IoT, and usually relies upon Bluetooth, Z-wave or ZigBee as the communication protocol, as they are ideal for the exchange of small amounts of information in relatively small areas. A drawback of this model, is that it requires its devices to collectively use the same communication protocol, limiting the devices that can be employed in this configuration.

In contrast, a more versatile model is the Device-to-Cloud model, where devices connect directly to a Cloud service provider to store collected data or receive instructions. This type of model allows the end-user to access their device through a Web interface or a smartphone app and view reports from a data collection, or change the state of the device. This model's drawback, is that in most cases, the Cloud provider and the Vendor who produced the device are one and the same, denying users from using a Cloud service provider of their choosing, a situation that is called 'vendor lock-in'.

Thirdly, an evolution of the Device-to-Cloud model, is the Back-end Data-Sharing model, which is an exact duplicate of the Device-to-Cloud model, with the added bonus that the user can extract data from the original Cloud provider and transmit it to other Cloud providers. This allows for data aggregation and has the benefit of giving the user the freedom of moving his/her data between Cloud providers.

Finally, in the Device-to-Gateway model, a device connects to the Cloud service provider through an Application Layer Gateway service, running on a local machine which functions as a proxy. The gateway in this model, apart from providing secure connectivity to the Cloud, allows for devices which use different communication protocols to interact, enhancing interoperability. In real world scenarios, in some situations, smartphones play the role of the gateway, with examples such as fitness tracking devices. In other scenarios, a 'hub' is used, which is a dedicated device that plays the role of the gateway and is most commonly found in the home automation scene.

An alternate way to view the IoT is given by the four-typed model, which splits the IoT into four layers [23]. The sensing layer, consisting of sensors and actuators, which enable perception of the world and the ability to act through the IoT. The networking layer, which handles communications between various network systems including heterogeneous devices of the IoT. The service layer, which allows applications to connect smoothly to the services provided by the IoT through the use of middleware. And the interface layer, which provides a means of interaction between various services in a system with the front end application.

The IoT ecosystem, which included its various components and the way with which they interact, can be viewed as follows, and as shown in Figure 1 [41]. First the IoT devices, sensors and actuators collect information and perform actions. The devices then through Coordinators, connect to the local Sensor Bridge which functions as a gateway, enabling at the same time interoperability between different protocols and technologies. Coordinators are tasked with health monitoring, data forwarding between devices and service provider and the creation of reports about all actions taken, while the Sensor Bridge connects the various heterogeneous IoT sub-networks with the service provider in the Cloud. The IoT Service handles many tasks, some of which are data storage, data processing and device management. Finally, through a Controller the end-user is able to connect the IoT Service and through that manage their devices.

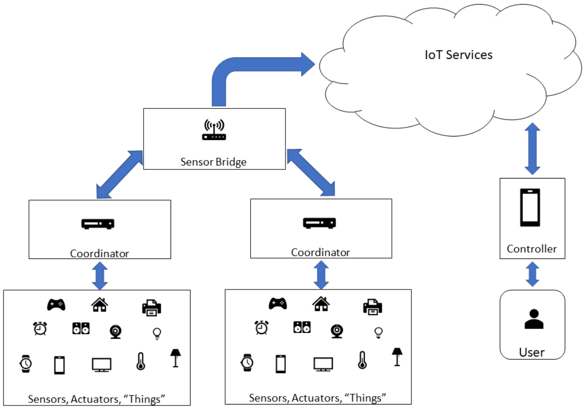

FIGURE 1. IoT Ecosystem adapted from [41].

D. IOT TECHNOLOGIES

As previously mentioned, the IoT is an amalgamation of several technologies and protocols employed on different levels of its ecosystem, enabling its functionality [67]. Different IoT technologies have emerged in the industry, for example Radio Frequency Identification (RFID), is used broadly as a cheap identification method for devices. For global communication between gateways and cloud service, the Internet Protocol (IP) is preferred, where both IPv4 and IPv6 are in use, with the latter allowing for close to 85,000 trillion IP addresses. For local communication between IoT devices and their coordinators, the most prominent technologies employed are Wi-Fi, Bluetooth, ZigBee, ZWave. Finally, regarding the actuators, they are generally split into three categories: Electrical, Pneumatic and Hydraulic, based on the medium they use for power. These technologies are vulnerable to cyber-attacks due to the IoT open-loop of communication, and heterogeneity of their protocols and services. We mainly focus on botnets, as they constitute considerable harm for IoT appliances and applications, as explained in the following section.

IV. BOTNETS

In this section, information related to Botnets, their origin, architecture and activities is given.

A. BOTNET BACKGROUND

Botnets have had a rich history and development over the years, corrupting and disrupting computer and network systems [101]. Originally, botnets were crafted for benevolent purposes, with their main functionality being to provide administrative assistance to Internet Relay Chats (IRC), a form of communication quite popular in the '90s. The first

IRC bot appeared in 1993, was named Eggdrop and provided assistance to IRC channel communication. Following Eggdrop, the first malicious bots made their appearance, with GTbot in 1998 being the first of its kind, which was able to execute scripts when prompted through its Command and Control (C&C) IRC channel.

In 2003, new more sophisticated bots appeared like the Agobot, which was more robust and flexible than previews types, as-well-as it incorporated a persistent C&C channel. In 2007, Storm made its appearance, a botnet that was characterized as one of the most powerful botnets of its time. It employed a P2P C&C infrastructure, with its main functionality being spam messages, DDoS attacks and had the capability to disrupt the Internet communication flow of entire countries. In the same year, appeared probably one of the most infamous botnets, the Zbot or Zeus. Having at the time close to 3.6 million bots under its control, other variants were later spawned, including a P2P version in 2011 named Gameover Zeus, which was capable of performing a wide range of malicious activities, including bank account theft, DDoS and spam [9]. It was eventually taken down by a collaboration of the FBI, the UK NCA, Shadowserver Foundation and Dell's CTU IN in June 2014 [71].

In 2008, some notable botnets that appeared were Asprox, Kraken, Torpig and Conficker. Asprox, like the original Zeus, used a centralized HTTP based C&C infrastructure and apart from its main purpose, which was the creation of spam, it was also capable of performing SQL Injection attacks to websites. Kraken, was part of the spammer botnet family, and was reported that in April 2008 included 400,000 bots in its army of zombies [51].Torpig was a data exfiltration botnet, which performed man-in-the-browser attacks and used a centralized HTTP-based C&C infrastructure [33]. Conficker, moved periodically from Centralized HTTP-based C&C to P2P [104].

The successor of the Storm botnet, Waldec was discovered in 2009, having a P2P C&C infrastructure, its primary function was to send spam messages reaching close to 7000 messages per day, though it also performed credential theft and DDoS attacks [104]. Eventually it was taken down in 2010. Also in 2009, Mariposa was discovered. Mariposa used a custom communication protocol which was a variation of UDP, was capable of launching DDoS attacks and even download and run executables, such as other bots. It was taken down in December 2009 [45]. Another notable milestone for botnets in 2009 was the appearance of the precursor of mobile botnets, where botnets use mobile phones as their bots (zombies), named SymbOS\Yxes which targeted Symbian devices and utilized SMS messages to self-propagate [27], [111].

Following the surfacing of SymbOS, the first botnet targeting Android devices named Geinimi was observed, during the end of 2010. Primarily found in China, it employed a simple HTTP-based C&C infrastructure and was capable of sending SMS, e-mails, fetch the location of the infected device and also made possible the further propagation of malware [80], [111]. Lately, botnet creators have taken advantage of the wide adoption and constant increase of the IoT, and we have already scene examples of IoT botnets and what they are capable of.

Botnets comprised of IoT devices were the next evolutionary step of botnets. The most well-known first appeared in September 2016, under was aliased as Mirai [59]. Mirai performed some of the most powerful DDoS attacks in Internet History, namely: 620 Gbps against Brian Kreb's website, 1.1 Tbps agains French Cloud service provider OVH and in October 2016 attacked Dyn service provider and took down portions of the internet like Twitter, Netflix and GitHub. After the release of Mirai's source code, various variants appeared like Persirai which is active since April 2017, a more refined version of Mirai which targets specific devices of select vendors. Other IoT botnets include Hajime, which appeared in October 2016, and utilized a decentralized C&C infrastructure which appeared to 'shield' devices from Mirai infections. Finally, BrickerBot was observed in April 2017, and as the name suggests attempted to 'brick' IoT devices in what can be considered a permeant DoS attack.

B. BOTNET ARCHITECTURES AND CHARACTERISTICS

Botnet architectures include several elements. To start with, a bot is a program which, after reaching a vulnerable host, infects it and makes it a part of the Botnet [58], [101]. Bots differ from other malware, in that they include a channel of communication with their creators, allowing them to issue commands to their network of bots (i.e., zombies) and thus making botnets versatile when it comes to their functionality [58], [101]. A botnet's malware gets delivered to vulnerable targets through what is known as a propagation mechanism. Most commonly there exist two types of propagation, passive and active.

Passive propagation techniques require users to access sites, emails or other compromised network elements and through user interaction download the malware (bot), infecting it and making it part of the botnet [58], [78]. Active or self-propagation techniques employ sub-portions of their network to actively scan the Internet for vulnerable devices, attempting to exploit the identified vulnerabilities, turning the compromised hosts into bots themselves [58], [78].

The characteristic that makes botnets unique is the fact that they allow their controller, commonly referred to as a botmaster (a.k.a botherder) to issue instructions to their network of infected devices and receive feedback, as shown in Figure 2. This is made possible through a Command and Control (C&C) infrastructure. There exist multiple different types of C&C infrastructures based on their topology and those types are: centralized, P2P, hierarchical and hybrid [14], [78]. In a centralized topology, bots connect, receive instructions and report/deliver their work in a central infrastructure, with most common technologies employed here being IRC and HTTP protocols [14], [101]. The main drawback of the centralized topology is that the C&C is a single point of failure.

A decentralized or P2P architecture is the natural successor of the Centralized, where the bots can assume either the

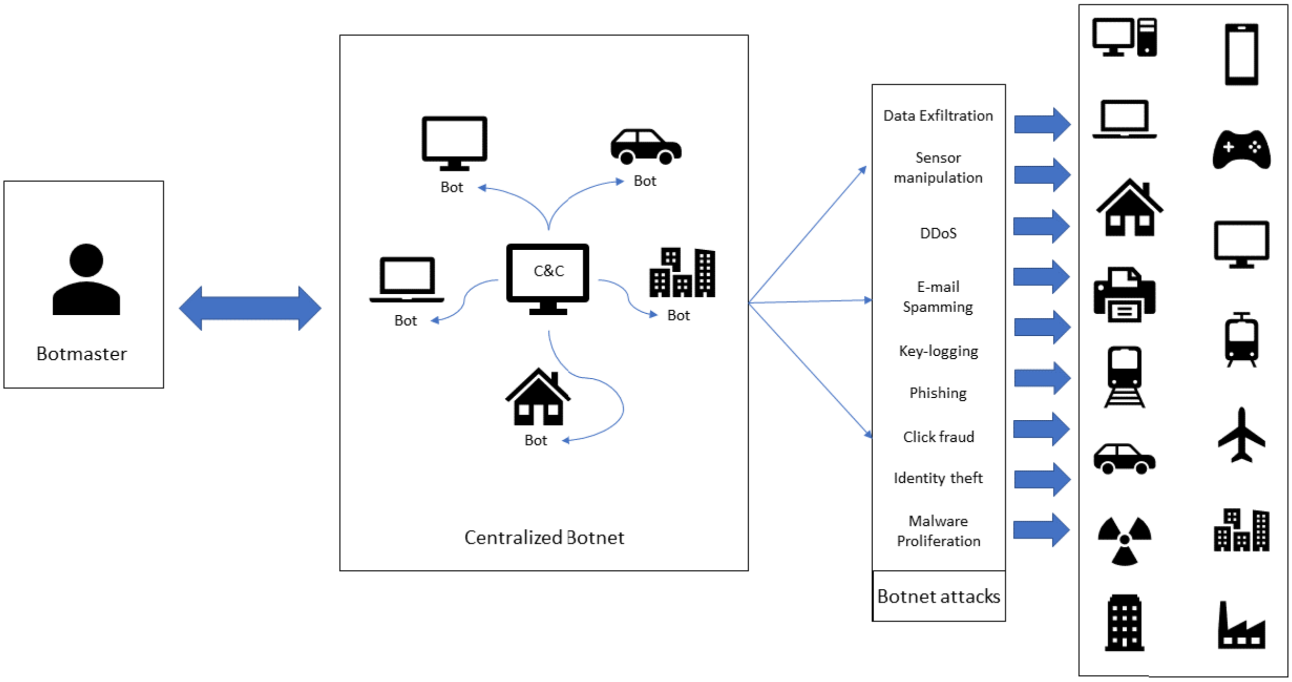

FIGURE 2. **Centralized Botnet and activities [99].**

responsibilities of a C&C server or a worker bot that performs tasks on behalf of the botmaster. Such architectures generally face higher latences than preferred regarding command distribution, though they are quite resilient to takedown attempts, as the compromise of a single host would only affect a small portion of the botnet [14], [58].

A Hybrid architecture is, in a way, a combination of the P2P architecture with the Centralized, reaping the benefits of both [14], [58]. Here, the C&C is implemented in P2P form, with the bots that make up the C&C (called servant bots) forwarding commands to each other and to the bots that perform the actions (client bots). Finally, in this architecture, a botamaster adds proxy bots between their machine and the botnet, with each bot forwarding commands to the bots that they compromised, creating a hierarchical topology and making takedown attempts difficult, as-well-as allowing the botmaster to rent portions of their botnet.

C. BOTNET ACTIVITIES

Botnets are some of the most versatile pieces of code to traverse the Internet. The main reason why they get so much attention is not because of the masterful ways that botmasters employ to obfuscate their bots from law enforcement, but rather the practical capabilities that botnets possess and the services they provide to the botmasters and their clients. There are various hacking techniques used by botnets, including Distributed Denial of Service attacks (DDoS), Keylogging, Phishing, Spamming, Click fraud, Identity theft and even the proliferation of other Bot malware [7].

Botnets tend to be specialized into performing a small subset of the aforementioned hacking techniques, though there have been cases where variants of botnets were capable of multiple types of malicious activities, an example being Gameover Zeus which could perform DDoS attacks, send spam e-mails and steal bank account information. There are many ways a botnet can perform a DDoS attack, and based on the technique and protocols employed, there are multiple examples of such attacks, some of which are: Denial of Sleep attack, UDP flood attack, TCP SYN flood, ICMP ping flood, Ping of Death, Smurf attack, DNS amplification, HTTP flood [68], [116]. In a Denial of Sleep attack, the attacker targets functionality provided by the Medium Access Control (MAC) layer, where devices are set into a 'low power mode', to decrease battery consumption, which is of vital importance for network sensors [68].

A UDP flood attack exploits the connectionless nature of the UDP protocol and sends a large number of forged packets to random ports of the target machine, forcing it to expend resources to detect any applications which could be waiting to receive the incoming information and then issuing ICMP responses when that check fails [116]. On the other hand, TCP SYN flood attacks utilize the structure of the 'three way handshakes' that is performed in order to set up the parameters for any TCP connection. In this case, the attacker floods the target with SYN packets, which are used to initiate a TCP session, receives the response from the target but does not send the final packet that establishes the connection, forcing the target to maintain the connection open and thus eventually cause the target to become unresponsive [68], [116].

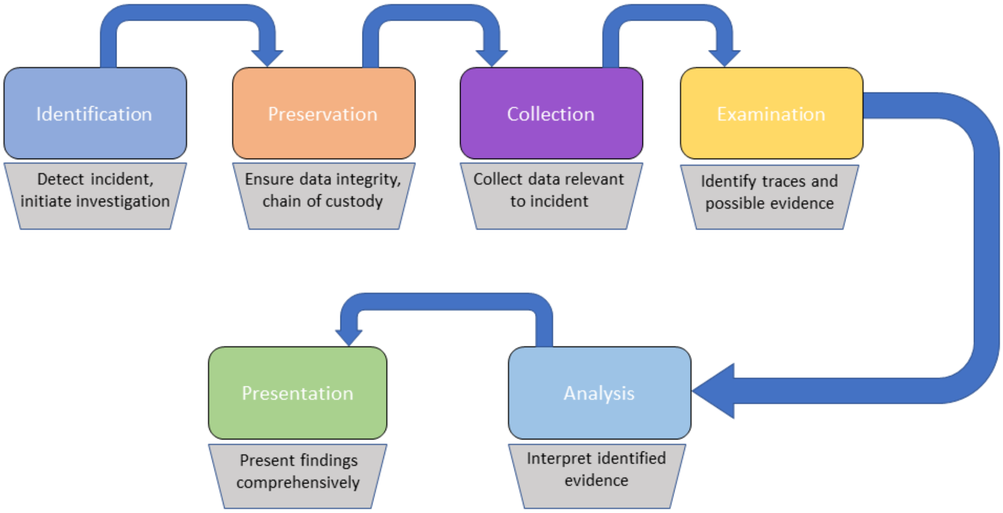
Identification
Detect incident, initiate investigation
Preservation
Ensure data integrity, chain of custody
Collection
Collect data relevant to incident
Examination
Identify traces and possible evidence
Analysis
Interpret identified evidence
Presentation
Present findings comprehensively

FIGURE 3. Phases of digital forensics mechanisms (DFRWS Investigaiton Model).

An ICMP ping attack is similar in nature to the UDP attack, in the sense that a large number of packets are sent to a target which forces the target to respond, taking up network and processing resources [68]. A Ping of Death, though it utilizes ICMP is somewhat more interesting, as it exploits the fact that the IP is designed with an upper limit of 65,535 Bytes, and when a larger packet is received, it causes memory overflow issues and eventually crashes the machine [68], [116]. Finally, a Smurf attack consists of a large number of ICMP packets which have the intended target's IP address spoofed in place of the packets' source address, causing the replies received from such packets to be sent to the target [116]. When a botnet performs keylogging, it silently records the keystrokes of a user and after a certain amount of time, it sends its gathered information to the botmaster. Phishing is the process through which the adversary attempts to trick a user in revealing sensitive information, such as login credentials or even bank account information, through carefully crafted messages, websites, and emails. Finally, botnets are sometimes used to proliferate other malware, spam email being one way to do so.

There are different security controls, for example threat detection and intelligence, as well as intrusion detection techniques have been used for recognizing and preventing botnets from network and IoT systems. These techniques are beneficial to some extent, because they can only detect knows cyber-attacks, but they cannot identify zero-day attacks (i.e., new/future attacks), as there are no signatures of those attacks stored in blacklists. This is the motivation of focusing on digital forensics mechanisms in order to track and define cyber-attack origins, and assist in examining how botnet structures occur in IoT systems; hence improving security controls in discovering known and new botnets.

V. DIGITAL FORENSICS

In this section, information related to Digital Forensics, its origin, investigation models, sub-domains and developed methods for investigating botnets in multiple fields including the IoT is presented.

A. ORIGINS AND EVOLUTION OF DIGITAL FORENSICS

As criminal activities moved to cyberspace, with cyber criminals exploiting systems to their own ends, it was only natural that law enforcement would also adapt their operations accordingly, as such, digital forensics was coined. Its roots can be traced back to 1984, when law enforcement entities, among which was the FBI laboratory, started developing programs in order to examine computer-related crimes [54], [117]. Over the years, many organizations have proposed their own definitions and standards for performing forensics investigations in the digital world, with multiple investigation models appearing, most of which share some common phases but are designed to be applied in different circumstances [117].

One such definition that describes the essence of digital forensics, is the one given by Rodney McKemmish, where he states that digital forensics is *"the process of identifying, preserving, analyzing and presenting digital evidence in a manner that is legally acceptable"* [91]. An investigation model proposed by the first Digital Forensics Research Workshop in 2001 named the DFRWS Investigation Model, which functioned as an inspiration for other such models, comprised of six phases: Identification, Preservation, Collection, Examination, Analysis and Presentation [54], as shown in Figure 3.

In the Identification phase, sources of possible evidence are identified. This phase takes under consideration that the amount of data an investigator can collect is constrained [117]. In the Preservation phase, proper chain of custody is established, and further actions are taken to ensure the integrity of data to be collected [117]. During the Collection phase, the investigators make use of appropriate techniques and tools to safely collect the data which has been identified as important for the case.

The Examination and Analysis phases are considered to be of utmost importance, as here the collected data is scanned, filtered and processed in order to identify crucial evidence and establish timelines which are then provided in reports during the Presentation phase [54]. During the Investigation phase, a range of devices could be investigated for potential evidence, from mobile phones and laptops to routers and lately even fridges and light-bulbs. Some well-known digital forensics investigation models are listed in Table 3.

Over the years, digital forensics have been further partitioned into sub-fields, each of which provides specialized techniques for investigating security incidents in different domains of the IT sector. Popular forensics sub-fields are Network forensics, Cloud forensics, and IoT forensics [49], as described in the following points.

- **Network forensics**- emerged as a way to identify, understand and ultimately amass evidence to pursue legal action for malicious activities that used the Internet and other networks as a bridge for attacks with some examples being DDoS attacks and data theft [57]. In a network, evidence is usually short lived, as packets are produced from one device and sent through intermediary nodes to their destination. As such, various Network forensics techniques have been developed over the years [42], [57], [74]. Famous tools in network security are Intrusion Detection Systems (IDS) and Honeypots [56], [60], [75]. IDSs are trained and validated to recognize patterns of malicious traffic in the network [42], while Honeypots mimic vulnerable legitimate devices, luring hackers and botnets into attacking them, with the added bonus of allowing investigators to observe what actions are performed by the attacker [57].
- **Cloud forensics**- is a branch of digital forensics tasked with investigating security incidents in the Cloud [96]. It is a cross disciplinary field combining disciplines like computer device forensics and network forensics, which poses some unique challenges, like difficulty in defining jurisdiction, as a Cloud provider could be based in Europe and be providing services in the U.S., breach of privacy, as a machine that could be investigated could host services for multiple users, including suspects and an increase in generated data quantity, as an ever-increasing number of devices utilize the Cloud [96].
- **IoT forensics**- is an emerging new field of forensics for investigating cybercrimes by analyzing IoT devices, protocols, in terms of software-, platforms-, and infrastructure- as services. IoT forensics is slowly being developed as in [55]. Major challenges that are hindering the adoption of conventional digital forensics techniques for investigating incidents in the IoT, are heterogeneity of systems and data, the high quantity of data produced as the number of IoT devices rises constantly and the speed with which data is generated.
- **Malware forensics**-is a discipline of forensics, which focuses on reverse engineering, and analyzing the source code of malware [17], [109] acquired from captured binaries. Analysis of malware samples can be categorized as *static, dynamic or code*, depending on the methodology used. In addition to these methods, virtual machines have also been used, as they provide a resilient environment where malware behavior can be observed in relative safety. Over the years, attackers have started to incorporate anti-forensics logic in their code, to elude detection [109]. With anti-forensics techniques, malware infections become more resilient, for example, allowing it to alter its behavior, if it identifies that it is running in a virtual environment.

B. NETWORK FORENSIC METHODS FOR INVESTIGATING BOTNETS

Investigating botnets, is a multifaceted problem. It requires interdisciplinary actions to be taken, to ensure effective analysis and a more spherical understanding of an infected network's actions, enabling the design of better counter measures, or at the very least attribution. As mentioned above, Network Forensics is the branch of Digital Forensics, where the evidence is network-related, and thus exist in the form of logs, packets and network flows. In this section, we focus on Network Forensic techniques, which have been developed to analyze Botnet activities, their general characteristics (lifespan, size) between the years 2011 and 2018. These techniques have been organized into distinct methodologies as follows:

- Honeypots
- Network Flow Analysis
- Deep Packet Analysis
- Attack Recognition
- Visualization of Network Traffic
- Intrusion Detection Systems (IDS)

These methodologies are further discussed below.

- **Honeypots**- [48], [62], [73], [77], [87], [87], [90]- are devices, many times simulated ones which run in a controllable virtual environment, that have been designed to appear as an appealing target to attackers and malware infections. Honeypots are generally separated into high interaction and low interaction honeypots, with the former imitating entire systems (e.g., Windows, Linux) and allowing for extended interactions and information gathering, while the latter simulates certain services available through the net, which are often targeted by attackers. The benefit of this method is that the attacks, malware binaries and access attempts, are monitored and logged by the Honeypot operators, allowing for the generation of rules to predict similar future attacks. Additionally, it enables the extraction of malware binaries and communication patterns between C&C and bot,

TABLE 3. Well-known digital forensics investigation models.

Author	Model name	Model description
Pollitt M. [91]	Computer Forensic Investigation Process 1984.	An investigation model general enough to be applicable in a wide range of computer-related crimes. It was proposed, as a parallelism to the conventional evidence handling process. Comprised of four steps, it does not include steps for preparation, investigation and subsequent returning of evidence to their rightful owner.
Palmer G. [86]	DFRWS Investigative Model 2001.	A general purpose investigation model proposed during the Digital Forensics Research Workshop (DFRWS) in 2001. It somewhat expanded the Computer Forensic Investigation Process, by trading the Acquisition step for three new steps, Identification, Preservation and Collection, thus emphasizing the necessary steps to identify potential sources of evidence, secure them in a way that provably prevents any alterations to their state and then proceeds with collection and the subsequent steps of the investigation.
Baryamureeba V. et al. [13]	Enhanced Digital Investigation Process Model (EDIP) 2004.	An extension of the previously proposed Integrated Digital Investigation Process model, the EDIP has the advantage of not only considering the digital crime scene but also the physical. The EDIP model includes: Readiness phase, Deployment phase, Traceback phase, Dynamite phase and Review phase, and each phase is comprised of multiple sub-phases. By partitioning the possible actions that an investigator needs to take, and including a traceback phase, this operational model is one that can be easily applied to real world scenarios.

while it does not take part in any of the attacks issued by the botmaster.

A low-interaction honeypot system was proposed by Pham and Dacier (2011) [90], who focused on developing an automated and systematic way of identifying Botnet attacks in vast datasets. By employing the services of *LeurrÃl.com*, which includes multiple machines in more than 25 countries, running low-interaction honeypots and collecting network traffic, they were able to show that by grouping together closely correlated traces of attacks, they were able to identify attack events.

Kumar *et al.* (2012) [62] proposed a distributed virtual and fully automated Honeynet architecture, capable of dynamically reconfiguring itself. Their proposed system is partitioned into three components. In this system, a distributed honeynet client, which is comprised of a combination of low interaction and high interaction honeypots, would run in a VirtulBox with the incoming and outgoing traffic moderated by a Honeywall.

A method for extracting Intrusion Detection System (IDS) rules from data collected from Honeypots, was developed by Mittal and Singh (2016) [73]. The researchers employed a Support Vector Machine, which they trained by using data collected from a Honeypot to produce the necessary rules for the IDS. Not much information is given regarding the source of the Honeypot-produced dataset or the parameters of the SVM that was used. Although Honeypots can produce extensive information regarding Botnet activities, they require huge amounts of storage space to maintain all the traces, extracted payloads and malware binaries. Low and High interaction honeypots have their tradeoffs, with the former needing a lot of effort to pass as a legitimate device, and the latter running the risk of being taken over completely, thus requiring data control measures to be taken, so that the honeypot does not take part in any malicious activities itself.

Attackers have been known to target social networks, where they are capable of gaining sensitive information from unsuspecting users, proliferating malware infections and more. As such, Paradise *et al.* (2018) [87] presented *ProfileGen*, a tool that produces realistic and compelling social profiles, which function as honeypots, in that they attract the attention of attackers. The proposed system, makes use of an automated process that relies on the generation of a Markov model from collected data. Emphasis was given in generating realistic education records for the crafted profiles.

It is not uncommon for cyber-attacks to span large segments of the Internet, with a prominent example being DDoS attacks. To that effect, Jeong *et al.* (2018) [48] worked on tracking large-scale events. Apart from improving accuracy, emphasis was given on reducing the communication toll due to sensors reporting an observed event in a distributed monitoring system, akin to the honeynet. In the proposed detection protocol *Bitmap-Based Widespread Event Detection*, bitmaps are exchanged between the agents and the coordinator, and are used by the latter to identify events monitored by all agents, which are then categorized as widespread events. This approach improves on previous schemes, as it does not produce any false positives.

Naik *et al.* (2018) [77] proposed a fuzzy-based technique that identifies fingerprinting attacks in a low-interaction honeypot. This technique identifies abnormal TCP, UDP and ICMP traffic, and uses fuzzy logic to produce a probability of a fingerprinting attack targeting the honeypot. One shortcoming of this technique, is that new fingerprinting attacks that rely on different patterns might not be identified effectively.

- **Network Flow Analysis**- [15], [16], [24], [28], [83], [103]- uses metadata gathered from network communication, to make inferences regarding the legitimacy of the traffic under scrutiny. Traffic is aggregated, and metrics are collected to create Network Flow Records. Records are identified by the source and destination IP addresses, port numbers and the protocol used for the communication. The benefit of this approach, is that privacy is no concern, as the actual data being communicated isn't investigated and it doesn't require extensive storage capacities, as traffic is aggregated, with only certain metrics maintained. At the same time, it is not affected by encrypted communications, a problem faced by Deep Packet Inspection (DPI) while at the same time it does not make use of signatures to identify network attacks, as it relies on statistics and the application of machine learning.

  In their work, Francois *et al.* [28] created a system which focused on identifying members of P2P botnets. In their implementation, they employed Hadoop, the open source implementation of MapReduce, to run the PageRank algorithm on trace data collected by honeypots. Hosts with high connectivity to each other were categorized as potential P2P bots, as this characteristic was deemed a good indication that a host is part of a P2P botnet. Their approach, performed adequately on certain types of P2P botnet topologies, where linkage between bots is high, although some legitimate P2P clients could be misrepresented as bots, if no prior knowledge is used to fine tune the PageRank process. Bijalwan *et al.* [15], focused on the investigation of UDP flooding attacks. In their experiments, they worked on identifying randomized UDP flooding attacks, which can be designed to pass undetected by IDS systems and other security mechanisms. They simulated an attack environment, by developing scripts which would extract the source IP address of the user that would be targeted, and then generated the randomized attack payload (forged packets).

  Detecting suspicious patterns in network traffic by utilizing a Linear Regression model was the focus of Divakaran *et al.* [24]. Their solution relies on the combination of network flows into *sessions*, and the detection of illegitimate TCP states by using Finite State Machines to determine whether an attack is taking place, gathering evidence in the process. Oujezsky *et al.* [83] focused on time behavioral analysis, by extracting information from Network Flow data aggregations. The researchers employed survival analysis, a technique based on probability theory, developed to study the duration of virtually any process, and focused on the identification of C&C communication, which tends to be periodic in nature. The merits of such a technique, as mentioned by the researchers, is that deep packet inspection is not necessary, as they focus on analyzing timing data from Network Flows, thus bypassing Law restrictions, concerns of privacy and the time-consuming process of inspecting all collected packets. Additionally, this method does not require knowledge of when a Network Flow occurred, but how long it lasted, rendering the process of trying to convert from one time system to another obsolete.

  Bou-Harb and Scanlon (2017) [16] focused on distributed malicious events that take place in vast areas of the internet, dubbing them campaigns, with their goal being the development of efficient techniques to analyze vast quantities of network traffic and produce network forensic evidence. However, this approach focused on the identification of probing botnets alone, which means that other botnets that do not employ such mechanics will not be detected. Additionally, as mentioned by the researchers, this approach suffers from poor scalability, that is, when the number of infected machines probing the system exceeded 1000, both false positive and false negative rates started to increase. A drawback of using network flow analysis, compared to other methods such as Deep Packet Inspection, is that it bypasses the payload of packets which means that certain information is ignored. Thus, it can't be used for malware and command extraction from traffic and some attacks that rely on the payload such as SQL injections can pass undetected.

  Sivaprasad *et al.* (2018) [103] proposed a monitoring system which relied on machine learning techniques applied to network flow-summarized data. The researchers used a combination data from the CTU-13 dataset and packets that were collected from a DDoS attack they performed. Their task was to create a user-friendly interface, where data was uploaded to the system, pre-processed with Weka and after the feature-extraction process, NaÃŕve Bayes and SVM classifiers were built. On a similar note, Mathur *et al.* (2018) [70] proposed a model for botnet traffic discrimination. The researchers investigated the predictive capabilities of five different classifiers, *randomized filtered classifier, logistic regression, random committee, random subspace, multi class classifier*. After features were extracted from a combination of data from CTU-13, ISOT and live captures, the five classifiers were trained and compared. It was then shown that the logistic regression and multi class classification algorithms had the best performance in both accuracy and false positive rates. This work is a good indication of the relative performance of various classifiers when tasked with botnet identification, even though neural networks were not included.

  Kozik (2018) [61] used a distributed Apache spark environment to train an *Extreme Learning Machine (ELM)* classifier. ELMs differ from other neural networks, as they often include a single hidden layer, and have different activation functions. The key contribution of this work, was that the distributed environment was incorporated in the training of the ELM classifier, by splitting calculations performed at the hidden layer into chunks of computation which could be performed by the distributed environment's elements. Results showed that the proposed method is promising.

Pektas and Acarman [88] proposed a deep learning system to process network flow patterns and identify botnets. In a botnet, communication between C&C and bots is frequent, as such their approach was to target these channels. Their choice of deep learning was justified, as their method relies on processing large quantities of data, in which deep learning thrives. During feature extraction, collected flows were turned into graphs, grouped by communication endpoints, which allowed them to produce new statistical features. A number of different configurations were tested, with varying numbers for layers and neurons. The researchers concluded that deep learning presents acceptable accuracy for botnet identification in flow data, with the added bonus that feature selection is not necessary, as deep networks identify the best features.

- **Deep Packet Analysis (also known as Deep Packet Inspection or DPI)**- [20]- is a form of packet filtering, that relies on the inspection of both headers and the data segment of a packet, for the purpose of identifying malicious traffic based on known patterns. Although it raises privacy concerns, faces problems when trying to parse encrypted traffic and relies on a signature database to perform its identification, thus being unable to identify Zero-day malware, DPI is still used to this day. By scanning the content of packets, more information can be gathered, and the behavior of the packet's origin can be better understood.

  Chen *et al.* [20] proposed and implemented a cloud-based collaborative network security management system, which takes advantage of Cloud storage and processing, to perform offline forensic operations on captured raw network traffic with emphasis given on SPAM incidents. The research team utilized the Collaborative Network Security Management System (CNSMS), which managed the security of four different sites (networks), by making use of NetSecu nodes and Probers to gather, monitor and manage network traffic. The NetSecu nodes have the advantage of interacting with locally deployed security mechanisms (such as firewalls and IDS), dynamically responding to security incidents threatening the network. Additionally, they can be integrated with self-protection solutions, in the event they becomes the target of an attack. Furthermore, NetSecu nodes communicate with similar devices deployed in other networks, thus creating an overlay network. It was observed that this cloud-based scheme could be applied to the investigation of other network-related security incidents.

  Cheng and Watson [21] developed D^2PI, a system that identified malware in network traffic, by using a Deep Neural Network. The proposed system was a Convolutional Neural Network (CNN) which classified collected traffic, into either 'malicious 'or 'benign ', based solely on the payload. After extracting the payload, their length was regulated to a predefined length and incorporated in a matrix, in order to be processed by the CNN. Results indicated that, although more work is needed to improve this method, it is a promising first step towards incorporating CNNs in DPI systems.

  Another mechanism that has been leveraged in DPI systems, is *finite state automata*. Finding ways to improve these mechanisms, in order to be able to handle the ever increasing variety and volume of traffic was the focus of Yin *et al.* [115]. As regular expressions, which can be used to identify complex patterns in packet bodies, are often implemented in finite state machines, improving their efficiency is of vital importance. Initially, the researchers discussed deterministic and non-deterministic finite state automata, comparing the memory requirements for each category. They concluded that a non-deterministic approach is needed for a DPI implementation, and provided improvements that help reduce processing time and memory consumption. Experiments s that were made, between two automata, based on regular expressions from Snort, showed that a non-deterministic finite state automaton machine with the proposed improvements used less memory, as the number of conversion edges were reduced.

- **Attack recognition**- [12], [38], [53], [120]- is a collection of machine learning techniques applied to any number of sources (e.g., network traffic and logs). It is utilized in investigating the identification of known patterns exhibited by malicious software. Although pattern identification is generally used in some form or another in different techniques, in this section, such patterns are identified not only in packets, but from other sources as well, like network logs, and are used to better understand the sequence of events which lead to an attack. This forensics method, in some situations, requires access to the physical device, from which logs and files are extracted and then examined offsite.

  In their approach, Zhu [120] developed an algorithm, that identified a sequence of attack events, by scanning network logs. Their algorithm identified what the research team dubbed 'attack bubbles', with high suspicion values. An 'attack bubble' was defined as a tuple containing: a collection of network events found in the logs, a suspected type of attack which these events might constitute, a probability that the identified events constitute the suspected attack, and an identification of the source of these events (IP address). Han *et al.* [38] proposed a technique that examined the communication behavior of network nodes. In their process, they focused their efforts on identifying Command and Control communication which exhibit a pattern of synchronicity, for instance, the C&C server sends instructions to all botnets simultaneously, and all bot respond at the same time.

Karthika [53] proposed a highly scalable system for detecting stealthy peer to peer botnets, akin to an Intrusion Detection System. Their system identified all P2P (Peer-to-Peer) traffic in a network, relying on DNS look-ups which the system collected, and reasoned that most P2P applications do not rely on DNS to establish the destination IP address in legitimate P2P communication. A system for investigating the existence of botnets, composed of several modules was proposed by Bansal *et al.* [12]. This system gathered all network data that are sent or received by the internal network, creating a repository of stored packets, which were later filtered, flagged as legitimate or illegitimate traffic and then used to identify the presence of any unknown botnets. The process would then scan any involved systems, to gather traces that were not identified by previous steps. In the final step a report was generated, with all identified evidence being visualized.

- **Visualization of Network traffic**- [8], [36], [50] a number of diverse visualization techniques have been employed, to improve Botnet investigations. One use of such techniques is as a support tool for investigators, which can assist security experts to track the rout taken by a malware infection carried out at large workstation areas.

  Such a visualization method was developed by Joslin *et al.* [50], who studied the representation of Network Flows (or IPFlows) as a directed graph, combined with relational information, making the argument that the proposed combination motivates the creation of new hybrid graph-relational systems. Concentrating more on the visual representation of network traffic and security incidents, Gugelmann *et al.* [36] produced Hviz, a traffic visualization program that processes HTTP/S traffic in order to reduce the number of events that an investigator would have to work on. They employed various mining techniques (FIM) to aggregate the sites visited by workstations during an investigation. Also, by comparing traffic between workstations, they attempted to figure out if the traffic in question is malicious or not. On the other hand, an investigation of UDP flooding attacks, by using the UDP flow graph was performed by Anchit and Harvinder [8]. The researchers made use of a testing environment, setup in their lab, choosing Wireshark for the collection of network traffic. As stated by the researchers, this approach could assist investigators in identifying network attack incidents, as patterns of network communication become easier to spot with the naked eye.
- **Intrusion Detection Systems**- [6], [56], [60], [74], [75] are generally a defensive mechanism deployed either in the network (Network IDS) or in a device (Host IDS), which can either employ pre-made signatures (signature-based IDS) or machine learning (anomaly-based IDS). In the context of Network Forensics, IDS systems can function as alarm triggering mechanisms which, after identifying malicious activities (for example Botnet traffic), can raise further forensic mechanisms, allowing for an automated solution.

  AlRoum *et al.* (2017) [6] developed a Botnet Detection System that focused on DNS records, as some botnets harness DNS communication in order to make their Command and Control infrastructure more resilient and avoid detection. Their solution relied on seven factors, *domain reputation, geo location, destination port, known C&C, domain owner, frequent DNS changes and behavior*.Weights were assigned to each factor, the sum of which would produce a DNS flag, based on which an alarm would be raised. Furthermore, the seven factors were further partitioned into two groups, the must-stop factors and the partial-stop factors. If one must-stop factor or three partial-stop factors were detected, then a flag would be raised, indicating a suspicious domain record. The research team reported an accuracy of close to 63% when they tested their solution against similar results derived from the cyber security company FireEye.

  In the field of anomaly network IDS, Aldwairi *et al.* [4] investigated the applicability of Restricted Boltzman Machines (RBM), in order to distinguish between normal and abnormal flow traffic. An RBM is a special kind of neural network, where layers are either visible or hidden and two layers of the same type cannot be connected. In their experiments a balanced subset of the ISCX dataset was used to train the model. The algorithms that were used to train the RBM, were constructive and persistent constructive divergence. Results showed that RBMs are a valid choice for an anomalous network IDS with the capability to identify novel abnormal traffic. On the other hand, the performance of several supervised classification models for network IDS was investigated by Ugochukwu and Bennett [107]. Four classifiers were tested, Naive Bayes, C4.5, Random Forest and Random Tree. Out of the four classifiers, Random Forest and Random Tree outperformed the rest.

C. NETWORK FORENSIC METHODS FOR INVESTIGATING BOTNETS IN THE IOT

In this section, we focus on network forensics methods, that were designed to be applied in an IoT environment. One might consider, that pre-existing network forensics mechanisms could be employed in the IoT, with the same accuracy and efficiency as when applied to conventional computing systems. The fact is that the quantity and speed with which data is produced in the IoT, as-well-as its diversity, require the development of new methods which take under account these characteristics of the IoT. Popular methods and recent studies are explained as follows.

- **Honeypots** [30], [34], [39], [65], [84], [110]- which would be an ideal decoy for malware that target IoT devices was developed by Pa *et al.* [84], named IoT-POT. Their proposed system was a combination of a low-interaction front-end interaction program, and a high-interaction back end system, named IoTBOX which is a collection of virtual IoT machines (Linux OS), that help make the Honeypot appear as a legitimate, more dynamic device. In this design, they also incorporated a Profiler, which stored incoming "malicious"commands and the responses produced by IoTBOX. This would allow the system to produce the appropriate response in future interactions without invoking IoTBOX. Another module that was used was a Downloader, which managed all downloads prompted by the attackers, and a Manager which handled the configuration of the system. It is mentioned that the virtual environment required manual OS image resets from time to time, a process which could possibly be automated. It should be mentioned that, some malware include anti-forensics capabilities which can thwart attempts to scan them in a virtual environment, which was not discussed in this work.

With the intent to produce an initial framework for a high-interaction, seemingly geographically-distributed and vendor/type-of-device diverse Honeypot, Guarnizo *et al.* [34] proposed SIPHON. Their implementation allowed for the deployment of more than 80 high-interaction virtual IoT devices, with their IP addresses being distributed around the globe, and having only 7 physical devices exposed. The projected scalability of this system was reported to be i*w, with 'i' being the number of physical IoT devices and 'w' being the wormholes in use. Collected data from a two-month period, showed a significant amount of incoming attack traffic which targeted SSH services that were exposed by the experimental system. Additionally, the team noted that the proposed honeypot was not identified as such by Shodan, making this framework a viable solution.

An IoT-based honeypot, which focused on the emulation of an entire IoT platform was developed by Wang *et al.* [110], named ThingPot. Their design was an open source project, that could be characterized as a "Middle Interaction Honeypot"which made use of both high and low interaction modules. The proposed framework of ThingPot included three groups of entities, Extensible Messaging and Presence Protocol nodes (client, server) used for communication between "user"and Controller, REST API which represented the IoT device that ThingPot was mimicking, Controller which was represented as a PC that gathered log files from the other nodes in the setup. To test the applicability of ThingPot, an arrangement of devices which ran on Raspberry Pi, connected to a PC, and simulated Philips Hue lightbulbs was setup. This implementation was, as reported by the research team, a proof of concept that focused solely on the Philips Hue lightbulb IoT platform, with support for other such devices pending. Also, the research team's experiments ran for 1.5 months, a possible extension of this time could have yielded different results.

In the field of military network security and operations, Hanson *et al.* [39] introduced the concept of 'Honeyman', an IoT honeynet architecture that, instead of functioning as a mitigation and attack analysis platform, its primary functionality would be to provide indication and warning as-well-as distributed deception capabilities. The proposed system's function would primarily be to deceive attackers about the location and status of military IoT devices, whose goal was to corrupt their intelligence about either geo-spatial or system data. A multi-tier architecture was proposed, where a combination of light-weight devices would be used in tandem with virtualized machines and a software defined network, gathering data from the attacks. All collected information would then be forwarded to an RNN-based analysis module, where inferences could be made about the motivation of an adversary. Several difficulties hinder the development of such a system, with two being discussed in this research being: *securely emulate embedded os communications*, *avoiding detection of emulated environments*.

A server-based IoT honeypot system was proposed by Gandhi *et al.* [30] and named *HIoTPOT*. The proposed system relied on a Raspberry Pi to act as the 'middle-man ', and divert users with unknown credentials, with such attempts being recorded in a database, to a virtualized image of the real IoT devices. From there, alerts would be sent to legitimate users and logs would be created, that record the attacker's interaction with the environment. The proposed system was shown to provide more interactions as-well-as new mechanisms than an existing one, further comparing their performance and showing that HIoTPOT outperformed the existing solution in packet loss, consumption of bandwidth and added delays.

The problem of developing realistic IoT honeypots was addressed by Luo *et al.* [65]. The researchers proposed an automated method for crafting low but intelligent-interaction honeypots. To build such a honeypot, attack requests were collected by an initial honeypot, which were then forwarded to a specialized module that probed live IoT devices to get legitimate responses. In this way, the honeypot would be able to gather relevant responses from real devices. Machine learning would then be applied on the collected responses, in order to craft a 'profile 'that best represents the IoT device that the honeypot would mimic. The evaluation of the proposed method indicated an improvement in the functionality of the honeypot, extending its interaction time with attackers.

- **Network Flow Analysis** [26], [29], [72], [79]- the research team of Galluscio *et al.* [29], having the intent to clarify the severity and magnitude of IoT infected devices worldwide, worked from an empirical point of view. They utilized unsolicited darknet-generated data, which by definition, imply a potential malicious scan.

As such, and to be able to identify scanning (otherwise known as probing) activities, they developed an algorithm which utilized network flow features. The proposed algorithm compared the observed packet count and rate of a flow within a set time window, to pre-determined packet count and rate values. If observed values exceeded the pre-determined thresholds, then the flow was flagged as a malicious scan. As the research team was interested in IoT infected devices, they then made use Shodan, to infer whether the identified malicious probe originated from an IoT device, which would imply that that IoT device was compromised. Their findings showed that IP cameras and routers were the top two most heavily exploited household IoT devices, while sectors like Manufacturing and Building automation had the largest portions of exploited devices. The proposed algorithm, although fast is simplistic in its nature, with the assumption behind it being that benign Darknet IP devices don't perform Internet-wide scans.

Being able to identify consumer IoT devices that are part of network attacks is an important task. Towards that goal, Doshi *et al.* [26] worked on the identification of IoT devices which took part in launching DDoS attacks. Their approach was to utilize several characteristics of IoT-generated traffic that distinguishes it from other non-IoT traffic, such as the frequency of communication. As such, through an initial feature engineering process, the researchers trained five machine learning models, including a neural network, and concluded that, real-time detection of DDoS originating from IoT devices is possible.

Akin to the previous study, Meidan *et al.* [72] developed a method that identified botnet membership in commercial IoT devices. Their proposed method, was based on deep autoencoders, one for each of the nine IoT devices used in their experiments, with the life-cycle of their method being: *data collection, feature extraction, training of autoencoder model, continuous monitoring*. The principle behind their work was that IoT devices have a finite set of states, and as such, their autoencoder was trained to model normal traffic, flagging its errors as abnormal/bot activities. By testing their approach against real IoT botnets Mirai and Bashlite, they demonstrated a FPR of close to 0 for the majority of IoT devices under scrutiny. On a similar note, Nguyen *et al.* [79] developed DÎŹoT. In this system, data was first gathered from on-line IoT devices, and fingerprints of device communication were derived from them. This system, utilized an unsupervised technique to create clusters of fingerprints of IoT devices, and distinguish between different device types and models. An anomaly detection module, that implemented a k-Nearest Neighbors classifier, identified abnormal traffic, which worked as an indication of a compromised IoT device.

- **Intrusion Detection Systems** [3], [95]- Roux *et al.* [95], created an intrusion detection system for IoT, which focused on identifying potential attacks, based on their relative position in the environment which was monitored by this IDS. Their design made use of wireless sensors, strategically positioned around the premises (house), which were tasked with gathering information regarding signal strength and direction. The gathered information would then be forwarded to a central device, where it would be processed by a neural network that has been trained to identify legitimate transmissions from legitimate positions inside the network. Any transmission that originated from outside the network, would then be flagged as illegitimate, causing alarms to be triggered. Such a technique can be used to counteract war-driving and war flying, which can be employed to infect IoT devices with bot malware. A possible extension of its functionality, would be to use the flagging mechanism to automatically trigger forensic solutions for the IoT, when such illegitimate transmissions are identified.

  Although not explicitly stated, Al-Dabbagh *et al.* [3] proposed a framework for designing distributed IDSs of an IoT-like wireless control network. The proposed distributed IDS consisted of individual IDSs in each node/actuator in the control network, with the network itself modeled as a linear time invariant system. This allowed for the identification of cyber-attacks in a neighboring group of nodes of the network.

  Abhishek *et al.* [1] introduced a centralized IDS for IoT clusters. The researchers identified the gateway of an IoT cluster as a weak point, and thus they focused their efforts on monitoring the gateway. The novelty of the proposed IDS is that focus was placed on the downlink channel of the gateway. The proposed IDS identified malicious gateway attacks that sought to corrupt packet integrity, thus forcing retransmissions and taxing battery life. These attacks were identified by investigating the packet drop probability of the downlink between each IoT device and its gateway. In the future the researchers intend to extend their work by studying the uplink packets, as-well-as different types of attacks.

  The process of active learning for the development of IDS in wireless IoT networks was investigated by Yang *et al.* [112]. The concept of active learning relies on training a model on a small group of unlabeled data and periodical re-training, by requesting the missing labels of a record from a human operator. The proposed method relied on an initially detection of outliers by using an unsupervised outlier detector followed by the application of the active-learning-based scheme. During the active-learning-base learning process, first supervised learning was employed, followed by label selection and finally labeling by the expert operator, with the process being repeated until precision and recall reach appropriate values. Still challenges exist in this field which affect active-learning, such as the constrained power of such devices.

An ensemble NIDS was proposed by Moustafa *et al.* [76]. The researchers focused on identifying botnet attacks targeting the DNS, HTTP and MQTT protocols. Statistical methods were used on data, to produce additional features that improved the classification process. Feature selection was performed through the calculation of the Correlation Coefficient value between features. The AdaBoost ensemble method was implemented, using Decision Tree, Naive Bayes and Artificial Neural Network classifiers. Results showed that the proposed method outperformed equivalent existing ones in processing DNS and HTTP flows.

VI. DEEP LEARNING AND ITS ROLE IN NETWORK FORENSICS

Artificial Neural Networks (ANNs) which were inspired by the inner mechanics of the human brain, specifically the underline interconnected networks of neurons, are a type of machine learning technique which convert input data into output by employing non-linear transformations. ANNs can be roughly grouped by the number of layers that make up their architecture (excluding input layer), into textitshallow and *deep* [97]. Although there exist no strict definitions for them, a shallow ANN typically has one to two layers, while deep ANNs can have hundreds [98]. With the wide adoption of deep ANN architectures in various fields (e.g. computer vision, pattern recognition, ...), new specialized architectures have emerged.

Deep NNs can be further classified based on the way that they view the data and the classification problem, as given by Hodo *et al.* [40]. These two groups are *discriminative* and *generative* models. Discriminative models are supervised methods tasked with separating the data into classes by focusing on the decision boundary of the classes and calculating the conditional probability of the class feature, with respect to the data features (P(Y/X)). Prominent examples include:

- **Recurrent Neural Network (RNN)** - as a discriminative model, an RNN can be useful when the information maintains some temporal relation to its previous states.
- **Convolutional Neural Networks (CNN)** - is a type of space-invariant Multilayer Perceptron, inspired by the interconnections present in the visual cortex of the brain. It is comprised of multiple hidden layers such as: convolutional layers, pooling layers, fully connected layers and normalization layers.

On the other hand, generative models are considered to be unsupervised, as they do not require labeled data, and instead calculate the joint probability of data and class features (P(X,Y)) and build models that best describe each class separately. Some prominent examples of such models are described below.

- **Deep Auto Encoder (DAE)** - is a type of NN that is used to learn efficient data coding in an unsupervised manner. Typically, it includes an input layer, multiple hidden layers and an output layer of the same size as the input layer, where the input data is reconstructed.
- **Deep Boltzman Machine (DBM)** - produces binary results by relying on stochastic units and energy states. A restricted Boltzmann machine (RBM) is comprised of a visible input layer and a single hidden layer. By stacking multiple RBMs, so that the hidden layer of one produces the input for the next, one can build a DBM.
- **Deep Belief Network (DBN)** - are networks of interconnected layers comprised of multiple stacked RBMs. Again, connections between nodes of the same layer are not allowed, similar to DBMs. Training a DBN in an unsupervised manner requires for each layer to be greedily trained.
- **Recurrent Neural Network (RNN)** - is a type of deep NN that can be trained either as a supervised or unsupervised model. The main difference between RNN and a deep Multi-Layer Perceptron, would be that the RNN maintains an internal memory of previous calculations performed inside the network. Hidden layers of RNN 'feed 'information that is used for the next iteration of the algorithm.

Multiple deep learning solutions have been proposed for application in the field of Network Forensics in recent years [5], [52], [64], [82], [100], [114], [118]. Yin *et al.* [114], proposed a Recurrent Neural Network based IDS which outperformed other classifiers used for the same purpose. Similarly, Kang and Kang [52], proposed an IDS for a vehicle network capable of performing in real-time, with an average accuracy of 98%. In work by Zhao *et al.* [118], a Deep Belief Network was first applied, to reduce data dimensionality, followed by the training of a probabilistic neural network. Shone *et al.* [100] combined non-symmetrical auto-encoders with a random forest classifier to classify network traffic from the KDD99 and NSL-KDD dataset, with results indicating an increase in accuracy, when compared to DBNs.

Niyaz *et al.* [82] used stacked auto-encoders in their implementation of a DDoS detection system for software defined networks. The multiple auto-encoders were greedily trained layer-by-layer, with the output of one layer being the input of the next. Then the entire network was fine-tuned as a classifier. Reported accuracy for distinguishing between normal and attack traffic was 99.82%, outperforming other classification methods such as shallow NN, while individual types of DDoS attacks were identified with an accuracy of 95.65%. Lotfollahi *et al.* [64] used a combination of a one-dimensional CNN and stacked auto-encoders for automatic feature extraction and classification of network traffic, achieving both application identification and traffic characterization in either encrypted or unencrypted traffic.

VII. INHERENT CHALLENGES IN NETWORK FORENSIC INVESTIGATIONS OF IOT BOTNETS

The process of designing IoT protocols and sensors and the lack of standards are the main reasons why the IoT is an easy target for botnets. This gives rise to many challenges for experts who intend to investigate such security incidents.

TABLE 4. Most vulnerable communication Layers in IoT systems.

Communication layers	Description
Link layer.	It is for device discovery and local communications, at the link layer, the most widely used protocols include RFID, Wi-Fi, Bluetooth, ZigBee and ZWave [67], [92]. However, having in mind the global and, sometimes distributed nature of the IoT, the most significant protocols to mention, from the scope of a Network forensics investigator, would be the ones located at the network layer.
Network/Transport layer.	The most commonly used protocols are IP (v4, v6) [67], [92], which are widely used by most devices in the Internet, UDP, a Transport Layer protocol ideal for real time communications, with light-weight alternatives also in use, such as NanoIP [92], a TCP/IP -like stack of protocols, appropriate for embedded devices.
Application layer.	It is one of the most widely used protocols, include HTTPv2, REST and SOAP, which handle communications between applications (Client - Server systems).

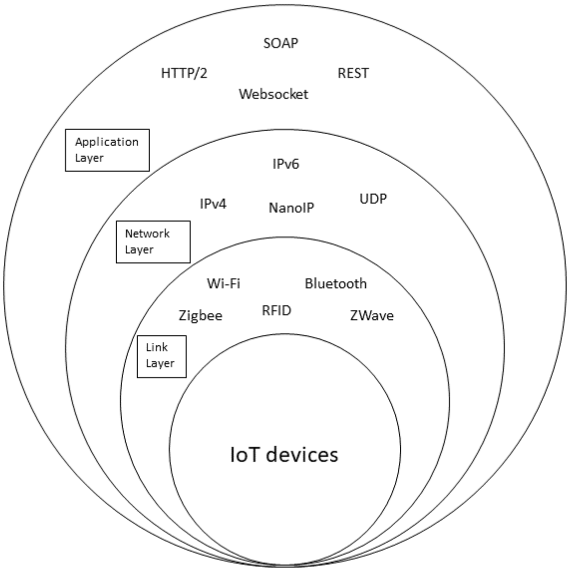

FIGURE 4. Architecture of IoT [67], [92].

We discuss the main challenges that could inhibit network forensic investigations of botnets in IoT systems, as explained in the following section.

- **Interoperability**- there are constraints in the interoperability of the IoT. As no single set of standards and specifications have been widely accepted [19], [46], every vendor implements their IoT products differently, choosing technologies, operating systems and protocols to serve the needs of their products, which often require a Sensor Bridge to co-exist, as shown in Figure 4. Moreover, we describe in Table 4 the most vulnerable communication layers and protocols in this architecture. It is obvious that the lack of specifications causes problems to the development of a single forensics solution that is capable of handling a family of IoT systems and devices.
- **Availability**- another harmful consequence of an IoT Botnet, is the depletion of the already constrained resources of deployed devices. As such, IoT-enabled services may exhibit a drop in performance or become entirely unavailable, which can be considered as a type of DoS attack. As such, businesses and industries relying on the constant function of these devices, or even the manufacturers that produced the hijacked IoT devices can be considered liable for any loss of service [46].
- **Cloud storage of information**- locating the evidence in an IoT Botnet-related security incident can also be challenging. In most implementations of the IoT, low-power physical devices are used as actuators, with local hubs and network nodes employed to gather and transport the collected information to a central Cloud Service provider. Through these Cloud Service providers, IoT services become available to users [41], as shown in Figure 1. With this scheme in mind and knowing that actuators (the intelligent "things") are equipped with a limited amount of memory and power, data is quickly gathered and transferred to the Cloud, freeing up space in the actuators for further tasks to take place. As such, evidence will most probably be found in the Cloud, which introduces a new family of challenges to forensic investigations, among which jurisdiction limitations and conflicting laws are two prominent examples.
- **Forensic soundness**- with the IoT designed to work in an autonomous and ubiquitous form, following a forensically sound process becomes a challenge [22]. Preserving the scene of a crime where IoT devices are involved is challenging, as data is in constant motion and the scope of investigation is not clear. There is a lack of documented methods and reliable tools for collecting evidence in a forensically sound manner. As most IoT devices don't retain metadata that can indicate alterations or manipulations of files, and the time when these changes occurred, correlation of evidence between IoT devices is challenging. Finally, without a forensically sound monitoring system, attribution becomes difficult.
- **Big Data characteristics generated from IoT systems**- some of the challenges present in investigating the IoT, coincide with the main characteristics of Big Data, indicating that the latter technology could be a possible approach of handling these challenges [19]. These Big Data characteristics are, Variety, Velocity, Volume, Value and Veracity as discussed below.
  - ▹ **Variety**- data produced by the IoT may exist in either structured (database tables), semi-structured (XML, JSON) or unstructured (audiovisual files) form, depending on the type of device in question [19], [22]. At the same time, data produced by a single IoT system may be thematically heterogeneous. For example, in an automated

home, a thermostat and some motion sensors can be connected to each other, so that when the motion sensors detect motion, the thermostat sets the room temperature to a preset value. Thus, with the possibility that in a single IoT system there can exist any combination of heterogeneous devices (Web cameras, digital locks, routers, thermostats), produced data, and thus also evidence, will vary in format, making it necessary for a process capable of scanning diverse data and locating traces a necessity.

- ▹ **Velocity**- the increases in Internet speed over the years, ensures that communications are nigh-instantaneous, which functions as one of the enabling factors of the IoT. This translates to a large number of constantly functioning, small sensors and actuators ('smart things') sending collected data and feedback, at high speeds, to the service providers (usually located in the Cloud), which in the context of IoT botnets means that such botnets will possess an army of high-speed and always available bots. With data and evidence produced fast, and having a relatively short life in the network, a need to analyze data in real time (or as near as possible) is essential, if the results produced by investigations are to be of any real value.
- ▹ **Volume**- with more than 8 billion IoT devices deployed in 2017 and projections for the near future rising even higher, it is evident that data produced by the IoT will also skyrocket. As such, having many small embedded devices in constant use produces huge quantities of data that, in the context of forensics (regardless of the type of investigation), will inhibit the effectiveness of investigations, burying useful traces and evidence under a sea of noise (in the form of collected data). On top of that, increasing number of deployed devices equates to an enhancement of numbers for potential hijacked Bots, allowing adversaries to take advantage of the sheer volume of data that the IoT can produce and thus launch massive and reliable Cyber-attacks (examples Mirai [59]).
- ▹ **Value**- with IoT introduced in several sectors of everyday life, such as home automation, the health domain and more, concerns about privacy arise [19], [22], as sensitive information is recorded, exchanged, maintained and stored by the IoT devices and their service providers. As such, forensics investigations need to be conducted with a level of transparency and steps must be taken to ensure that private data are not exploited.
- ▹ **Veracity**- deployed in a dynamic world, it is easy for environmental conditions to change inexplicably, causing the validity of recordings from finely calibrated IoT sensors to be faulty and producing inaccurate, low quality or noisy data [19]. As such, identifying such problems with the collected data is a challenge, as 'contaminated' data could lead to false results in an investigation.

VIII. FUTURE DIRECTIONS OF RESEARCH

In this section, we discuss research directions for future work based on existing challenges of investigating botnets using forensics mechanism. Having reviewed some of the work conducted in the discipline of network forensics, initially with regards to botnets in general, and then narrowing down our search for solutions applied to the Internet of Things, as explained in the following section.

- **Honeypot development**- With some work already done in the field of building convincing Honeypots specifically targeting IoT-related adversaries [34], [84], [110], it is expected that further advances will be made. Some ways of enhancing Honeypot implementations might include, making them more resilient against anti-forensics mechanisms, increasing the number of supported protocols thus increasing the range of mimicked IoT devices and handling the massive quantities of incoming traffic which could be generated by an IoT Botnet.
- **Network Flow Analysis**- Although Network Flow Analysis can be implemented in the context of IoT Botnet investigations [29], such methods are still being developed. More work is needed in this area, as it is an easy to implement and non-intrusive way of utilizing real world data, without the fear of privacy violation, it requires less space for saved data compared to other solutions DPI) and is resilient to encryptions of the payload.
- **Providing Forensic Soundness**- Having presented a number of diverse ways by which researchers conducted network forensic investigations of botnets (including IoT botnets), one topic that wasn't discussed much, was proving that the process they provided was forensically sound. In other words, a forensic process that is used by law enforcement, needs to produce results that would be admissible in a court of law [22]. As such, in the future, steps need to be taken in order to consider how new techniques can be enhanced to produce acceptable forensic results.
- **Dealing with Variety, Velocity and Volume of IoT data**- Although the field of network forensics as applied to the Internet of Things is still in development, by reviewing research done on this conjunction of fields (network forensics of IoT-botnets) it was observed that not much emphasis was given on dealing with the following problematic issues: its polymorphic nature, the sheer quantity and speed with which information is recorded and transmitted. These characteristics of the IoT make applying network forensic techniques established in conventional IT systems inapplicable, thus directing future research towards dealing with these challenges [19].

IX. CONCLUSION

In this review paper, we explored the effects that the expanding IoT domain has had in Network Forensic Investigations of IoT botnets. We initially provided background of the Internet of Things, botnets and Digital Forensics, as a foundation. We give a new definition for the IoT, which places the interconnection of "Things "and their service-like functionality in the forefront. We argue that Deep Learning is a viable solution to handling the types of data produced in the IoT, and thus discuss its applicability in Network Forensics. Furthermore, we provided a taxonomy of Network Forensic mechanisms which could be applied to botnets in both non-IoT and IoT environments, including their strengths and weaknesses. The Network Forensic mechanisms that were discussed, were Honeypots, Network Flow Analysis, Deep Packet Analysis, Attack Recognition, Visualization of Network Traffic and Intrusion Detection Systems. Several challenges were presented, including regional jurisdiction issues derived from Cloud computing, providing forensics soundness from short lived traces and evidence and interoperability issues due to lack of standards. Finally, future directions for research in the area were discussed. Some such directions were: development and improvement of Honeypots and Network Flow analysis, handling vast quantities of high-speed and heterogeneous data produced by the IoT, and proving that any produced solutions are Forensically sound and results produced would be admissible in a court of law.

REFERENCES

[1] N. V. Abhishek, T. J. Lim, B. Sikdar, and A. Tandon, "An intrusion detection system for detecting compromised gateways in clustered iot networks," in *Proc. IEEE Int. Workshop Tech. Committee Commun. Qual. Rel. (CQR)*, May 2018, pp. 1–6.

[2] M. U. Ahmed, M. Björkman, A. Čaušević, H. Fotouhi, and M. Lindén, "An overview on the Internet of Things for health monitoring systems," in *Proc. Int. Internet Things Summit*. Cham, Switzerland: Springer, 2015, pp. 429–436.

[3] A. W. Al-Dabbagh, Y. Li, and T. Chen, "An intrusion detection system for cyber attacks in wireless networked control systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 8, pp. 1049–1053, Aug. 2018.

[4] T. Aldwairi, D. Perera, and M. A. Novotny, "An evaluation of the performance of restricted Boltzmann machines as a model for anomaly network intrusion detection," *Comput. Netw.*, vol. 144, pp. 111–119, Oct. 2018.

[5] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2016, pp. 195–200.

[6] K. AlRoum, A. Alolama, R. Kamel, M. El Barachi, and M. Aldwairi, "Detecting malware domains: A cyber-threat alarm system," in *Proc. Int. Conf. Emerg. Technol. Developing Countries*. Cham, Switzerland: Springer, 2017, pp. 181–191.

[7] P. Amini, M. A. Araghizadeh, and R. Azmi, "A survey on botnet: Classification, detection and defense," in *Proc. Int. Electron. Symp. (IES)*, Sep. 2015, pp. 233–238.

[8] B. Anchit and S. Harvinder, "Investigation of UDP Bot flooding attack," *Indian J. Sci. Technol.*, vol. 9, no. 21, pp. 1–6, 2016.

[9] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos, "Highly resilient peer-to-peer botnets are here: An analysis of gameover zeus," in *Proc. 8th Int. Conf. Malicious Unwanted Softw., Americas (MALWARE)*, Oct. 2013, pp. 116–123.

[10] K. Angrishi. (2017). "Turning Internet of Things(IoT) into Internet of vulnerabilities (IoV): IoT botnets." [Online]. Available: https://arxiv.org/abs/1702.03681

[11] K. Ashton *et al.*, "That 'Internet of Things' ThingThing, in the real world things matter more than ideas," *RFID J.*, vol. 22, no. 7, pp. 97–114, 2009.

[12] S. Bansal, M. Qaiser, S. Khatri, and A. Bijalwan, "Botnet forensics framework: Is your system a bot," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Eng. (ICACCE)*, May 2015, pp. 535–540.

[13] V. Baryamureeba and F. Tushabe, "The enhanced digital investigation process model," in *Proc. 4th Digit. Forensic Res. Workshop*, 2004, pp. 1–9.

[14] A. Bijalwan, M. Thapaliyal, E. S. Pilli, and R. C. Joshi, "Survey and research challenges of botnet forensics," *Int. J. Comput. Appl.*, vol. 75, no. 7, pp. 43–50, 2013.

[15] A. Bijalwan, M. Wazid, E. S. Pilli, and R. C. Joshi, "Forensics of random-UDP flooding attacks," *J. Netw.*, vol. 10, no. 5, pp. 287–293, 2015.

[16] E. Bou-Harb and M. Scanlon, "Behavioral service graphs: A formal data-driven approach for prompt investigation of enterprise and Internet-wide infections," *Digit. Invest.*, vol. 20, pp. S47–S55, Mar. 2017.

[17] M. Brand, C. Valli, and A. Woodward, "Malware forensics: Discovery of the intent of deception," *J. Digit. Forensics, Secur. Law*, vol. 5, no. 4, p. 2, 2010.

[18] *Internet of Things, Strategic Research Roadmap*, European Commission, Brussels, Belgium, 2009.

[19] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data mining for the Internet of Things: Literature review and challenges," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 8, p. 431047, 2015.

[20] Z. Chen, F. Han, J. Cao, X. Jiang, and S. Chen, "Cloud computing-based forensic analysis for collaborative network security management system," *Tsinghua Sci. Technol.*, vol. 18, no. 1, pp. 40–50, Feb. 2013.

[21] R. Cheng and G. Watson, "D^{2}PI: Identifying malware through deep packet inspection with deep learning," Tech. Rep., 2018.

[22] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 78, pp. 544–546, Jan. 2018.

[23] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[24] D. M. Divakaran, K. W. Fok, I. Nevat, and V. L. L. Thing, "Evidence gathering for network security and forensics," *Digit. Invest.*, vol. 20, pp. S56–S65, Mar. 2017.

[25] F. Donovan, "A brief history of the Internet of Things," Tech. Rep., 2014.

[26] R. Doshi, N. Apthorpe, and N. Feamster. (2018). "Machine learning DDoS detection for consumer Internet of Things devices." [Online]. Available: https://arxiv.org/abs/1804.04159

[27] P. Farina, E. Cambiaso, G. Papaleo, and M. Aiello, "Are mobile botnets a possible threat? The case of SlowBot net," *Comput. Secur.*, vol. 58, pp. 268–283, May 2016.

[28] J. François, S. Wang, W. Bronzi, R. State, and T. Engel, "Botcloud: Detecting botnets using mapreduce," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Nov./Dec. 2011, pp. 1–6.

[29] M. Galluscio *et al.*, "A first empirical look on Internet-scale exploitations of IoT devices," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–7.

[30] U. D. Gandhi, P. M. Kumar, R. Varatharajan, G. Manogaran, R. Sundarasekar, and S. Kadu, "HIoTPOT: Surveillance on IoT devices against recent threats," *Wireless Pers. Commun.*, vol. 103, no. 2, pp. 1179–1194, 2018.

[31] S. García, A. Zunino, and M. Campo, "Survey on network-based botnet detection methods," *Secur. Commun. Netw.*, vol. 7, no. 5, pp. 878–903, 2014.

[32] I. Ghafir *et al.*, "BotDet: A system for real time botnet command and control traffic detection," *IEEE Access*, vol. 6, pp. 38947–38958, 2018.

[33] R. B. Gilbert, *Defending Against Malicious Software*. Santa Barbara, CA, USA: Univ. of California, 2011.

[34] J. D. Guarnizo *et al.*, "Siphon: Towards scalable high-interaction physical honeypots," in *Proc. 3rd ACM Workshop Cyber-Phys. Syst. Secur.*, 2017, pp. 57–68.

[35] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[36] D. Gugelmann, F. Gasser, B. Ager, and V. Lenders, "Hviz: HTTP(S) traffic aggregation and visualization for network forensics," *Digit. Invest.*, vol. 12, pp. S1–S11, Mar. 2015.

[37] S. Haller, S. Karnouskos, and C. Schroth, "The Internet of Things in an enterprise context," in *Future Internet Symp.* Heidelberg, Germany: Springer, 2008, pp. 14–28.

[38] F. Han, Z. Chen, H. Xu, H. Wang, and Y. Liang, "A collaborative botnets suppression system based on overlay network," *Int. J. Secur. Netw.*, vol. 7, no. 4, pp. 211–219, 2012.

[39] P. J. Hanson, L. Truax, and D. D. Saranchak, "IoT honeynet for military deception and indications and warnings," *Proc. SPIE*, vol. 10643, May 2018, Art. no. 106431A.

[40] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson. (2017). "Shallow and deep networks intrusion detection system: A taxonomy and survey." [Online]. Available: https://arxiv.org/abs/1701.02145

[41] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of Things," in *Proc. IEEE World Congr. Services (SERVICES)*, Jun./Jul. 2015, pp. 21–28.

[42] R. Hunt and S. Zeadally, "Network forensics: An analysis of techniques, tools, and trends," *Computer*, vol. 45, no. 12, pp. 36–43, Dec. 2012.

[43] E. Husni, G. B. Hertantyo, D. W. Wicaksono, F. C. Hasibuan, A. U. Rahayu, and M. A. Triawan, "Applied Internet of Things (IoT): Car monitoring system using IBM BlueMix," in *Proc. Int. Seminar Intell. Technol. Appl. (ISITIA)*, Jul. 2016, pp. 417–422.

[44] T. S. Hyslip and J. M. Pittman, "A survey of botnet detection techniques by command and control infrastructure," *J. Digit. Forensics, Secur. Law*, vol. 10, no. 1, p. 2, 2015.

[45] "Mariposa botnet analysis," Defence Intell., Kanata, ON, Canada, Tech. Rep., Oct. 2009. [Online]. Available: https://defintel.com/docs/Mariposa_Analysis.pdf

[46] IoTAA. (Feb. 2017). *Internet of Things Alliance Australia Internet of Things Security Guideline v1.0*. [Online]. Available: http://www.iot.org.au/wp/wp-content/uploads/2016/12/IoTAA-Security-Guideline-V1.0.pdf

[47] Z. Ismail and A. Jantan, "A review of machine learning application in botnet detection system," *Sindh Univ. Res. J.-SURJ (Sci. Ser.)*, vol. 48, no. 4D, pp. 111–118, 2016.

[48] J. Jeong, S. M. A. Naqvi, and M. Yoon, "Accurate and communication-efficient detection of widespread events," *IEEE Access*, vol. 6, pp. 61728–61734, 2018.

[49] A. Joshi and D. S. Bhilare, "Digital forensics: Emerging trends and analysis of counter-security environment," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 12, pp. 1116–1120, 2013.

[50] C. Joslyn, S. Choudhury, D. Haglin, B. Howe, B. Nickless, and B. Olsen, "Massive scale cyber traffic analysis: A driver for graph database research," in *Proc. 1st Int. Workshop Graph Data Manage. Exper. Syst.*, 2013, p. 3.

[51] M. E. Kabay. (Mar. 2009). *Kraken the Botnet: The Ethics of Counter-Hacking*. Network World, Inc. Accessed: Mar. 14, 2016. [Online]. Available: https://search-proquest-com.wwwproxy1.library.unsw.edu.au/docview/223736574?accountid=12763

[52] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS ONE*, vol. 11, no. 6, 2016, Art. no. e0155781.

[53] K. S. Karthika, "Peer to peer botnet detection system," in *Proc. Int. Conf. Inf. Image Process. (ICIIP)*, 2014, pp. 196–197.

[54] R. Kaur and A. Kaur, "Digital forensics," *Int. J. Comput. Appl.*, vol. 50, no. 5, pp. 5–9, 2012.

[55] V. R. Kebande and I. Ray, "A generic digital forensic investigation framework for internet of things (IoT)," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2016, pp. 356–362.

[56] M. Keshk, N. Moustafa, E. Sitnikova, and G. Creech. (2017). "Privacy preservation intrusion detection technique for SCADA systems." [Online]. Available: https://arxiv.org/abs/1711.02828

[57] S. Khan, A. Gani, A. W. A. Wahab, M. Shiraz, and I. Ahmad, "Network forensics: Review, taxonomy, and open challenges," *J. Netw. Comput. Appl.*, vol. 66, pp. 214–235, May 2016.

[58] S. Khattak, N. R. Ramay, K. R. Khan, A. A. Syed, and S. A. Khayam, "A taxonomy of botnet behavior, detection, and defense," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 898–924, 2nd Quart., 2014.

[59] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[60] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay. (2017). "Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques." [Online]. Available: https://arxiv.org/abs/1711.02825

[61] R. Kozik, "Distributing extreme learning machines with apache spark for netflow-based malware activity detection," *Pattern Recognit. Lett.*, vol. 101, pp. 14–20, Jan. 2018.

[62] S. Kumar, P. Singh, R. Sehgal, and J. S. Bhatia, "Distributed honeynet system using gen III virtual honeynet," *Int. J. Comput. Theory Eng.*, vol. 4, no. 4, pp. 537–541, 2012.

[63] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu, "Security vulnerabilities of Internet of Things: A case study of the smart plug system," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1899–1909, Dec. 2017.

[64] M. Lotfollahi, R. S. H. Zade, M. J. Siavoshani, and M. Saberian. (2017). "Deep packet: A novel approach for encrypted traffic classification using deep learning." [Online]. Available: https://arxiv.org/abs/1709.02656

[65] T. Luo, Z. Xu, X. Jin, Y. Jia, and X. Ouyang, "IoTCandyJar: Towards an intelligent-interaction honeypot for IoT devices," in *Proc. Black Hat*, 2017, pp. 1–11.

[66] A. MacDermott, T. Baker, and Q. Shi, "IoT forensics: Challenges for the IoA era," in *Proc. 9th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Feb. 2018, pp. 1–5.

[67] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A literature review," *J. Comput. Commun.*, vol. 3, no. 3, pp. 164–173, 2015.

[68] K. N. Mallikarjunan, K. Muthupriya, and S. M. Shalinie, "A survey of distributed denial of service attack," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–6.

[69] M. McKeay *et al.*, "Q4 2016 state of the Internet/security report," *Akamai Technol.*, vol. 3, no. 4, 2017, pp. 1–25.

[70] L. Mathur, M. Raheja, and P. Ahlawat, "Botnet detection via mining of network traffic flow," *Procedia Comput. Sci.*, vol. 132, pp. 1668–1677, Dec. 2018.

[71] J. McCallion. (Oct. 14, 2015). *Dell, FBI and NCA Bring Down Botnet Behind 20m Cyber Bank Heist*. Accessed: Oct. 10, 2015. [Online]. Available: https://search.proquest.com/docview/1722048861?accountid=12763

[72] Y. Meidan *et al.*, "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul./Sep. 2018.

[73] S. Mittal and R. Singh, "A support vector approach for formulating IDS rules using honeypot data," *Adv. J. Comput. Sci. Eng.*, vol. 4, pp. 1–5, Jun. 2016.

[74] N. Moustafa and J. Slay. (2017). "RCNF: Real-time collaborative network forensic scheme for evidence analysis." [Online]. Available: https://arxiv.org/abs/1711.02824

[75] N. Moustafa, J. Slay, and G. Creech, "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks," *IEEE Trans. Big Data*, to be published.

[76] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things," *IEEE Internet Things J.*, to be published.

[77] N. Naik, P. Jenkins, R. Cooke, and L. Yang, "Honeypots that bite back: A fuzzy technique for identifying and inhibiting fingerprinting attacks on low interaction honeypots," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2018, pp. 1–8.

[78] N. Negash and X. Che, "An overview of modern botnets," *Inf. Secur. J., Global Perspective*, vol. 24, nos. 4–6, pp. 127–132, 2015.

[79] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi. (2018). "DÏoT: A federated self-learning anomaly detection system for IoT." [Online]. Available: https://arxiv.org/abs/1804.07474

[80] R. Nigam, "A timeline of mobile botnets," *Virus Bull.*, pp. 1–23, Mar. 2015.

[81] A. Nisioti, A. Mylonas, P. D. Yoo, and V. Katos, "From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3369–3388, 4th Quart., 2018.

[82] Q. Niyaz, W. Sun, and A. Y. Javaid. (2016). "A deep learning based DDoS detection system in software-defined networking (SDN)." [Online]. Available: https://arxiv.org/abs/1611.07400

[83] V. Oujezsky, T. Horvath, and V. Skorpil, "Botnet C&C traffic and flow lifespans using survival analysis," *Int. J. Adv. Telecommun., Electrotechnics, Signals Syst.*, vol. 6, no. 1, pp. 38–44, 2017.

[84] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: Analysing the rise of IoT compromises," *Proc. EMU*, 2015, pp. 1–9.

[85] Hewlett Packard. (2015). *Internet of Things Research Study–2015 Report*, vol. 2. [Online]. Available: http://www8.hp.com/h20195

[86] G. Palmer, "A road map for digital forensics research-report from the first digital forensics research workshop (DFRWS)," in *Proc. Digit. Forensic Res. Conf.*, Utica, NY, USA, 2001. [Online]. Available: http://dfrws.org/sites/default/files/session-files/a_road_map_for_digital_forensic_research.pdf

[87] A. Paradise, D. Cohen, A. Shabtai, and R. Puzis. (2018). "Generation of automatic and realistic artificial profiles." [Online]. Available: https://arxiv.org/abs/1807.00125

[88] A. Pektaş and T. Acarman, "Botnet detection based on network flow summary and deep learning," *Int. J. Netw. Manage.*, vol. 28, no. 6, 2018, Art. no. e2039.

[89] I. Peña-López *et al.*, "ITU Internet report 2005: The Internet of Things," Int. Telecommun. Union, Genèva, Switzerland, Internet Rep. 7, 2005.

[90] V.-H. Pham and M. Dacier, "Honeypot trace forensics: The observation viewpoint matters," *Future Gener. Comput. Syst.*, vol. 27, no. 5, pp. 539–546, 2011.

[91] M. Pollitt, "Computer forensics: An approach to evidence in cyberspace," in *Proc. Nat. Inf. Syst. Secur. Conf.*, vol. 2, 1995, pp. 487–491.

[92] Postscape. (2018). *IoT Standards & Protocols Guide | 2018 Comparisons on Network, Wireless Comms, Security, Industrial*. Accessed: Apr. 2, 2018. [Online]. Available: https://www.postscapes.com/internet-of-things-protocols/

[93] E. Ronen, A. Shamir, A.-O. Weingarten, and C. O'Flynn, "IoT Goes nuclear: Creating a ZigBee chain reaction," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 195–212.

[94] K. Rose, S. Eldridge, and L. Chapin, "The Internet of Things: An overview," in *Proc. Internet Soc. (ISOC)*, 2015, pp. 1–50.

[95] J. Roux, E. Alata, G. Auriol, V. Nicomette, and M. Kâaniche, "Toward an intrusion detection approach for IoT based on radio communications profiling," in *Proc. 13th Eur. Dependable Comput. Conf.*, Sep. 2017, pp. 147–150.

[96] K. Ruan, J. Carthy, T. Kechadi, and I. Baggili, "Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results," *Digit. Invest.*, vol. 10, no. 1, pp. 34–43, 2013.

[97] M. Saber, I. El Farissi, S. Chadli, M. Emharraf, and M. G. Belkasmi, "Performance analysis of an intrusion detection systems based of artificial neural network," in *Proc. Eur. MENA Cooperation Adv. Inf. Commun. Technol.* Cham, Switzerland: Springer, 2017, pp. 511–521.

[98] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[99] SearchSecurity. (2018). *Get the Best Botnet Protection With the Right Array of Tools*. Accessed: May 2, 2018. [Online]. Available: http://searchsecurity.techtarget.com/feature/Get-the-best-botnet-protection-with-the-right-array-of-tools?src=5711374&asrc=EM_ERU_88560916&utm_content=eru-rd2-rcpG&utm_medium=EM&utm_source=ERU&utm_campaign=20180129_ERU%20Transmission%20for%2001/29/2018%20

[100] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[101] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles, "Botnets: A survey," *Comput. Netw.*, vol. 57, no. 2, pp. 378–403, 2013.

[102] H. Singh and A. Bijalwan, "A survey on malware, botnets and their detection," *Int. J. Adv. Eng. Res. Sci.*, vol. 3, no. 3, pp. 85–90, 2016.

[103] A. Sivaprasad, N. Ghawalkar, S. Hodge, M. Sanghavi, and V. Shinde, "Machine learning based traffic classification using statistical analysis," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 6, no. 3, pp. 187–191, 2018.

[104] S. Soltani, S. A. H. Seno, M. Nezhadkamali, and R. Budiarto, "A survey on real world botnets and detection mechanisms," *Int. J. Inf. Netw. Secur.*, vol. 3, no. 2, pp. 116–127, 2014.

[105] P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy, "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment," in *Proc. Int. Conf. Sci. Eng. Manage. Res. (ICSEMR)*, Nov. 2014, pp. 1–8.

[106] T. Truong, A. Dinh, and K. Wahid, "An IoT environmental data collection system for fungal detection in crop fields," in *Proc. IEEE 30th Can. Conf. Elect. Comput. Eng. (CCECE)*, Apr./May 2017, pp. 1–4.

[107] C. J. Ugochukwu and E. O. Bennett, "An intrusion detection system using machine learning algorithm," *Int. J. Comput. Sci. Math. Theory*, vol. 4, no. 1, pp. 2545–5699, 2018.

[108] R. van der Meulen. (Feb. 2017). *Gartner Says 8.4 Billion Connected 'Things' will be in use in 2017, up 31 Percent From 2016*. Accessed: Feb. 8, 2017. [Online]. Available: https://search.proquest.com/docview/1865709412?accountid=12763

[109] A. Verma, M. S. Rao, A. K. Gupta, W. Jeberson, and V. Singh, "A literature review on malware and its analysis," *Int. J. Current Res. Rev.*, vol. 5, no. 16, pp. 71–82, 2013.

[110] M. Wang, J. Santillan, and F. Kuipers, "ThingPot: An interactive Internet-of-Things honeypot," *Joint*, to be published.

[111] C. Xiang, F. Binxing, Y. Lihua, L. Xiaoyi, and Z. Tianning, "Andbot: Towards advanced mobile botnets," in *Proc. 4th USENIX Conf. Large-Scale Exploits Emergent Threats*, 2011, p. 11.

[112] K. Yang, J. Ren, Y. Zhu, and W. Zhang. (2018). "Active learning for wireless IoT intrusion detection." [Online]. Available: https://arxiv.org/abs/1808.01412

[113] I. Yaqoob *et al.*, "The rise of ransomware and emerging security challenges in the Internet of Things," *Comput. Netw.*, vol. 129, pp. 444–458, Dec. 2017.

[114] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[115] C. Yin, H. Wang, X. Yin, R. Sun, and J. Wang, "Improved deep packet inspection in data stream detection," *J. Supercomput.*, vol. 74, pp. 1–14, Nov. 2018. doi: 10.1007/s11227-018-2685-y.

[116] M. A. M. Yusof, F. H. M. Ali, and M. Y. Darus, "Detection and defense algorithms of different types of DDoS attacks," *Int. J. Eng. Technol.*, vol. 9, no. 5, pp. 410–414, 2017.

[117] Y. Yusoff, R. Ismail, and Z. Hassan, "Common phases of computer forensics investigation models," *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 3, pp. 17–31, 2011.

[118] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE)*, vol. 1, Jul. 2017, pp. 639–642.

[119] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1606–1616, Apr. 2019.

[120] Y. Zhu, "Attack pattern discovery in forensic investigation of network attacks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 7, pp. 1349–1357, Aug. 2011.

NICKOLAOS KORONIOTIS was born in Athens, Greece, in 1992. He received the B.S. degree in informatics and telematics and the M.S. degree in web engineering and applications from the Harokopio University of Athens, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree in cyber security with the School of Engineering and Information Technology (SEIT), University of New South Wales (UNSW) Canberra, Australia, with particular interests in network forensics and the IoT.

NOUR MOUSTAFA received the bachelor's and master's degrees in information systems from Helwan University, Egypt, in 2009 and 2014, respectively, and the Ph.D. degree in cyber-security from University of New South Wales (UNSW) Canberra, Australia, in 2017. He is currently a Lecturer in cyber-security with the School of Engineering and Information Technology (SEIT), (UNSW) Canberra. His research interests include cyber-security, in particular, network security, intrusion detection, threat intelligence, privacy-preserving, and digital forensics for Indystry 4.0, the Internet of Things (IoT), cloud, and fog computing. His methodologies include statistical learning, machine/deep learning, big data analytics, and artificial intelligence (AI) planning.

ELENA SITNIKOVA received the Honors degree in electrical engineering and the Ph.D. degree in communication control systems. She is currently an academic and researcher and the Program Coordinator for the Master in Cyber Security Program with the University of New South Wales (UNSW) Canberra. Her current research interests include the critical infrastructure protection area, carrying out research projects in the area of intrusion detection (IDS) for control systems cyber security, and the Industrial IoT (IIoT). She is one of the first Australians to be certified in CSSLP - Certified Secure Software Lifecycle Professional. She is an award winning academic, and received the Senior Fellowship of the Higher Education Academy (SFHEA) and the national Australian Office for Learning and Teaching (OLT) Team Citation award for Outstanding Contributions to Student Learning.