IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 25, 2020, accepted November 27, 2020, date of publication December 2, 2020, date of current version December 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041951

A Survey on Machine Learning Techniques for Cyber Security in the Last Decade

KAMRAN SHAUKAT[1,4], SUHUAI LUO[1], VIJAY VARADHARAJAN[1], (Senior Member, IEEE), IBRAHIM A. HAMEED[2], (Senior Member, IEEE), AND MIN XU[3], (Member, IEEE)

[1]School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW 2308, Australia
[2]Department of ICT and Natural Sciences, Norwegian University of Science and Technology, 7491 Trondheim, Norway
[3]School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia
[4]Punjab University College of Information Technology, University of the Punjab, Lahore 54590, Pakistan

Corresponding authors: Kamran Shaukat (kamran.shaukat@uon.edu.au) and Ibrahim A. Hameed (ibib@ntnu.no)

This work was supported by the Norwegian University of Science and Technology, Norway.

ABSTRACT Pervasive growth and usage of the Internet and mobile applications have expanded cyberspace. The cyberspace has become more vulnerable to automated and prolonged cyberattacks. Cyber security techniques provide enhancements in security measures to detect and react against cyberattacks. The previously used security systems are no longer sufficient because cybercriminals are smart enough to evade conventional security systems. Conventional security systems lack efficiency in detecting previously unseen and polymorphic security attacks. Machine learning (ML) techniques are playing a vital role in numerous applications of cyber security. However, despite the ongoing success, there are significant challenges in ensuring the trustworthiness of ML systems. There are incentivized malicious adversaries present in the cyberspace that are willing to game and exploit such ML vulnerabilities. This paper aims to provide a comprehensive overview of the challenges that ML techniques face in protecting cyberspace against attacks, by presenting a literature on ML techniques for cyber security including intrusion detection, spam detection, and malware detection on computer networks and mobile networks in the last decade. It also provides brief descriptions of each ML method, frequently used security datasets, essential ML tools, and evaluation metrics to evaluate a classification model. It finally discusses the challenges of using ML techniques in cyber security. This paper provides the latest extensive bibliography and the current trends of ML in cyber security.

INDEX TERMS Cyber security, deep learning, intrusion detection, malware, machine learning, spam.

I. INTRODUCTION

The Internet is increasingly becoming a widely utilized source of both information and (online) services. There is rapid growth in Internet usage: in 2017, about 48% of the total world population used the Internet as a source of information [1]. This figure increased up to 81% in developed countries [2]. The primary purpose of the Internet is to transport data from one node to another over the network. Internet is a universal collection of millions of distinct interconnected computers, networks, and associated devices. The innovation of computer systems, networks, and mobile devices has dramatically increased the usage of the Internet. Consequently, the Internet has become the target of cybercriminals and enemies [3].

A secure and stable computer system must ensure the confidentiality, availability, and integrity of information. The integrity and security of a computer system are compromised when an illegal penetration, unauthorized individual or program enters a computer or network intending to harm or disrupt the normal flow of activities [4]. Cyber security is the set of security measures that can be taken to protect the cyberspace and user assets against unauthorized access and attacks. The main objective of a cyber defence system is that data should be confidential, integral, and available [5].

National defence plays a crucial role in the integrity of any country. Computer networks are (or should be) designed to provide controls, which allow only authorised persons to access data. Bush Administration started the Comprehensive National Cyber Security Initiative (CNCSI) in January 2008 [6]. The purposes of the initiative were to highlight several issues for instance identification of current and evolving cyber security threats, finding and plugging existing cyber vulnerabilities, and apprehending actors that were trying to gain access to secure federal information systems. The next president of the United States, president Obama continued

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

it and declared that the 'cyber threat is one of the most serious economic and national security challenges we face as a nation' and that 'America's economic prosperity in the 21st century will depend on cyber security' [7].

The cyberattack that should be underscored is the attack that suffered by Estonia in 2007. Different Estonian financial, educational, and newspaper websites were hacked for three weeks [8]. It was considered the first cyberwar, which took the attention of the NATO Bucharest Summit Declaration. NATO announced a policy on cyber defence in 2008 [9].

Inherent and internal weakness in the configuration and implementation of a computer system and network creates vulnerabilities that render them susceptible to cyberattacks and threats. Incorrect configuration, lack of adequate procedures, inexperienced or untrained personnel are examples of vulnerabilities in building a computer network system. These vulnerabilities increase the chances of threats and attacks within a network or from outside a network. A significant number of people from different fields are becoming dependent on cyber networks. Using a particular penetration technique, an agent that causes harmful and undesirable effects in activities and behaviour of a computer or network is called a threat [10]. Cyber security is to protect the integrity of the data, networks, and programs from cyber threats to cyberspace [11].

Since the inception of the first computer virus in 1970, there is a race between cybercriminals and defenders [12]. It is getting more and more challenging to fight against these cyber security attacks and to keep a match with the speed of security attacks. Currently, researchers are focusing on the urgent need of finding new automated security methods to cope with these security challenges. One of the best and effective considered practice is to use automated machine learning techniques to detect new and previously unseen cyber threats [13].

A. EVOLUTION OF MACHINE LEARNING AND CYBER SECURITY IN LAST DECADE

The usage of machine learning and artificial intelligence techniques is getting expanded rapidly in different areas of life such as finance [14]–[16], education [17], medicine [18]–[21], manufacturing industry [22], and particularly in the field of cyber security [23]–[28].

ML techniques are playing a vital role in numerous applications of the cyber security for early detection and prediction of different attacks such as spam classification [29]–[32], fraud detection [33]–[36], malware detection [37]–[40], phishing [41]–[43], darkweb or deepweb sites [44], [45], and intrusion detection [46]–[49]. ML techniques can address the scarcity available of required personnel with expertise in these niche cybercrime detection technologies. Moreover, vigorous approaches are needed to detect and react against the cyberattacks of the new generation (automated and evolutionary). Machine learning is one of the possible solutions to act quickly against such attacks because ML can learn from experiences and respond to newer attacks on time. There is a lot of literature available on the Internet that describes the application of ML for the predication of cyber threats on darkweb or deepweb. Mohammad *et al.* [45] applied ML models to predict cyber threats by evaluating the social networks of hackers on darkweb. Sarkar *et al.* [50] used a suite of social network features and applied ML models to predict whether there would be an attack on a particular organization on the predicted date or not. They have performed experiments by gathering the data from 53 forums on darkweb. The predications of attacks through the discussion of darkweb are out of scope from this survey paper. However, recent advancements in this area can be found in [51]–[54]. Figure 1 depicts the trends of cyber security and the two areas related to data science (i.e., ML and deep learning (DL)) as a whole and separately. We had got the stats from Scopus on June 23, 2020. Though deep learning can be considered as a subset of machine learning, some articles have used the term of deep learning instead of machine learning in dealing with cyber security. We have searched and checked the trends of cyber security and ML and the trends of cyber security and DL separately to study them in more details. We have shown the trends in Figure 1 for the last ten years. In the first half of the decade, the ML models were applied for the detection of attacks on cloud security, malware, and intrusions. However, the trend has been increased at a phenomenal rate with the emerging development in the field of deep learning.

Currently, machine learning and deep learning models are being applied almost in all areas of cyber security to detect and respond against cyberattacks [55]. Note that the publications count of these terms is not intended to be comprehensive as we have targeted the Scopus database to show the publication trends and to give an idea of the importance of both research areas. It can be observed that the popularity of both areas is emerging with an abrupt growing pace. Besides the search strategy, we have followed in the following section, we have also provided the trends with multiple perspectives in the Appendix.

Currently, many traditional cyber security systems are being used including SEIM Solutions [56], intrusion prevention system (IPS) [57], unified threat management (UTM) [58], Firewalls, and antiviruses, to name a few. These traditional systems have a lack of automation (usage of AI techniques) and have a dependency upon static control of devices according to predefined rules for network security. The AI-based system performs better than traditional threat detecting techniques in the context of error rate, performance, and responding to the cyberattack [59]. The error rate both in terms of detecting and responding to an attack of AI-based systems is better than traditional systems. The performance of AI-based systems, including error rate, correct prediction of an attack, and count of the false positive, is better than that of traditional systems while detecting and responding to an attack. AI-based systems also reduce the amount of time to the investigation of network vulnerabilities, fixing and patching networks infected by malware [60]. According

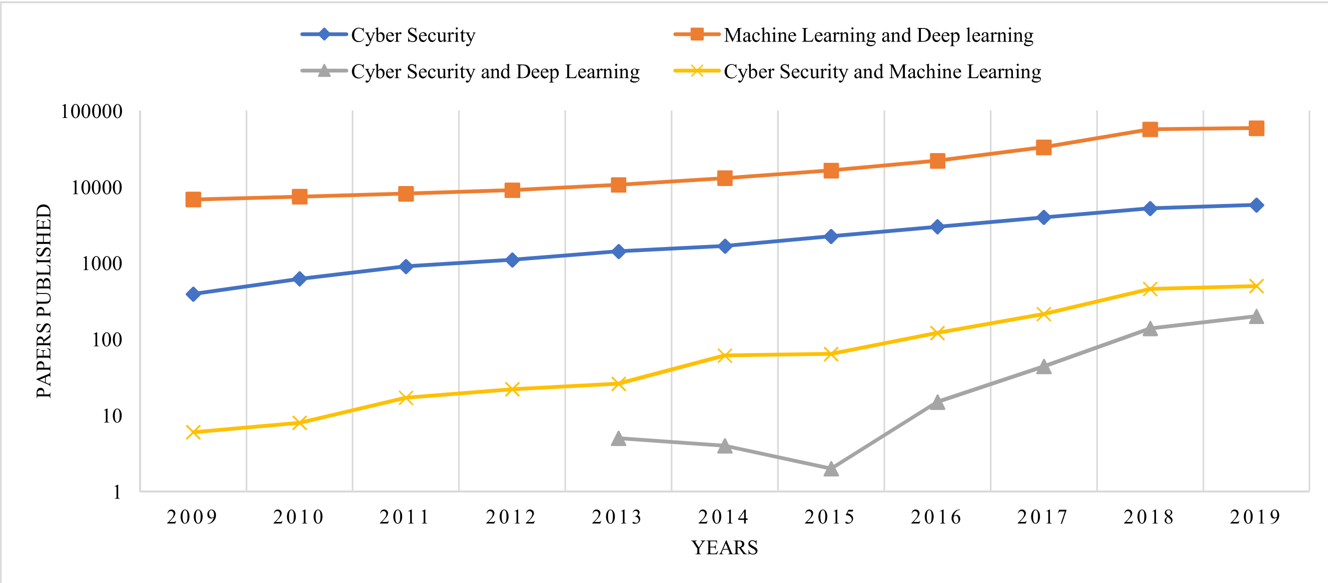


FIGURE 1. Publications Trends of Machine Learning and Cyber Security (source: Scopus).

to a study, more than 60% of the attacks are identified once they have already caused damages to the cyberspace [61]. Currently, there is a need to have new automated security methods to cope with these security challenges and threats. With the rapid growth of smartphones and the availability of sophisticated functions, smartphones are victims of cybercriminals. Machine learning approaches are also playing a vital role in improving the efficiency of detection and prevention techniques against threats to mobile devices [62].

Machine learning techniques are playing their roles on both sides, i.e. attacker side and cyber security side. On the cybercriminal side, cyber attackers and criminals are using ML techniques to find the vulnerabilities of the system and sophisticated ways of attack to pass through the defence wall. On the defence side, ML models are playing a vital role to provide robust and smarter techniques to improve the performance and early detection of attacks to decrease the impact and damage that occurred [63], [64]. Machine learning techniques are combined to enhance the accuracy of correct and early classification of cyberattacks [65]. However, most of the studies are performed with an inadequate dataset. None of the investigated surveys focused on a comprehensive and combined overview of cyber threats and attacks on both mobile devices and computer networks.

B. CONTRIBUTION OF THE PAPER

The purpose of this article is to review the key machine learning techniques applied in cyber security and point out the trend of using machine learning techniques for cyber security. We have provided a brief description of machine learning techniques, and how machine learning techniques have been, or could be, used to detect and classify cyberattacks such as intrusion detection, malware detection, and spam detection on both computer networks and mobiles or smartphones devices.

Any search strategy must allow the completeness of the search to be assessed. To identify relevant contributions in cyber security and machine learning, IEEE Xplore, ACM digital library, Emerald Insight, SpringerLink and ScienceDirect were queried for papers having ('Machine Learning' and 'Cyber Security'), ('Machine Learning' and 'Cybersecurity'), ('Deep Learning' and 'Cyber Security'), ('Deep Learning' and 'Cybersecurity'), ('Machine Learning' and 'Malware'), ('Machine Learning' and 'Intrusion Detection'), ('Machine Learning' and 'Spam'), ('Deep Learning' and 'Malware'), ('Deep Learning' and 'Intrusion Detection'), and ('Deep Learning' and 'Spam') in title, abstract or keywords. Also, Web of Science, Google Scholar, and Scopus were queried to double-check the findings and to find other related papers in less-known libraries. Google Scholar was also used for forward and backward searches. We have focused on recent advancements in the last ten years. These online databases were chosen as they offer the most significant peer-reviewed full-text journals and conference proceedings, book chapters, and reports covering the field of machine learning and cyber security. In total, 7915 documents were retrieved. The duplicated items were removed. The title and abstract of 1728 documents were screened to identify potential articles. The full-text assessment of 770 was made according to the relevancy of the inclusion criteria. Further, 486 studies were excluded. We have excluded the articles that were discussing (1) social network forensics, (2) irrelevant cyber threats, (3) threats to cyber-physical grids, (4) threats to cloud security, IoT devices, (5) smart grids, and smart cities, and (6) satellite communication, 5G and wireless communication. With forward and backward search, 28 more studies were retrieved. In total, 312 studies were finally selected for data extraction purpose. Figure 2 illustrates the process of article inclusion and selection. In addition to these, the previous survey and review articles were used to provide a comprehensive survey of machine learning techniques in cyber security.

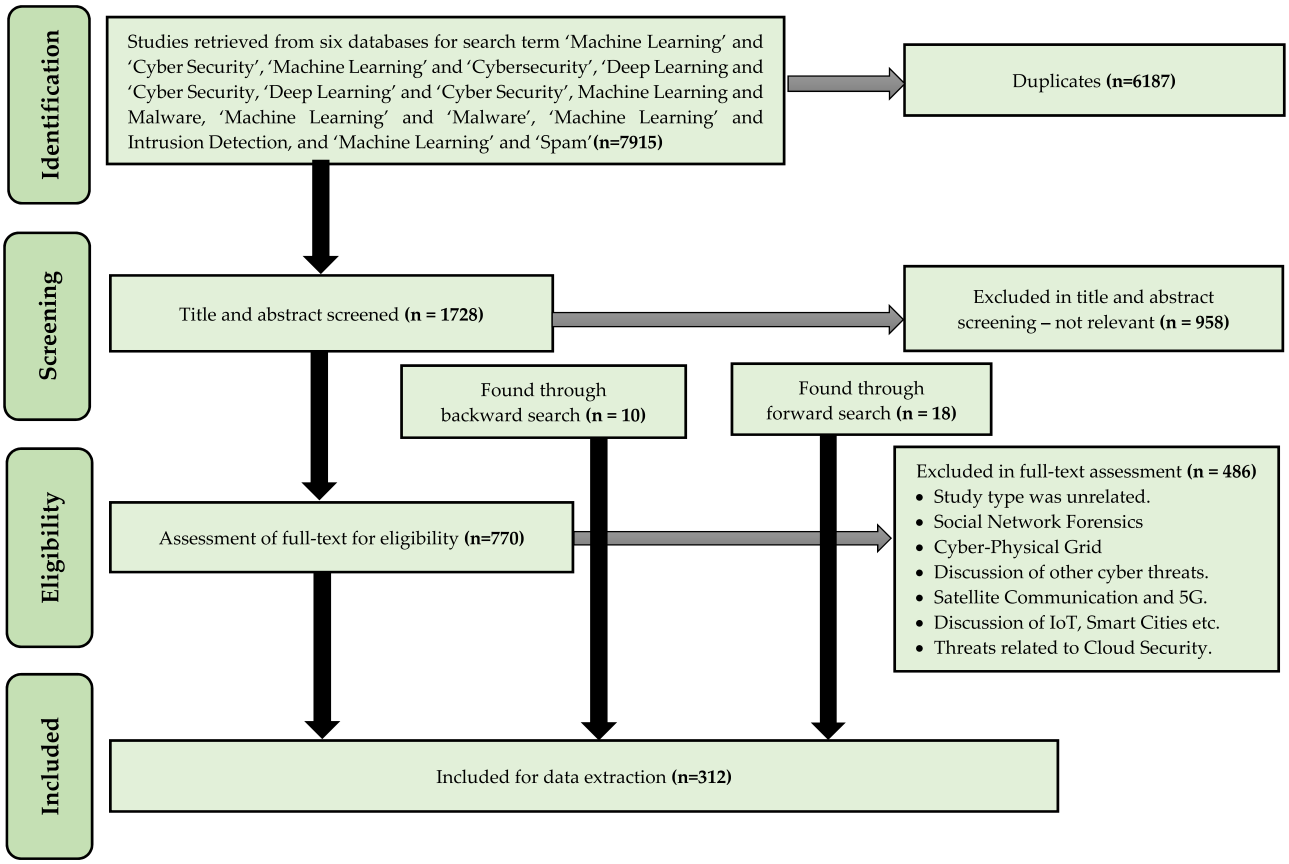


FIGURE 2. An Illustrative View of the Process for Article Selection.

It is expected that the used search terms will cover most, if not all, of the work incorporating machine learning methods for cyber security.

Nevertheless, Google Scholar is further utilized to check the citation of found papers (forward-searching) to update our search and to look for other scientific resources to make sure nothing is neglected. The last update of the searching of papers was done on May 3rd, 2020. Table 1 depicts the list of acronyms used in this article for convenient referencing.

We are unaware of any existing survey that provides the application of ML techniques in cyber security on both computer and mobile networks. Our work also presented commonly used ML tools, security datasets, graphical summary of significant components of cyber security and available ML techniques to fight against threats and attacks on cyberspace, and future challenges such as trustworthiness and adversarial machine learning under one umbrella. Table 2 presents a comparison of our paper with existing surveys and review articles. Many current surveys, either present applications in a particular domain or lack of giving basic knowledge that a new researcher requires to get in or understand this domain. Furthermore, most of the survey articles discuss particular threats and attacks on a network only. We have focused on significant cyber security such as intrusion detection, malware detection, and spam classification on both networked computers and mobile devices.

In particular, machine learning techniques have not only increased threats on computer networks but also held a lot of promises for detection and classification of attacks and threats on mobile devices and networks. Our survey covers cyber threats on both mobile devices and computer networks.

Comparing to existed survey papers in the area, our survey is inclusive and unique in providing the following aspects: providing basic insights of cyber security threats on both mobile devices and computer networks, giving descriptions of commonly used security datasets, summarizing the state-of-the-art ML techniques to handle these threats, indicating popular ML tools, describing evaluation metrics to evaluate the performance of ML techniques, and pointing out current challenges of ML techniques for cyber security. We have provided a graphical summary of major components of cyber security and available machine learning techniques to fight against these attacks on cyberspace. The last updating on the paper's citations count (source: Google Scholar) was done on June 05, 2020, in Table 2.

C. ORGANIZATION OF THE PAPER

Figure 3 depicts the overall organization of this paper. Section II provides cyber security basics, including the basics of attacks and threats to cyber security, commonly used security datasets, and evaluation metrics. Section III presents an introduction to the key machine learning models and commonly used ML tools for cyber security. Section IV reviews applications of ML techniques in the detection and classification of spam, intrusion, and malware on both computer networks and mobile devices, particularly in the last ten years.

TABLE 1. List of acronyms.

ADFA	Australian Defence Force Academy
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under Curve
CNN	Convolutional Neural Network
DBN	Deep Belief Networks
DL	Deep Learning
DNN	Deep Neural Network
DoS	Denial of Service
DPI	Deep Packet Inspection
DT	Decision Tree
FDR	False Discovery Rate
FNR	False Negative Rate
FOR	False Omission Rate
FPR	False Positive Rate
HIDS	Host Intrusion Detection System
HMM	Hidden Markov Model
ID	Intrusion Detection
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
k-NN	K Nearest Neighbour
LDA	Latent Dirichlet Allocation
LR	Linear Regression
ML	Machine Learning
MLP	Multi-Layer Perceptron
NB	Naïve Bayes
NLP	Natural Language Processing
NN	Neural Network
PCA	Principal Component Analysis
R2L	Remote to Local
RBM	Restricted Boltzmann Machine
RF	Random Forest
RNN	Recurrent Neural Networks
SMS	Short Message Service
SOM	Self-Organizing Map
U2R	User to Root
URL	Uniform Resource Locator
UTM	Unified Threat Management
WNN	Wavelet Neural Network

Section V presents current challenges to machine learning for cyber security and the trustworthiness of classification techniques. Finally, Section VI concludes the whole work.

II. CYBER SECURITY BASICS

A. BASICS OF ATTACKS AND THREATS

The possible breaches and security violations on a computer system or mobile devices include obtaining unauthorized access, destruction, and alteration of information with an intention to possibly harm, to name a few. The possible risk and danger of all mentioned security violations are called threats, and any attempt to do any violation is called an attack [92]. Cyber security can be defined in several ways. Kaspersky's [93] definition of cyber security includes having a defensive mechanism against malicious attacks on computers, servers, and data on a computer network and mobile device. Kaspersky further divided cyber security into network security, information security, and other categories [93], [94]. Cyber security field overlaps with all major categories defined by Kaspersky and International Organization for Standardization (ISO). It is an accepted fact that attackers are evolving and adapting new techniques at a faster pace than that of the defenders who detect and defend those penetrations, intrusions, and attacks [95]. The annual report released by Cisco in 2018 provided the fact that more than half percent of attacks caused damage of $500 million or more [96]. Cyber security aims to protect personal information, government data, and business reports from illegal penetration, misuse, and handling with malintent. Furthermore, cyber security covers a) the protection of software, tools, and equipment, and b) ensuring and guaranteeing the privacy and integrity of the information being protected from several threats and attacks [97].

Phishing and malware are considered as the most critical attacks [88]. Phishing, also called brand spoofing, is a process of accessing personal data to disrupt or misuse by showing itself as a legitimate user. One example of phishing can be showing web pages as legitimate web pages and behaving like tricksters to acquire personal information [98]–[100].

Malware is broadly categorized into three main categories: worms, Trojan horses, and viruses. A virus is a program that negatively affects computer operations without the knowledge of the user. A virus can damage the files and operating system of the computer. Elk Cloner was the first computer virus spread through a floppy drive in 1981 [101]. A worm is a program that repeatedly copies itself hence consumes the resources on the system or network. Trojan horse, unlike viruses or worms, does not replicate itself but presents itself as a legitimate program and triggered against a particular operation or action [102], [103].

Another threat to cyber security is unwanted and unsolicited spam email messages. These emails not only take much time and fill the mailbox but also become the source for the execution of Java applets when an email is read. Spams on mobile devices and mobile networks can be in the form of spam calls, text, and video messages [104]–[107]. Spam messages as text on Twitter and as video on YouTube are extensive spreading venues for spammers.

Each network security system consists of a protection mechanism such as firewalls, anti-virus programs, and intrusion detection systems. The intrusion detection system (IDS) helps to discover and identify any illegal penetration or unauthorized access with malign intentions.

Network analysis for IDS is categorized into three main categories: a) signature-based that is mostly used to detect known attacks by avoiding a large number of false alarm rate (FAR), b) anomaly-based that is mainly used to identify anomalous behaviour of network and system, and c) hybrid-based that is the combination of a) and b) to decrease the FAR for unknown attacks. Others have categorized the attacks into four major categories [108]. Denial of service (DOS) is an attack where a cybercriminal makes the network system busy or shortage of memory resource in a way that the access request from the legitimate user is not entertained. Remote to Local (R2L) attack is an attack where a remote user tries to gain local access over a network by exploiting its vulnerabilities. User to Root (U2R) attack

TABLE 2. Overview and comparison of existing surveys with our paper (legend: √ means covered; ≈ means partially covered; × means not covered), (citation's count source: google scholar, last updated: october 05, 2020).

Sr.#	Reference No.	Year	Citations	No of References	Cyber Security							Machine Learning					Graphical Summary of threats to Cyberspace and available ML Tech.
					Mobile Based			Computer-Network/Host Based									
					Spam Detection	Malware Detection	IDS	Spam Detection	Malware Detection	IDS	Security Datasets	Techniques	Metrics	Tools	Trustworthiness	Adversarial ML	
1	[66]	2010	28	38	×	×	×	×	×	√	√	≈	×	×	×	×	×
2	[67]	2012	42	16	×	×	×	×	×	√	×	√	×	×	×	×	×
3	[68]	2013	79	21	×	×	×	×	×	√	×	√	×	×	×	×	×
4	[69]	2014	48	17	×	√	×	×	≈	×	×	≈	√	×	×	×	×
5	[70]	2014	264	51	×	×	×	×	√	×	×	×	×	×	×	×	×
6	[71]	2014	14	18	×	×	×	×	×	√	√	×	√	×	×	×	×
7	[72]	2014	21	24	×	×	×	≈	×	√	×	×	×	×	×	×	×
8	[73]	2015	971	113	×	×	×	×	×	√	√	√	√	×	×	×	×
9	[74]	2016	16	164	×	×	×	√	√	√	×	×	×	×	×	×	×
10	[75]	2016	02	10	×	×	×	×	×	√	×	×	×	×	×	×	×
11	[76]	2017	04	21	×	×	×	×	×	√	≈	×	√	×	×	×	×
12	[77]	2017	96	154	×	×	√	×	×	√	√	√	√	×	×	×	×
13	[78]	2018	125	78	×	×	×	×	≈	√	√	√	√	×	×	×	×
14	[79]	2018	10	68	×	×	×	×	√	√	×	√	×	×	×	√	×
15	[80]	2018	27	107	×	×	×	×	√	×	≈	×	×	×	×	×	×
16	[81]	2018	42	40	×	×	×	≈	≈	√	×	√	≈	×	×	×	×
17	[82]	2018	07	14	×	×	×	≈	≈	≈	×	√	×	×	×	×	×
18	[83]	2018	05	76	×	≈	×	×	√	√	×	×	×	×	×	×	×
19	[84]	2018	27	84	×	×	×	×	√	×	≈	√	≈	≈	×	×	×
20	[85]	2019	01	12	×	×	×	×	≈	≈	×	×	×	×	×	×	×
21	[86]	2019	158	45	×	×	×	×	×	√	√	√	×	×	×	×	×
22	[87]	2019	44	174	×	×	×	≈	×	√	√	√	√	×	×	×	×
23	[88]	2019	08	200	×	×	×	√	√	×	×	√	√	×	×	×	×
24	[89]	2019	07	55	×	×	×	×	√	√	×	×	×	×	≈	√	×
25	[90]	2020	0	142	×	×	×	×	×	√	√	√	×	√	×	√	×
26	[91]	2020	2	204	×	×	×	×	×	√	√	≈	√	×	×	√	×
27	**Our Pap.**	**2020**	**-**	**668**	√	√	√	√	√	√	√	√	√	√	√	√	√

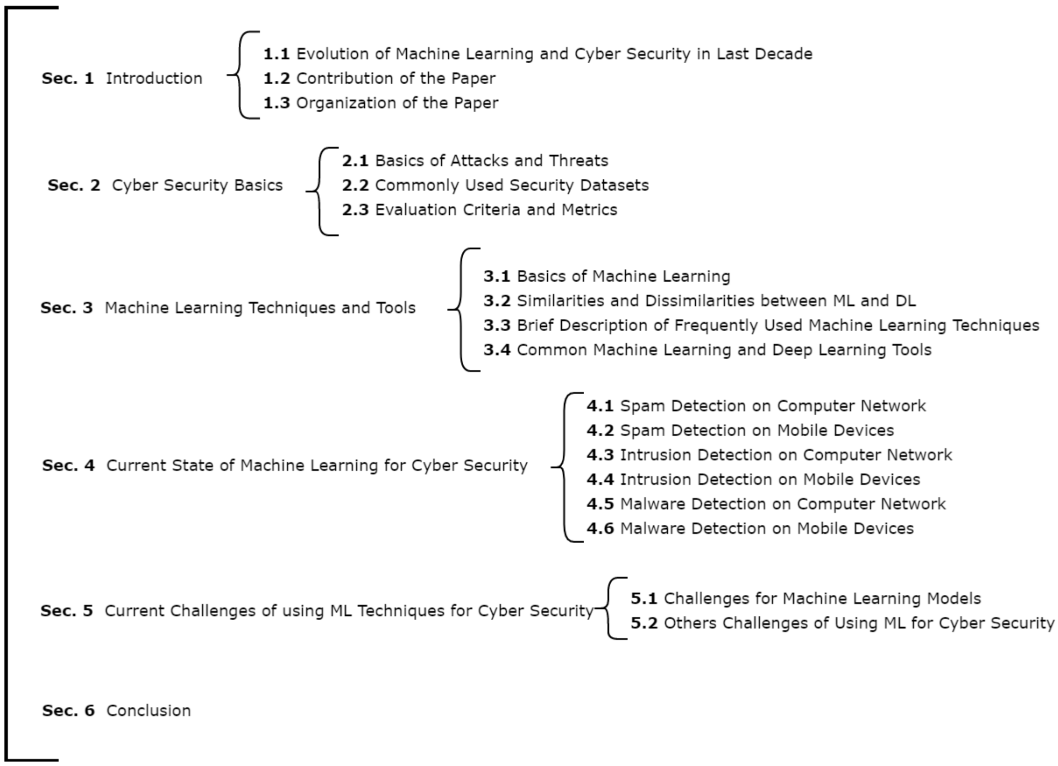


FIGURE 3. Outline of this Paper.

happens where a legitimate user with limited access to the network tries to gain privileges as a root user. An attack where a cyber-criminal scan a computer system or network to exploit the weakness and vulnerabilities for future exploitation is called probing.

ML-based techniques performed better than the conventional signature-based system because a slight variation in attack pattern can easily bypass the signature-based IDS. However, ML-based systems learn from traffic behaviour. They can easily detect the attack variants. Further, the range of CPU load is from low to moderate in ML-based system as they do not analyse all signatures in the database. ML-based systems also show better performance in terms of accuracy and speed while capturing and exposing the complex properties of attack behaviour.

There are other types of attacks and threats such as SQL injection attack, drive-by attack, password attack, a man in the middle, authentication attacks, wrapping attacks, watering hole, and webshell [65], [109]. However, we have just considered intrusion detection (ID), malware detection, and spam detection in this review article. We have highlighted how ML techniques are being applied to improve cyber security against these attacks both on computer systems and mobile devices.

The researchers have proposed different taxonomies and provided different classifications of attacks. Kotapati *et al.* [110] divided the attacks into interception, fabrication, modification, denial of service, and interruption with respect to the physical access on the 3G network. Chris *et al.* [111] classified the attacks based on the nature of attacks, including attack vector, operational and informational impact, defense, and attack target. However, the proposed taxonomy didn't consider physical and defense strategies. Narwal *et al.* [112] characterized cyberattacks based on the sector of applications such as industrial applications, web applications, mobile devices and computer operating systems, etc. Others in [113], [114] classified the attacks into active attacks and passive attacks. The detailed discussion on different attack taxonomies can be found in [115], [116]. Nevertheless, intrusion detection, malware, and spam classification and detection are the main focus of this article.

B. COMMONLY USED SECURITY DATASETS

Malicious activities are performed on the computer and mobile networks to disrupt, deny, and destroy the data and services available. These activities involve network attacks, phishing, spams, and the spreading of malware on vital information available on networks. These activities compromise the integrity, availability, and confidentiality of systems and have a negative impact on the global economy [117], [118]. A drastic increase in the amount of cybercrimes has initiated the application of machine learning techniques to provide solutions for early detection and prevention of such cybercrimes [43]. Machine learning techniques offer better results in cases that they are trained on diverse, massive, and real-time datasets. This section will briefly give insights into different datasets used by machine learning techniques for security applications. An overview of various frequently used security datasets is provided in Table 3.

TABLE 3. Overview of various frequently security datasets.

Sr .#	Name of Dataset	# of Attacks	Attribute	Count of Attribu te	Referred In
1	KDD Cup 99 *	22	Features	41	[119-121]
2	NSL-KDD *	22	Samples	50,000	[122, 123]
3	ADFA/ADFA -Linux *	7	Traces	5206	[124, 125]
4	ISOT *	-	Flows	167542 4	[126-129]
5	DARPA IDS *	38	-	-	[130-133]
6	CTU-13 *	-	Scenario	13	[134-136]
7	HTTP CSIC-2010 *	3	HTTP Requests	61,000	[137-139]
8	UNSW-NB-15 *	9	Features	49	[140-145]
9	CICIDS2017 *	15	Features	83	[146-148]
10	Bot-IoT *	8	Features	33	[149, 150]
11	Spambase †	-	Emails	4601	[151-153]
12	Enron †	-	Emails	0.5M	[154-157]
13	SMS Spam Collection †	-	SMS	5574	[158, 159]
14	Email Spam †	-	Emails	3052	[160, 161]
15	VirusShare ‡	-	Samples	34,506, 159	[162-164]
16	Malicious URL ‡	4	Features	83	[165, 166]
17	CICAndMal‡	4	Samples	5491	[167-169]
18	Kharon Malware ‡	7	-	-	[170, 171]
19	Android Validation ‡	-	Apps	8000	[172, 173]

* Intrusion Detection Dataset
† Spam Dataset
‡ Malware Dataset

Defence Advanced Research Project Agency (DARPA) datasets were collected and made publically available by the DARPA ID Evaluation Group [130]. DARPA ID Datasets are composed of three subsets of data, namely, 1998 DARPA ID Assessment Dataset, 1999 DARPA ID Assessment Datasets, and 2000 DARPA ID Scenario Specific datasets. 1998 DARPA version of the dataset is considered as a benchmark for the ID's assessment. DARPA Datasets are mostly used for attack detection. KDD Cup 99 dataset [120] was created in 2007 for the European Conference for ML and Knowledge Discovery. This dataset is based on the 1998 DARPA dataset that included 41 different types of features. These features are categorized as basic, content and traffic features. Out of the 41 features, 34 fixed features are of type continuous, whereas the rest of the seven features are symbolic type. This dataset is mostly used and observed for intrusion detection. It contains 22 types of attacks. Attacks are further categorized as Normal, DoS, R2L, U2R and Probe. NSL-KDD is an improved version of the KDD Cup 99 dataset, also used for intrusion detection. It contains four categories of 22 attacks which are DoS, Probe, R2L and U2R. DARPA and other benchmark datasets were collected more than ten years ago and cannot handle host-based anomalies of modern computer systems.

Czech Technical University (CTU) proposed a dataset named CTU-13 in 2011 [136]. This dataset is a collection of 13 different seizures (samples/scenarios) of real botnet traffic with a combination of normal and background traffic. This dataset was labelled carefully in a controlled environment. Australian Defence Force Academy (ADFA) released a Linux based dataset that coped the limitation of DARPA in 2013 [125]. ADFA made public two versions of subsets, i.e. Windows-based and Linux-based which record the system call's order. Each system call was provided with a parallel system call number. This dataset was provided with seven attacks in 5206 traces for intrusion detection. Information security and object technology (ISOT) dataset was provided with 1,675,424 traffic flow [140]. This dataset is considered as the biggest dataset for Ericson Research Lab located in Hungary. This dataset is a combination of publically available botnets and dataset collections from LBNL. This dataset contains three subcategories of datasets, including the ISOT Botnet dataset, ISOT Ransomware, and ISOT HTTP Botnet Datasets. Australian Centre for Cyber Security created the UNSW-NB 15 dataset with 49 features and nine types of attack's categories for ID. Authors in [140] used this dataset to apply support vector machine, Logistic regression and decision tree techniques on the cloud security domain. HTTP CSIC-2010 dataset is a collection of hundreds of thousands of web requests and is typically used to test for web attacks. This dataset is a collection of 61,000 HTTP requests. Illegal, dynamic, and static requests are three major attack categories in this dataset. This dataset is recommended and widely used for the detection of attacks on the web [174]. CICIDS2017 is another dataset collected from 03-07 July 2017 contains various attack scenarios implemented by this dataset, including DoS, Web attack, and Botnet [48]. The bot-IoT dataset was proposed in 2018 for IoT devices [175]. The bot-IoT dataset consists of more than 72,000,000 records. This dataset implements data exfiltration attacks, service scan and keylogging. Node-red tool is used for Bot-IoT dataset for network behaviour simulations. Bot-IoT dataset uses a lightweight protocol named as MQTT protocol [176]. The datasets mentioned so far are used for intrusion detection.

Spambase is an email dataset comprising of 57 attributes of integer and real data types. The dataset has 4601 instances and is mostly used for spam email classification purposes [177]. Enron is another commonly used email dataset. It is used for spam email classification [178]. This dataset is publically available, containing personal and official emails. There are six versions of the Enron dataset. Enron dataset contains 517,413 emails from 151 users. Other commonly used spam datasets are PU datasets [179] and Ling-Spam [180]. SMS Spam Collection is another dataset contains 5,574 labelled SMS [158]. The SMS messages in this dataset are extracted from various resources, including 425 SMS from Grumbletext, 3,375 from NUS SMS Corpus, and 450 SMS ham (not spam) messages from Caroline Ph.D. Thesis [181], respectively. Email Spam is another dataset collected from Spam Assassin and contains 3052 files [160].

VirusShare is a collection of malware that contains 34,506,159 samples. It is mostly infected and commonly

TABLE 4. Confusion matrix.

		Predicted Class	
		Benign /Positive	Malicious/Negative
Actual Class (Ground Truth)	Benign /Positive	True Positive (TP)	False Negative (FN)
	Malicious/Negative	False Positive (FP)	True Negative (TN)

used for malware detection and analysis [182]. The uniform resource locator (URL) dataset [165] contains instances of Internet traffic. It was mainly proposed to blacklist malicious URLs. CICAndMal2017 is an Android malware dataset consists of benign and malware applications [183]. CICAndMal2017 dataset categorises malware into four classes which include: Scareware, SMS malware, ransomware, and adware. This dataset was also proposed to identify and blacklist malicious Android applications. Kharon malware dataset was collected in 2016 to gauge the performance of research experiments [184]. Kharon malware dataset is a collection of android documented malware attacks [185].

The Android adware and general malware dataset comprises of adware applications, general malware applications, and benign applications [186]. A lightweight detector was used for this dataset to distinguish between these three categories of application. There were 1900 applications used to compose this dataset. UNB ISCX Android validation dataset [172] is another Android-based dataset that shows the different relationships between apps, for example, false siblings, siblings, cousins, and step-siblings. Figure 4 depicts a more brief and compact overview of the evolutionary timeline of frequently used security datasets.

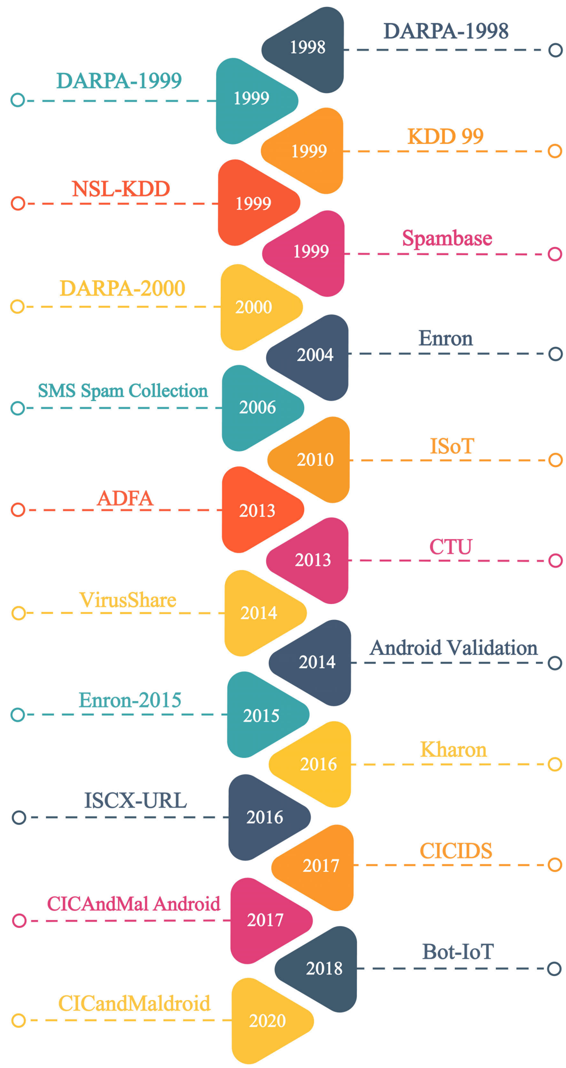


FIGURE 4. Evolution of Frequently Used Security Datasets.

C. EVALUATION CRITERIA AND METRICS

There are different indicators and measures to evaluate an ML model. Every learning task has an emphasis on various measures. A confusion matrix is regarded as one of the formal ways to present the details of the learning model. A confusion matrix, also termed as an error matrix, is a table that describes the performance of a prediction or classification model [187]. A confusion matrix, as shown in Table 4, presents the results of binary classification into four different categories. It provides the result of classifier in the form of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values that further build other measures. Apart from error rate, other criteria such as time complexity, space complexity, and adaptability of learning algorithms should also be focused. However, the priority of the metric varies from application to application. Suppose, while classifying a financial transaction into either genuine or fraudulent, it is essential to consider false negatives. A single value of FN for a financial transaction can result in a substantial financial loss. Therefore we cannot specify what metrics are specifically important for a class of intrusion/attack. Usually, classification models for cyber security are assessed based on the following terms:

1) True Positive: the count of normal traffic/non-malignant/positive samples/applications that are correctly classified by the model.
2) True Negative: the count of attack/malicious/negative samples/applications that are correctly classified by the model.
3) False Positive or False Alarm: the count of attack/malicious/negative samples/applications that are misclassified as normal/positive by the model.
4) False Negative: the count of normal traffic/non-malignant/positive samples/applications that are misclassified as abnormal/negative by the model.

The aforementioned terms in the confusion matrix are further used to calculate the following metrics:

1) PRECISION/POSITIVE PREDICTIVE VALUE

It is a ratio of correctly classified benign/positive samples/ applications to all classified benign/positive samples/ applications in the dataset (Eq. 1). A higher value of precision is desirable and shows better performance of a classifier.

$$\text{Precision} = TP/(TP + FP) \tag{1}$$

2) RECALL/ SENSITIVITY/TRUE POSITIVE RATE (TPR)

It is a percentage of benign/positive samples/applications correctly classified to the total benign/positive samples/ applications in the dataset (Eq. 2). A higher value of recall is desirable and shows better performance of a classifier.

$$\text{Recall} = TP/TP + FN \tag{2}$$

3) SPECIFICITY/TRUE NEGATIVE RATE (TNR)

It is a ratio of correctly classified attack/malicious/negative samples/applications to the total number of attack/malicious/ negative samples/applications in the dataset (Eq. 3). A higher value of specificity is desirable and shows better performance of a classifier.

$$\text{True Negative Rate} = TN/(TN + FP) \tag{3}$$

4) ACCURACY

It is a ratio of correctly classified samples/applications to all samples/applications in a dataset (Eq. 4). The higher value of accuracy shows the correctness of the classifier. A higher value of accuracy is desirable.

$$\text{Accuracy} = (TP + TN)/(TN + FP + FN + TP) \tag{4}$$

5) ERROR RATE

It is a ratio of incorrectly classified samples/applications to all samples/applications in the dataset (Eq. 5). A lower value of the error rate is desirable and shows better performance of a classifier.

$$\text{Error Rate} = (FP + FN)/(TN + FP + FN + TP) \tag{5}$$

6) FALL OUT/FALSE POSITIVE RATE (FPR)

It is a ratio of incorrectly classified malicious/negative samples/applications to the total actual number of attack/ malicious/negative samples/applications in the dataset (Eq. 6). A lower value of FPR is desirable and shows better performance of a classifier.

$$\text{False Positive Rate} = FP/(FP + TN) \tag{6}$$

7) MISS RATE/FALSE NEGATIVE RATE (FNR)

It is a ratio of incorrectly classified benign/positive samples/applications to the total actual number of benign/positive samples/applications in the dataset (Eq. 7). A lower value of FNR is desirable and shows better performance of a classifier.

$$\text{False Negative Rate} = FN/(FN + TP) \tag{7}$$

8) FALSE DISCOVERY RATE (FDR)

It is a ratio of incorrectly classified malicious/negative samples/applications to the total number of classified attack/malicious/negative samples/applications in the dataset (Eq. 8). A lower value of FDR is desirable and shows better performance of a classifier.

$$\text{False Discovery Rate} = FP/(FP + TP) \tag{8}$$

9) FALSE OMISSION RATE (FOR)

It is a ratio of incorrectly classified benign/positive samples/applications to the total actual number of classified benign/positive samples/applications in the dataset (Eq. 9). A lower value of FOR is desirable and shows better performance of a classifier.

$$\text{False Omission Rate} = FN/(FN + TN) \tag{9}$$

10) F1-SCORE

It is a measure of calculating the accuracy of the model using the values of precision and recall (Eq. 10). This measure will be helpful if the user seeks a balance between recall and precision, and sample distribution is an uneven class distribution. A higher value of the F1-score shows the ML model is performing better than other models.

$$\text{F1-score} = 2.(\text{precision}^*\text{recall})/(\text{precision} + \text{recall}) \tag{10}$$

11) G-MEAN

It is calculated using the true predicted values by the classifier (Eq. 11). In the case, where the number of negative samples is more than the positive samples, the accuracy will not project the correct picture for positive samples. G-Mean will help in that case.

$$\text{G-mean} = \sqrt{(TP/(TP + FN)XTN/(TN + FP))} \tag{11}$$

12) RECEIVED OPERATING CHARACTERISTIC (ROC) CURVE

The commonly used graph that provides a summary of all threshold's performance by plotting the values of TPR (y-axis) against FPR (x-axis).

13) AREA UNDER CURVE (AUC)

The size of the area which comes under ROC is called AUC that ranges from 0.5 to 1.0 values. A higher value of AUC shows better performance of a classifier.

14) MEAN SQUARED ERROR (MSE)

This metric can be calculated by taking the average of the squared difference or error that occurred between the actual values and predicted values of the classifier. A lower value of MSE is desirable and shows better performance of a classifier.

15) MEAN ABSOLUTE ERROR (MAE)

This metric can be calculated by taking the average of the absolute difference or error that occurred between the actual values and predicted values of the classifier. A lower value of MAE is desirable and shows better performance of a classifier.

16) MEAN ABSOLUTE PREDICTION ERROR (MAPE)

The MAPE is the average value of the absolute difference between the actual and predicted values of the classifier. A lower value of MAPE is desirable and shows better performance of a classifier.

17) ROOT MSE (RMSE)

This measure can be calculated by taking the square root of the mean squared error. A lower value of RMSE is desirable and shows better performance of a classifier.

III. MACHINE LEARNING TECHNIQUES AND TOOLS

A. BASICS OF MACHINE LEARNING

Artificial Intelligence (AI) is a branch in the field of computer science that develops techniques, theories, and applications. Artificial Neural Networks (ANNs) developed from early attempts to implement a simplified model inspired by the way, neurons activate other neurones in a biological system such as an organic brain. Machine learning (ML) is a sub-branch of AI. ML algorithms build models based on training data, which allow the models to make predictions (or decisions) about new data without being explicitly instructed on how to do so [188], [189]. ML has applications in different areas of life [190], [191]. ML techniques are being applied to improve cyber security and early detection of several automated and new attacks [81], [192], and phishing website detection [193], [194].

Machine learning can be classified into three major categories concerning methodology: supervised machine learning, unsupervised machine learning, and semi-supervised machine learning. In supervised machine learning, the targeted labels or classes are already known for the data, and those labels and classes are used to learn for the computations, e.g. classification and regression. In unsupervised machine learning, the targeted value is not already known. Unsupervised learning mainly focuses on finding out relationships between samples. It works by finding the patterns among data such as clustering. Where there is a portion of data labelled or needing human experts during the acquisition of data, then the process is called semi-supervised ML. The human expert during the labelling process will surely help to solve the problem and improve the accuracy of the model [73]. Reinforcement learning (RL) is another subdomain of machine learning. Sometimes, RL is also termed as learning with a critic because there is input to the algorithms against any wrong prediction. However, it has not been told to the algorithm of how to correct it. Instead, the algorithm has to figure out and try several possibilities until it learns the correct answer [195]. This phenomenon works based on a reward and penalty scheme. A famous example of this technique is AlphaGo [196], [197]. Deep Reinforcement learning is used in cyber security in [63], [198], [199].

Deep learning (DL) is a subset of machine learning. Both machine learning and deep learning have the same techniques and tasks but having different capabilities. The human brain inspires DL algorithms for analytical and logical thinking. There are two main research directions in DL, i.e., convolutional neural networks and deep belief networks. These areas attracted the research and academic community over the last decade [200]–[203]. Nowadays, automatic car driving is an example of DL. There are many studies in the literature that are applying DL models to improve cyber security [204]–[206]. We have put more emphasised on the ML and DL relationship in the following section.

B. SIMILARITIES AND DISSIMILARITIES OF ML AND DL

As we have mentioned in the previous section, deep learning is considered as the trend and subset of machine learning. Classical and traditional machine learning models in the past need human intervention for an optimal outcome. Traditional ML models performed better on smaller datasets. However, DL models are data-hungry models that show excellent performance on larger datasets [207] , [208]. However, if the data is insufficient (a smaller number of training samples) or poorly distributed (biased), then ML-models will be biased or perform better for particular cases. Therefore, for higher performance, a properly distributed and sufficient number of training samples are required for better performance. Although we may say that deep learning is a child of machine learning, there are some similarities and dissimilarities between the two fields which we have highlighted in Table 5.

C. BRIEF DESCRIPTION OF FREQUENTLY USED MACHINE LEARNING TECHNIQUES

This section describes common machine learning techniques. Table 6 provides a compact overview of ML models including the time complexity, pros, and cons, proposed year, and reference (ref) number.

1) SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is considered as the mostly used and successful technique of ML for cyber security tasks, especially for IDS. SVM classifies and separates the two data classes based on the notation to the margin on either side of the hyperplane. Figure 5 gives the pictographic explanation of SVM. The accuracy in classifying a data point can be maximized by increasing the margin and distances between hyperplanes. The data points that lie on the border of the hyperplane are called support vector points. SVM is classified into two major categories. It can be linear and non-linear based on the kernel function. It can also be one-class and multi-class based on detection type [209], [210]. SVM requires a lot of memory for processing and time for training.

TABLE 5. Similarities and dissimilarities between DL and traditional ML (Legend: ≈ means similar; ≠ means dissimilar).

Sr#	Nature	Similar/ Dissimilar	Explanation
1	Major Goals	≈	Both can 'learn' to do things without human intervention to produce the desired output.
2	Purpose	≈	Both are used in AI research.
3	Layering	≈	Both are layered according to their requirements.
4	Data Dependencies	≠	Traditional ML models show excellent performance on a small/medium dataset, whereas deep learning models, are known as data-hungry and have excellent performance on a bigger dataset [207].
5	Scalability	≈	Both are scalable.
6	Working	≠	ML techniques can learn through pre-programmed defined criteria. In contrast, DL is only able to identify edges (concepts, differences) within layers of neural networks when exposed to over a million data points.
7	Resource-Intensive	≈	Both are resource-intensive.
8	Number of Data Points	≠	ML used a few thousand data points for analysis. In DL, there are a few million data points used for analysis.
9	Management	≠	Algorithms are directed by analysis, whereas DL algorithms are usually self-directed.
10	Solving Technique	≠	DL is based on solving the problem end-to-end, but ML follows the divide and conquers concept.
11	Hardware Dependencies	≠	DL requires a powerful machine, preferably with GPU, and performs a significant amount of matrix multiplication. ML works on a low-end machine.
12	Time	≠	DL techniques take a long time in training but require less time for testing. ML techniques take less time in training but longer while generating the results. Nevertheless, the size of the dataset affects training and testing time.
13	Origin	≠	DL originated from the 1970s, whereas ML was originated from the 1960s.
14	Methods for Algorithms	≠	ML includes feature engineering, training, and evaluation of model performance to classify or predict. DL methods include the same steps except feature extraction is automated rather than manual.
15	Commonly used Algorithms	≠	DL Algorithms include Convolutional Neural Network, Recurrent Neural Network. Nevertheless, commonly used ML algorithms include K-nearest neighbour, Decision Tree, etc.
16	Feature Engineering	≈	Both need to understand the features that represent the data
17	Feature Extractor	≠	DL does not depend on hand-crafted features like local binary patterns, a histogram of gradients, etc., and performs a hierarchical feature extraction. ML relies on hand-crafted features as an input to perform well.
18	Applications	≈	Both are used in medical, banks, natural language processing, etc.

SVM needs training at different time intervals for better results to learn the dynamic user's behaviour.

Kernel function and parameters also affect the performance of the classifier.

2) DECISION TREE

Decision Tree (DT) is a supervised ML technique based on a recursive tree-structure. DT is composed of three things: a root or intermediate node, path and leaf node, as depicted in Figure 7. The root/intermediate node of a tree represents an object/attribute. Each divergence path of the tree represents the possible values of the parent node (object). Leaf node corresponds to the predictive category/classified attribute. The resultant tree is further represented in the form of if-then rules. During the construction of the tree, entropy and information gain measures are used to select the best possible intermediate node further. CART [211], C4.5 [212] and ID3 [213] are considered important algorithms of decision tree. ID3 works based on a greedy approach. However, it cannot handle numeric attributes. C4.5 is an improved version of ID3 and overcomes the limitations of ID3 by handling the problem of overfitting using techniques of tree pruning. An open-source implementation of C4.5 can be found as J48 in Waikato Environment for Knowledge Analysis (Weka) [214]. It can handle the problem of overfitting except when there is noisy data. CART supports both numerical and categorical attributes and handles missing values that cannot be handled by ID3.

3) K-NEAREST NEIGHBOR

K-nearest neighbor (kNN) is an unsupervised learning algorithm. It is based on a distance function that measures the difference/dissimilarity of two data instances. It takes less time in training than other classifiers. However, its computation time is overhead during the process of classification. Figure 6 depicts the working of kNN. This classifier works on the assumption that similar data points in the space will be closer to each other than those that are dissimilar. There are two broader categories of kNN based on anomaly scores. The two ways of calculating the anomaly scores are (1) It is calculated based on the difference between the k^{th} neighbor and data point. (2) It is calculated based on the density of each data instance [215]. The value of the k^{th} data point affects the overall performance of the classifier [216].

TABLE 6. An overview of frequently used ML techniques.

Model	Year	Ref. No	Time Complexity	Description	Limitations
SVM	1995	[221]	$O(n^2)$[1]	• Can be used for classification and regression. • Less overfitting	• Unable to handle large or noisier datasets efficiently. • High computational cost.
Naive Bayes	1960	[222]	$O(mn)$[2]	• A probabilistic classifier that takes less computational time. • Assumes that a feature is entirely independent of all other present features.	• Assigns 0 probability if some category in the test data set is not present in the training data set. • Stores entire training examples • Need massive data to obtain good results.
Random Forest	1995	[223]	$O(Mm \log n)$[3]	• Composed of many DTs. • Every DT yields a prediction. • The prediction having a maximum number of votes will be the final prediction of the model.	• Computational cost is higher. • Slow prediction generator
ANN	2000	[224]	$O(emnk)$[4]	• Adaptive and composed of Interconnected Artificial Neurons. • Next Layer input depends on Previous Layer Output.	• High cost and time-consuming. • Black-box model hence shows no relation between input and output variable.
Decision Tree	1979	[225]	$O(mn^2)$[5]	• Works on an if-then rule to find the best immediate node. • Continue the process until the predicted class is obtained.	• Difficult to change the data without affecting the overall structure. Complex, expensive and time-consuming.
K-mean	1960	[226]	$O(kmni)$[6]	• Starts from random centroids refine centroids in iterations till the final cluster analysis.	• High dependency on initial centroids. Inefficient clustering for varying cluster sizes
DBN	2006	[227]	$O\left(m\Sigma_l^L(I_l J_l)\right)$[7]	• Higher performance and efficiency is achieved because of the addition of the layers. • Better ability to handle noisy data. • Convenient identification of complex relationships between nodes. • Hidden layers are efficiently used.	• Higher hardware resources consumption. • Higher time consumption because of the addition of the layers. • Unable to provide an explanation for the decisions.
RNN	1982	[228]	-	• Efficiently models sequential data. • Quickly memorize the sequential events • Different various, i.e. LSTM is available.	• Difficult training of the network. • It may face short memory issues while modelling long sequences of data. • Vanishing Gradient and gradient exploding problems.
CNN	1988	[229]	$O(\sum_{l=1}^{d} n_{l-1} \cdot s_l^2 \cdot n_l \cdot m_l^2)$[8]	• Less number of neurons are needed in contrast with traditional NN. • Different variants, e.g. VGG, AlexNet, are available.	• It requires more number of convolutional layers (CL). • A larger tagged dataset is necessary for working.

[1] n=number of instances
[2] n=number of instances, m=number of attributes
[3] n=number of instances, m=number of attributes, M=number of trees
[4] n=number of instances, m=number of attributes, e=number of epochs, k=number of neurons
[5] n=number of instances, m=number of attributes
[6] n=number of instances, m=number of attributes, k=count of clusters, i=iteration count until the threshold is reached
[7] m=number of training samples, I=count of neurons in the input layer, J=count of neurons in the output layer, L=count of RBM models, l=RBM model
[8] l=index of CL, d=count of CLs, n_l= count of filters, s_l=length of filters, m_l=length of the output feature map, n_{l-1}=count of input channels on of l^{th} layer

The classifier is sensitive to the noisy data and the choice of the distance function to find the distance/difference between data points. KNN requires ample storage for manipulation and is computationally expensive. Euclidean distance d(x, y) is typically used as the distance function to calculate the distance between data points x and y.

4) RANDOM FOREST

Random Forest (RF) comes under the category of ensemble learning that combines multiple classifiers to produce a hypothesis of a problem to set up a typical result. It is also termed as a random decision forest and is used for classification and regression purposes. RF is considered as an improved version of CART. RF is typically a collection of prediction results generated by multiple decision trees. The random forest has applications in the literature, such as to measure the volume of spam [217] and in intrusion detection [218]. It gives better performance on non-linear problems and takes less computation cost during the training phase of the model. However, as the random forest combines

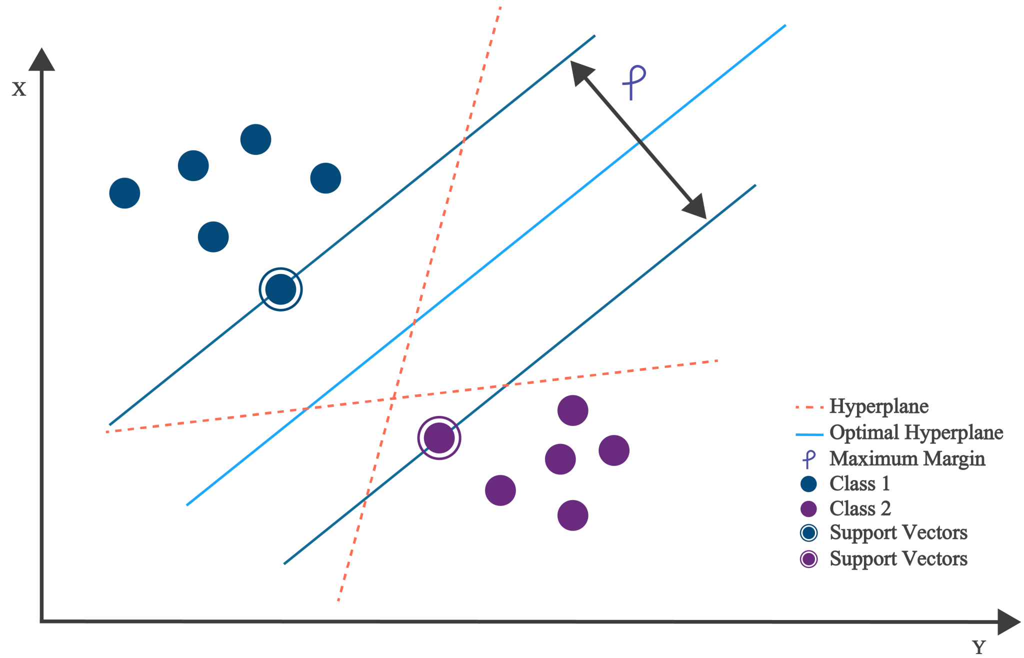


FIGURE 5. Support Vector Machine.

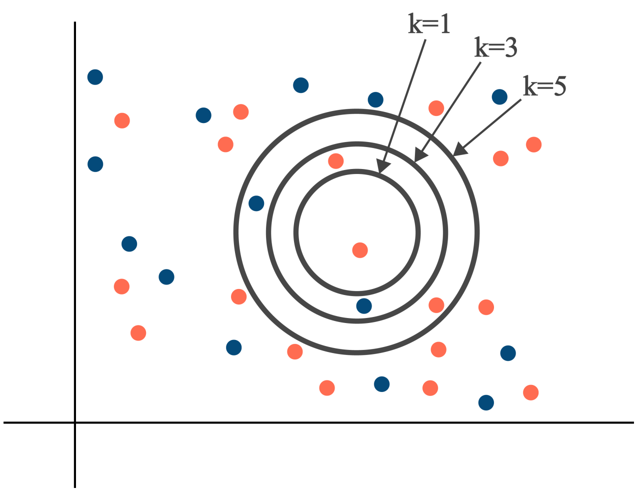


FIGURE 6. K-Nearest Neighbour.

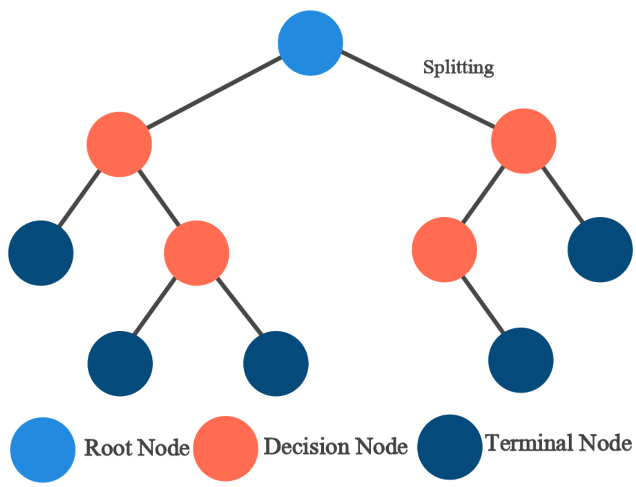


FIGURE 7. Decision Tree.

the prediction of multiple decision trees, there is a need to select the decision trees that should be considered during the prediction process [219].

5) NAÏVE BAYES

Naïve Bayes (NB) is a class of classifier is based on Bayes' theorem, (or Bayes' Rule), which decomposes the conditional probability of a problem being analysed. However, in cyber security this condition of independence does not hold in case of various attack types. Multiple features of a dataset are highly dependable on each other such as features of KDD'99. Hidden NB is an improved version to handle such kind of issues with an accuracy of 99.6% [220]. NB classifier works well with discrete type attributes. This classifier is considered as more straightforward and has a faster detection speed. Three significant techniques fall under Naïve Bayes such as multinomial, Bernoulli, and gaussian. Multinomial Naïve Bayes is used to handle discrete values. Feature vectors in these values represent the number of occurrences in which this event occurs [230]. Bernoulli Naive Bayes is used for the classification of binary feature vectors. Bags of words is an example of such a technique [231]. Gaussian Naïve Bayes is a classifier that is used for continuous values of data. These values are distributed based on Gaussian distribution [232].

6) ARTIFICIAL NEURAL NETWORK

ANN's are trained through a sequence of forward pass and backpropagation cycles. In feedforward, the data are entered into every node of a hidden layer. The activation value is calculated for each node of a hidden layer and output layer. The activation function affects the performance of a classifier. Error is calculated by taking the difference between the network output and the desired value. In backpropagation, this difference is sent back to the input layer to adjust the weights between hidden and output nodes using the Guardian Descent method. This process is repeated until the desired threshold is achieved [233]. ANN is easy to use, considered as robust to noise, a non-linear model but takes much time in training.

Taveras [236] attempted to analyse the importance of password entering practices of end-users in account security. They have suggested improvements in the password entering habits to minimize the risk of account hacking. Their study was done by asking the participants to write down any password of their choice. This study used machine learning algorithms, specifically neural networks, to get the predictions. As an overall result, the study found that neural networks could be used to get the predictions quite effectively, but there were still some limitations. One limitation was that most of the participants were from an information technology background, so the user's behaviour did not follow a logical sequence. Extensive data collection can improve the accuracy of the model and identify the vulnerabilities caused by the password entering habits of end-users.

7) RECURRENT NEURAL NETWORK

A recurrent neural network (RNN) is a branch of neural networks. RNN contains hidden states [228]. Each state uses the output of the previous state as its input, as depicted in Figure 8(a). In this way, information circulates between the states in RNN. The main purpose of the RNN is to process time-series data and analysis of data streams. RNN possesses memory which means it keeps the information from previous experiences and later uses it as an input for the next states [237].

8) CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural network (CNN) is a multi-layer neural work that is an extension of feed-forward ANN [238]. It is comprised of three kinds of layers, including, one or more convolutional layers, one or more fully connected layers, and pooling layers, as depicted in Figure 8(b). ZFNet [239], GoogLeNet [240], and ResNet [241] are commonly used architectures of CNN. It extracts the features at higher resolution and converts them into complex features from higher to coarser resolution. CNN is widely being used in image recognition [242], drug discovery [243], and anomaly detection [244], [245], to name a few. Riaz *et al.* proposed an improved version of CNN for intrusion detection with an accuracy of 99.23% using the KDD99 dataset [246]. CNN has also been used widely for the classification of malicious traffic [247]–[249]. A deep neural network (DNN) was used for passenger profiling in aviation to classify ordinary passengers and potential attackers [250]. Authors in [251] proposed a wavelet-based neural network model to detect cyber security problems.

9) DEEP BELIEF NETWORK

A deep belief network (DBN) is a branch of deep neural networks that follows an unsupervised greedy approach. DBN was generated to simulate the human brain to process complex information and to recognize complex patterns [227]. DBN can be referred to as a stack of Restricted Boltzmann Machine (RBM) with essential generative nature. However, unlike RBM, in DBN, there is no node to node communication within the same layer of the network. Each node of the deep belief network is connected with all the previous and next layer nodes. DBN takes input in the form of probabilities. In DBN, every layer of the network needs to learn complete input to generate output [252]. Each layer keeps generating optimal choices at each step is repeated over and over until the training stage is completed to a desired level, as illustrated in Figure 8(c).

10) AUTOENCODER

Autoencoders are unsupervised neural networks. It reduces the input size and dimensions of the data by decomposing, compressing the data, and by eliminating the noise in the data. Also, the original shape of the input can be regained by applying the reconstruction process. Autoencoder follows a principle that targeted output values should be equal to the original input values. An autoencoder consists of four main parts. First, an encoder is used to learn how to compress the data. Secondly, the bottleneck is a layer that is used to hold the fully compressed data. Moreover, by using the decoder, the model learns how to perform data reconstruction. Finally, in the fourth part, reconstruction loss gauges how much the output is close to the targeted output values [206].

11) REINFORCEMENT LEARNING

Reinforcement learning (RL) is another subdomain of machine learning. Sometimes, RL is also termed as learning with a critic because there is input to the algorithms against any wrong prediction. However, it has not been told to the algorithm of how to correct it. Instead, the algorithm has to figure out and try several possibilities until it learns the correct answer [195]. This phenomenon works based on a reward and penalty scheme. Deep learning methods and RL are combined together to solve many complex problems. An example of this technique is AlphaGo [196], [197]. Deep Reinforcement learning is used in cyber security such as intrusion detection on host [253], defending DDoS attacks [254], detection of phishing emails [255], and cyber-physical system [256], to name a few. RL is considered the technique that is closest to the modeling how human reasoning is understood to occur by exploiting the unknown

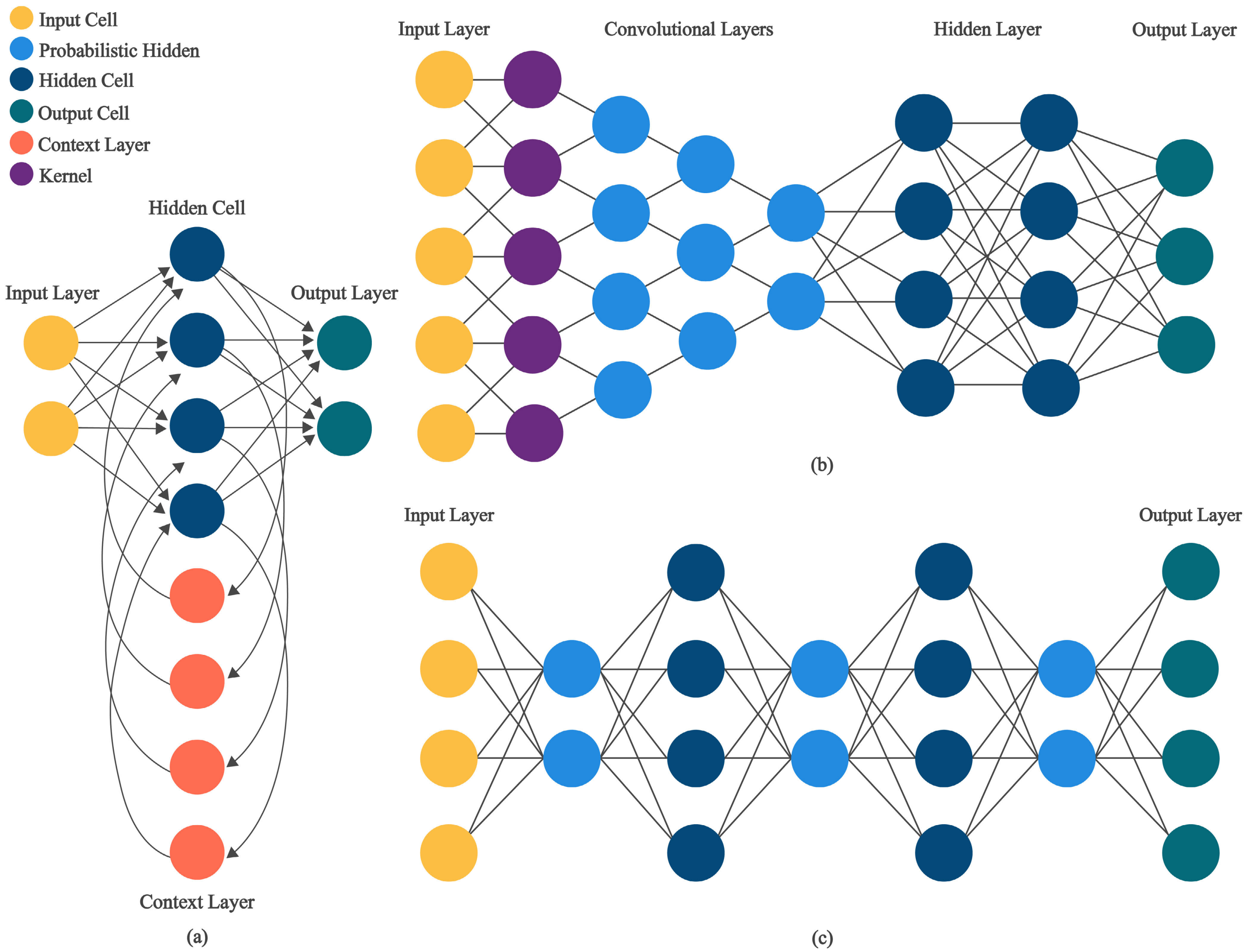


FIGURE 8. A Graphical Representation of Various Neural Architectures (a) Recurrent neural network (RNN) (b) Convolutional Neural Network (CNN) (c) Deep Belief Network (DBN).

and new environment. The working of RL is composed of five components, namely, agent, environment, reward, state, and reward, as depicted in Figure 7. An agent formulates its own learning experiences through direct interaction with the environment. The two changes have occurred as a result of this interaction. Firstly, the state of the environment is changed into a new state. Secondly, the environment imposed a penalty or a reward based on the action. Given a state, the reward function tells the agent how good or bad action has been performed. The agent learns from the reward and filters out the bad action.

D. COMMON MACHINE LEARNING AND DEEP LEARNING TOOLS

Machine learning techniques are being applied in various fields to solve real-life problems. In this section, we provide a brief description of the popular tools used for machine learning and deep learning.

1) Weka [257]: This is a commonly used machine learning tool that can be used for regression, clustering, visualization, and other data analytics related tasks. This is a freely available tool that is provided with online support and can work on Mac, Linux, and Windows platforms.
2) Caffe [258]: This is considered as one of the early and significant industry-level tools in the field of deep learning. This tool is specialized in the area of image processing. This tool trains models directly without explicitly writing the code. However, it requires coding in the case of adding new layers. This is an open source with faster runtime and mobile-supported.
3) Torch [259]: This tool is implemented in C and Lua languages. It supports many ML algorithms. Facebook and Twitter also adopted this framework because this tool is fast running and provides excellent flexibility. This tool has included several pre-trained models and provided easiness in writing code for new layers. It is well documented and easy to debug. This is also optimized with GPUs. However, it does not provide any visualization tool.

4) Keras [260]: This tool offers more extendibility with fast prototyping. This tool is written in Python, so it does not need any files for model configuration. This is compatible and provides support for both convolutional neural networks and recurrent neural networks.
5) TensorFlow [261]: This is an open-source library provided by Google. This tool is compatible with classic machine learning techniques and uses a data flow graphical structure. This tool supports multiple GPU and provides faster compilation, portability, and distributed training. This tool also provides mobile supported, distributed training, and a visualization tool (TensorBoard). However, it needs more significant memory for execution, difficulty in debugging, and packages are heavier.
6) Theano [262]: This tool was developed in Python. It supports a recursive network. This tool is portable and provides much flexibility for other DL packages. However, the compilation process is slower and has difficulty in modifying the code for the developer.
7) Shogun [263]: This tool can work well with more massive datasets and supports various ML tasks such as regression, classification, and clustering. This was developed using the C++ programming language and is freely available for use.
8) Accord.Net [264]: This is a freely available tool that provides most libraries for audio and image processing. However, it supports only the work implemented in .Net. It provides algorithms for statistical work and graph plotting.
9) MXNET [265]: This tool is written in c++ that is lightweight and memory efficient. This is highly scalable and provides mobile support. However, this tool provides a less user base and not easy to learn.

There are other tools available that are used to develop mobile systems, including, RapidMiner, Chainer, Lasagne [266], Blocks [267], Deeplearning4j [268], and CNTK [269]. However, for a beginner who intends to apply deep learning models in the networking domain, PyTorch is a recommended tool. It is easy to build a neural network using PyTorch. TensorFlow is recommended for the implementation of advanced operations and large-scale implementation. CoreML [270], ncnn [271], and DeepSence [272] are recommended DL platforms for mobile devices.

IV. CURRENT STATE OF MACHINE LEARNING FOR CYBER SECURITY

Cyber security promises to provide a defence against cyberattacks and threats to cyberspace. There are various aspects of cyber security, including detection and classification of malicious URL, financial fraud, spam classification, IDS, malicious domain generation, probing, cyber extortion, and malware, to name a few. Furthermore, with the drastic growth of mobile devices and networks are the targets of cybercriminals besides computer networks. To the best of our knowledge, there does not exist any survey that targeted any aspect of the attack on both computer networks and mobile devices in one place. Figure 10 presents the major areas of cyber security, attacks on cyberspace along with the list of significant ML references targeting that specific class of attack. Cyber security overlaps with other components of cyberspace, including Internet security, network security, and ICT security.

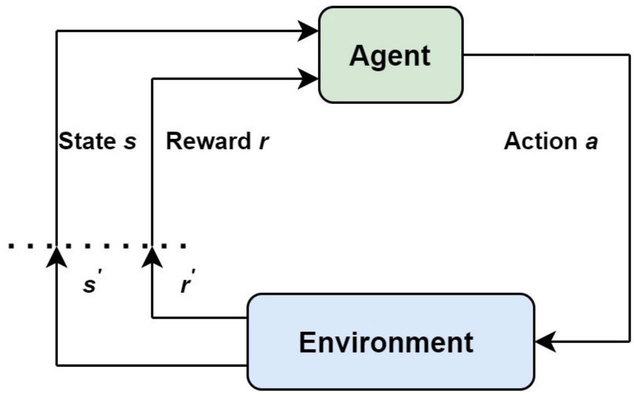


FIGURE 9. Reinforcement Learning.

We have targeted three significant challenges (detection and classification of IDS, spam, and malware) to cyber security in which ML techniques are playing an important role. We have further elaborated on these threats on mobile devices and computer networks. The intrusion detection system on a computer network is further sub-divided into signature-based/misuse-based, anomaly-based, and hybrid-based techniques. Sub-types of intrusion are further categorized into either applied on a host or a computer network. Spam detection is further elaborated with respect to the medium, including image, email, SMS, video, and Twitter. Malware is also explored regarding static analysis and dynamic analysis. ML techniques are being implemented in the literature to handle various types of cyberattacks.

ML is one of the possible solutions to act quickly against cyberattacks. ML techniques are employed to deal with such matters because the learning techniques can learn from experiences and respond to newer attacks promptly. We have mentioned the references of a few articles that deal with such kind of cyberattack. The following sub-headings elaborate on each cyber threat to the computer network and mobile network and how the state-of-the-art ML techniques are playing their roles to fight against these cyber threats.

A. SPAM DETECTION ON COMPUTER NETWORK

1) BACKGROUND

Electronic mail, usually termed as 'Email' or 'E-mail', is a method of information sharing among individuals using electronic devices through the Internet. It is commonly used as a service and becoming popular nowadays. An irrelevant, unsolicited and unwanted email, massively used for marketing that annoys the user is called a spam email [29], [273], or called ham otherwise. Spam email consumes bandwidth, storage,

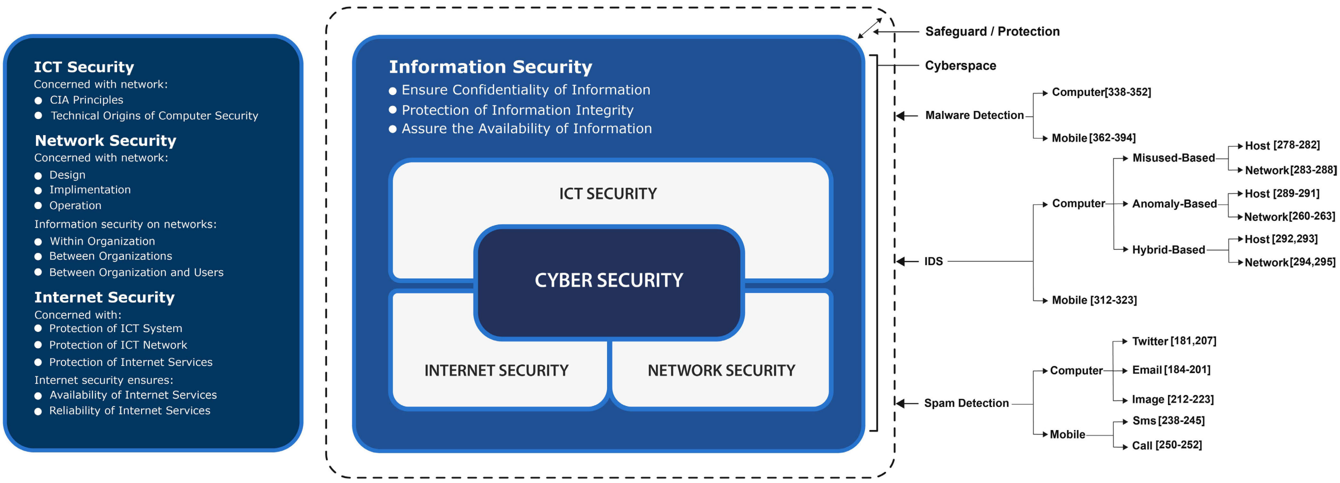


FIGURE 10. Graphical summary of threats to cyberspace and reference of ML techniques to fight against these attacks.

and time of Internet users and significantly decreases the efficiency of system and network [274], [275]. Nowadays, more than 85% of received emails or messages are spam [184]. Emails and web search engines are considered as the early victims for spam attackers. Email spam is not the only affected area, spam has proliferated in different media such as mobile devices, blogs, newsgroups, instant messaging, calls, video sites. Facebook, Twitter, YouTube, and other social platforms have given the liberty to contribute and share the content freely, which has stimulated the spammers to exploit them for their benefits. It has taken the attention of information scientists to provide quick and needful solutions to it. The process to classify an email as either ham or spam and rule out unsolicited emails is called spam filtering [276]–[278]. Numerous spam filtering techniques have been proposed in the literature. However, they are inefficient as spammers are smart enough to alter the spam words. Anti-spamming or spam combating techniques are a set of measures that are taken against an array of spam attacks not to hamper the productivity of targeted media [279].

2) TRENDS

Machine learning techniques are being postulated to improve efficiency and counter the spammer's attack. Several ML techniques have been proposed in the literature for spam classification [273]–[277], [279], [280], spam filtering [278], [281] and spam identification [282], [283]. ML techniques have been applied in the different domains under spam detection, such as Twitter, image-based, email, and blogs. Every domain has a different best-suited classifier. However, in most of the studies, the SVM technique has shown better accuracy than other classifiers. Some authors applied feature selection methods followed by any classifier to improve the accuracy of the classifier significantly. Moreover, combining multiple classifiers to improve the classification accuracy can be a future area of research. Commonly used ML techniques are decision tree, J48, Naïve Bayes, SVM, and Random Forest. Deep learning techniques such as deep belief network (DBN) and clustering techniques have also been applied for spam filtering and detection. Table 7 presents the summary of various machine learning models, their performance evaluations, and used dataset over a decade.

3) TECHNIQUES AND METHODS

The signature-based technique is a traditional spam filtering technique used to identify malicious behaviour by the signature. Nevertheless, it has a poor detection rate in fighting new spam attacks [281]. A brief account of techniques applied to fight against spam on social media can be found in [282]. Though many email programs have embedded with essential filtration utility, a user can purchase filtration software to have extra protection and control. Collaborative filtering [284], machine learning [285], and blacklisting [286] methods are also used to achieve the same results. In [283], the authors provided various spam filtering tools and techniques. [287] further elaborated primary methods used to script injection, URL shorteners, clickjacking, and malicious browser extensions for spam filtering. Spambase, Enron, PU Datasets, and Ling-Spam have commonly used datasets for spam classification and filtering [288]–[291]. The following sections will discuss the applications of ML models to detect and classify spam on Twitter, images, videos, email, and blogs.

a: ML AND SPAM ON EMAILS

Emails are considered as a common entry point for any malicious software. A wrong click on any malicious URL written on email can place computing devices and networks in danger. There is a high dimensionality of feature space because the email and documents contain hundreds to thousands of words. Finding the optimal subset of the most prominent features is called feature selection [292]. Feature selection can significantly improve the accuracy and applicability of the learning and classification process [293], [294]. Feature

TABLE 7. A comparision and summary of ML models for spam detection over a decade.

Published Year	Ref.	Dataset	Sub-Domain	Learning Model	Results		
					Accuracy	Precision	Recall
2010	[273]	Spambase	Email Spam	RF	95.43%	-	-
2010	[302]	Customized	Email Spam	NB	96%	-	-
2010	[303]	Enron	Email Spam	SVM	-	-	-
2011	[153]	Spambase	Email Spam	SVM, NB	SVM 96.90%, NB 99.46%	SVM 93.12%, NB 99.66%	SVM 95%, NB 98.46%
2011	[304]	Spam-Archive	Image Spam	ANN	93.70%	87%	94%
2011	[305]	Customized	Spam Tweets	RF	95%	95.70%	95.70%
2011	[153]	Spam Assassin	Email Spam	NB, SVM, KNN	99.46%	-	-
2011	[306]	Customized	URL	-	94.14%	-	-
2013	[307]	Spambase	Email Spam	DT	92.34%	93.90%	93.50%
2013	[308]	Spambase	Hybrid	ANN	93.71%	95%	
2013	[309]	Spambase	Email Spam	RF	99.54%		
2013	[310]	Customized	Emails	NB	85.96%	-	-
2014	[311]	SMS Collection	SMS Spam	SVM, DT	SVM 98.61%, DT 96.60%	SVM 98.60%, DT 96.50%	SVM 98.60%, DT 96.60%
2014	[312]	Spambase	Email Spam	DT	92.08%	91.51%	88.08%
2014	[313]	Spambase	Email Spam	DT	94.27%	-	91.02%
2015	[152]	Spambase	Email Spam	SVM, NB	SVM 79.50%, NB 76.24%	SVM 79.02%, NB 70.59%	SVM 68.67%, NB 72.05%
2015	[314]	Twitter Dataset	Spam Tweets	SVM	95.20%	-	93.60%
2015	[315]	Spambase	Email Spam	NB	84%	89%	78%
2015	[316]	Customized	Email Spam	J48, NB, ID3	J48 89.3%, NB 91.4%, ID3 93.6%	-	-
2016	[317]	Enron	Email Spam	DT	96%	98%	94%
2016	[318]	TARASSUL	Email Spam	DBN	96.40%	95.31%	93.59%
2016	[319]	Enron	Email Spam	DBN	95.86%	96.49%	95.61%
2016	[209]	Spambase	Email Spam	ANN	91%	-	-
2016	[320]	Twitter Dataset	Spam Tweets	RF	96.20%	98.60%	75.50%
2016	[321]	Customized	Images	CNN, SVM	75.11%	-	-
2018	[322]	Twitter Dataset	Spam Tweets	SVM	93.14%	92.91%	93.14%
2018	[151]	Spambase	Email Spam	DBN	89.20%	96%	-
2018	[323]	Spambase	Email Spam	ANN	92.41%	92.40%	92.40%
2018	[324]	Customized	Images	SVM	97%	-	-
2019	[325]	Customized	Tweets	DNN	86.2%	-	-
2019	[326]	Enron	Email Spam	TF-IDF+PCA+(SVM,RF)	-	SVM 98.10%, RF 97.60%	SVM 98.10%, RF 97.60%
2019	[327]	Customized	SMS Spam	KNN, ANN, NB	KNN 90.4%, ANN 97.4%, NB 97.67%	KNN 91.37%, ANN 97.41%, NB 97.64%	KNN 90.4%, ANN 97.4%, NB 97.67%
2020	[27]	Twitter Dataset	Spam Tweets	SVM	98.88%	-	94.47%
2020	[328]	UCI Repository	SMS	LSTM, CNN	LSTM 95.33%, CNN 99.44%	-	-
2020	[329]	Customized	Emails Spam	PCA+SVM, PCA+NB, PCA+RF	SVM 84%, NB 85.2%, RF 93.4%	SVM 83.7%, NB 84.8%, RF 93.3%	SVM 84%, NB 85.2%, RF 93.4%
2020	[330]	Customized	SMS Spam	LR, KNN, DT	LR 99%, KNN 95%, DT 98%	LR 93%, KNN 80%, DT 95%	LR 86%, KNN 60%, DT 86%

selection techniques obtained better accuracy than different similar methods [273], [276]. Authors in [252] compared deep belief networks with SVM on three different datasets to filter spam emails. DBNs outperformed with slightly better accuracy of up to 1% more than SVM for all datasets. However, there is a lack of benchmark datasets for spam detection. Authors in [295] provided a comparative study of various decision tree classifiers such as AD Tree, Decision Stump, and REP Tree. They claimed that Rep Tree provided the highest accuracy for email spam classification.

J48, Bayes Net, and SVM were used for the detection of spam emails in [291], where SVM performed the best among these approaches. Comparatively, J48 performed better in [291], [296], [297] whereas SVM showed the worst performance in [291], [298]–[300] for spam email classification.

b: ML AND SPAM ON BLOG

Authors in [301] used logistic regression to detect blog spam on a dataset gathered from social media comments.

Instead of detecting individual spams, authors in [331] detected spam campaigns and clustered them with an accuracy of above 80%. Random Forest and Decision Tree techniques were used to identify the bookmarking sites having location information. They reported an accuracy of 89.2% with Decision Tree and 89.76% with Random Forest [332]. Authors in [333] applied Naïve Bayes (NB), k-NN, and SVM for spam detection and concluded that NB and SVM performed better. Others compared ten classification techniques on a single benchmark dataset and reported an accuracy of 95.45% using SVM as the best among all [334].

c: ML AND SPAM ON TWITTER

The proliferation of Twitter users contributes to the growth of spam tweets. Spam tweets are unsolicited and unwanted tweets contain malicious code leading to other security threats like phishing, scams, drug sales, or malware downloads, etc. Authors in [314] evaluated various ML techniques for streaming spam tweets. They have found that NB performed better with an accuracy of 97.3%. Authors in [277] applied Decision Tree, Random Forest, and NB techniques and obtained better accuracy with the Random Forest classifier.

d: ML AND SPAM ON IMAGES

Spam detection methods are categorized into two major categories, namely textual based and image-based analysis. Text analysis tools are ineffective in detecting image-based spam which is a subsequent target of spammers [335]–[337]. Authors in [338] applied various pattern recognition and computer vision techniques to detect image-based spam. Further similar studies can be found in [339]–[343].

e: ML AND SPAM ON VIDEOS

Apart from textual based and image-based venues, authors also applied ML techniques to detect spam blogs ('splogs') and video spams. SVM is a commonly used ML technique to detect spam on blogs [344]–[348] and video [349], [350]. Decision Tree was further used in [274], [313], [351]–[355] for spam classification. Authors in [356] applied Firefly and Bays classifiers for spam detection. Clustering techniques were used in [275] for spam detection.

f: ML AND GOOD WORD ATTACKS

Researchers have also investigated the problem of 'good word attacks'. The good word attacks are commonly used to fool the filtration process to classify between spam and legitimate email. Jorgensen *et al.* [357] proposed a counterattack strategy using multiple instance learning. Their approach divided an email text into a bag of multiple segments. They considered each segment in a bag as a separate instance and claimed that their technique of multiple instances is more robust than a single instance against good word attacks. Good word attacks have also been investigated in [358]–[360] for spam filtering.

4) TOOLS

There are several anti-spam tools available in the market to protect nuisance and unsolicited emails. Some available anti-spam tools are SolarWinds MSP Mail Assure [361], SpamTitan [362], SPAMfighter [363], and ZEROSPAM [364].

B. SPAM DETECTION ON MOBILE DEVICES

1) BACKGROUND

Mobile devices and services are getting immensely popular nowadays. Mobile services such as short message service (SMS), email apps, images, data, mobile clouds, and calls are also the victims of spammers. SMS is considered as a straightforward and inexpensive approach for phishing attacks. Smartphones are enclosed with personal information such as debit/credit card details, sign in details such as user name/passwords, and so on [365]. Free services, advertising, promotions, packages, and awards are typical examples of spam SMS [366].

2) TRENDS

ML techniques are playing a vital role to detect and identify the spams on mobile device such as spams in SMS, calls, email apps, data on mobile, images, and videos. Researchers applied SVM, Naïve Bayes, kNN, RNN, and k-means machine learning techniques for spam detection. As a prominent association rule algorithm, Apriori was also used for classifying spams on SMS [367]. Spambase and Enron have been commonly used datasets for spam classification. NB and SVM performed better in most of the experiments to classify spams on emails. Overall, ML techniques improved the accuracy of distinguishing spam or not spam calls, SMS, and emails.

3) TECHNIQUES AND METHODS

Unwanted and unsolicited SMS can be detected with techniques involved user participation or content-based methods [237], [368]. Techniques included user participation are rarely used because they work with user feedback and experience sharing. In contrast, content-based techniques were based on text and content analysis. The filtration method for unwanted SMS is similar to that of spam email. However, SMS contains up to 160 characters comprised of languages slags, short text, and Internet abbreviations [335], [369]. The following sections will discuss the applications of ML models to detect and classify spam spams in SMS, calls, email apps, data on mobile device, images, and videos.

a: ML AND SPAM ON SMS

Bayesian learning methods were applied in [370] for spam SMS filtration. Authors in [367] used NB with the Apriori algorithm for SMS classification. NB, KNN, and SVM were used in [371] for the detection of unwanted SMS. Authors in [372] proposed a filtration approach that used KNN on a rough set in the first phase and again applied KNN in the second phase. A comparison of various ML techniques was performed in [373]. It concluded that NB outperformed other ML techniques. Hybrid NB on data from three users with six different datasets in the Enron Spam dataset was tested. They applied a local classifier for each user, followed by a global classifier. They claimed that their hybrid method performed better than individual NB [374]. Others have applied Twitter-LDA to filter spam SMS and achieved a better accuracy of 96.49% [375].

Authors in [237] applied a Bayesian-based classifier to distinguish spam or ham mobile-based messages. Authors in [368] used recurrent neural networks (RNN) for the classification of unwanted and normal messages and obtained an accuracy of 98%. They have also proposed Spanish and American based SMS spam datasets. K-Means clustering algorithm was used in [365] for filtering spam SMS. NB, SVM, and Decision Tree were used for spam SMS filtering, where SVM outperformed with 85% accuracy to filter spam SMS [376].

b: ML AND SPAM ON IMAGES

The sharing of images using various communications applications such as Instagram, WhatsApp, and Facebook has grown exponentially. Several studies for image spam filtering and classification can be found in [377]–[381]. Authors in [382] proposed an ML technique to classify and delete spam images.

c: ML AND MALICIOUS CALLS

Malicious calls, including scams and spams over the telephone, are challenging issues for the last few years that cost billions of dollars globally. Authors in [383] used SVM, Random Forests, and Logistic Regression to detect the spam call and reduced the malicious call by 90%.

C. INTRUSION DETECTION ON COMPUTER NETWORK

1) BACKGROUND

Cyber analytics for intrusion detection system is broadly classified into three main categories. They are misuse-based, anomaly-based, and hybrid-based detections. Misuse-based detection is used for the detection of known attacks. Anomaly-based detection monitors the normal behaviour and differentiates the abnormal behaviour of network and system. Lastly, the hybrid-based detection approach combines both misuse-based and anomaly-based techniques to improve the accuracy of detection [73]. Attackers may successfully exploit the flaws ubiquitously existed in these traditional defence approaches, hence protection of the user from unknown and evolving threats is questionable. Cyberinfrastructure has an enormous amount of data. Criminals attempt to gain unauthorised access to the data. Learning the patterns and behaviours of intrusion and attacks is very critical. Therefore, machine learning techniques play a vital role to detect and predict future intrusion and attacks promptly.

2) TRENDS

ML techniques are widely being used to detect intrusion. Commonly used approaches are ANN, Fuzzy association, SVM, decision tree, and statistical models. Case-based reasoning and various unsupervised learning techniques are also applied to improve the accuracy and detection rate of intrusion. Various classifiers have shown better performance than other classifiers in different domains and tasks in ID. However, early and prompt detection of new and zero-day attacks is still a challenging area of research. Various machine learning techniques were applied for misuse-based detection [127], [384]–[388], anomaly-based detection [389]–[392] and hybrid-based detection [393]–[397]. Some papers summarized intrusion detection techniques and ML techniques in detail [3], [73], [77], [398]–[406]. DARPA, KDD 99 are commonly used but outdated datasets. Many researchers have mentioned different metrics to evaluate the accuracy of any applied classifier. However, there should be a standard metric and the latest benchmark datasets to evaluate any classification model. Table 8 and Table 9 present an overview of the performance evaluation of various ML techniques to detect intrusion over a decade.

3) TECHNIQUES AND METHODS

Cyber security attacks on cyberspace can be on two levels: network-based and host-based. Cyber defence system provides a defence mechanism on both levels. Controlling the network flow is the responsibility of the network-based defence system. However, a host-based defence system combats against upcoming data in a workstation/computer by a firewall and other defence mechanisms installed on a host [407], [408]. As discussed in section II-A, there are four major categories of attacks for instruction detection purposes, including Denial of Service (DoS), Phishing/Scanning/Probe, Remote to Local (R2L), and User to Root (U2R). The following sections summarize the crossover between ML models and the attacks for intrusion detection.

a: ML AND DOS ATTACKS

Different ML techniques were applied to detect DoS attacks such as Decision Tree with an accuracy of 97.24% [409], Neural Networks with an accuracy of 97% [410], Naïve Bayes with an accuracy of 96.65% [409] and SVM with an accuracy of 91.6% [411].

b: ML AND PROBE ATTACKS

Naïve Bayes, Fuzzy Association, Decision Tree, Neural Network, and SVM were applied to detect probe attack with an accuracy of 88.83%, 88.50%, 77.92%, 71.63%, and 36.65% respectively [409], [410], [412].

c: ML AND R2L ATTACKS

With the KDD dataset, R2L attacks were detected with Neural Net, SVM, and Naïve Bayes where the Neural Net obtained maximal accuracy of 26.68% [409]–[411].

d: ML AND U2R ATTACKS

ML techniques were also applied to identify User to Root attacks where Fuzzy association, SVM, DT, and NB achieved an accuracy of 68.60%, 12%, 13.60%, and 11.84% respectively.

e: ML AND HOST-BASED ATTACKS

ML techniques were applied to detect attacks on host and computer networks. Machine learning techniques such as Rule-based, ANN, Fuzzy association rules, and different statistical methods were applied to detect the misuse-based attacks on a host [413]–[417]. Statistical models, association

TABLE 8. A comparision and summary of ML models for intrusion detection over a decade.

Published Year	Ref.	Dataset	Sub-Domain	Learning Model	Attack Types	Results		
						Accuracy	Precision	Recall
2010	[423]	DARPA	Anomaly-Based	NB	-	91.60%	-	61.60%
2010	[394]	KDD 99	-	ANN, Fuzzy Clustering	DoS, U2R, Probing, R2L	96.71%	91.32%	99.08%
2010	[424]	KDD 99	Anomaly-Based	Logistic Regression	DoS, U2R, Probing, R2L	98.68%	99.08%	91.32%
2010	[425]	KDD 99	-	NN	DoS, U2R, Probing, R2L	99.99%	-	-
2010	[426]	DARPA 1998	-	SVM	-	80.1%	81.1%	-
2011	[395]	KDD CUP99	Hybrid-Based	SVM	DoS, U2R, Probing, R2L	95.72%	-	-
2011	[427]	Customized	Anomaly-Based	One Class SVM	DDoS UDP, DDoS TCP	91.5%	-	-
2011	[428]	KDD 99	-	AdaBoost, NB	DoS, U2R, Probing, R2L	Adaboost 99.92%, NB 99.55%	-	-
2011	[429]	KDD 99	Anomaly-Based	K-NN	DDoS	97.42%		-
2011	[384]	KDD 99	-	NB, DT, NN	DoS, Probing	NB 78.2%, DT 99.4%, NN 98.6%	-	-
2011	[430]	KDD 99	Anomaly-Based	K-Means	DoS, U2R, Probing, R2L	86.4%	-	-
2012	[431]	KDD CUP99	Anomaly-Based	ANN	-	62.90%		
2012	[432]	NSL-KDD	Anomaly-Based	NB	-	99%	83%	78.90%
2012	[433]	KDD 99	-	NB	DoS, U2R, Probing, R2L	78.32#	-	-
2012	[386]	KDD 99	Anomaly-Based	SVM, DT	DoS, U2R, Probing, R2L	SVM 99.03%, DT 98.85%	-	-
2012	[434]	KDD 99	-	RF, C4.5	DoS, U2R, Probing, R2L	RF 89.21%, C4.5 921%	-	-
2012	[435]	KDD 99	-	SVM, K-Means	DoS, U2R, Probing, R2L	SVM 98.62%	-	-
2013	[436]	KDD 99	-	SVM, Fuzzy NN	DoS, U2R, Probing, R2L	-	-	-
2013	[437]	KDD 99	-	MLP, K- Means	DoS, U2R, Probing, R2L	MLP 100%, K-means 97.17%	-	-
2013	[438]	KDD 99	-	NN	DoS, U2R, Probing, R2L	96.23%	-	-
2013	[439]	ISCX	Anomaly-Based	NB, K-Means	Attack, Normal	NB 88.28%, K-Mean 99.03%	NB 85.07%, K-Means 98.84%	NB 99.70%, K-Means 99.71%
2013	[440]	KDD 99	-	Group methods using Ensemble	DoS, U2R, Probing, R2L	90%	-	-
2014	[441]	NSL-KDD	Hybrid-Based	SVM	-	82.37%	74%	82%
2014	[442]	DARPA	Anomaly-Based	SVM	-	95.11%		
2014	[121]	KDD CUP99	Hybrid-Based	SVM	-	99.30%	-	-
2014	[443]	KDD 99, NSL KDD	-	ANN	DoS, U2R, Probing, R2L	KDD 99 99.41%, NSL-KDD 97.76%	-	-
2014	[397]	NSL –KDD	Hybrid-Based	DT, One Class SVM	-	-	-	-
2014	[444]	KDD 99	Anomaly-Based	K-Medoids	DoS, U2R, Probing, R2L	Acc 96.38% Dos 96.12%, U2R 70.51%, Probe 70.13%, R2L 90.10%	-	-
2014	[445]	KDD 99	-	SVM	DoS, U2R, Probing, R2L	95.3%	-	-
2014	[450]	NSL-KDD	Anomaly-Based	ANN	-	97.53%	-	-
2015	[446]	KDD Cup 99	-	RBF- SVM	-	99.9%	-	-
2015	[447]	KDD 99	-	DT	DoS, U2R, Probing, R2L	91%	-	-
2015	[448]	KDD CUP99	Hybrid-Based	SVM	-	96.08%	-	-
2015	[449]	NSL-KDD	Misuse-Based	NB	-	81.66%	-	-
2015	[391]	KDD 99	-	KNN, K-Means	DoS, U2R, Probing, R2L	80.65%	-	80.32%
2016	[450]	KDD Cup 99	-	LSTM	-	99.8%	-	-
2016	[451]	KDD 99	-	PCA, K-NN	DoS, U2R, Probing, R2L	-	PCA 97.80%, KNN 93.20%	PCA 51.20%, KNN 50%

TABLE 9. *(Continued.)* **A comparision and summary of ML models for intrusion detection over a decade.**

Published Year	Ref.	Dataset	Sub-Domain	Learning Model	Attack Types	Results		
						Accuracy	Precision	Recall
2016	[452]	NSL-KDD	Anomaly-Based	SVM, PCA, NB, MLP	DoS, U2R, Probing, R2L	SVM 99.63%, PCA 97.35%, NB 95.16%, MLP 97.16	-	SVM 99.16%, PCA 97.98%, NB 91.65%, MLP 96.77%
2016	[453]	KDD CUP 99	Anomaly-Based	RF	-		98.10%	98.10%
2016	[454]	NSL-KDD	Anomaly-Based	SVM	-	98.89%	-	--
2017	[249]	ISCX		CNN	-		97.3%	
2017	[455]	NSL-KDD	Hybrid-Based	RF	-	97.10%		
2017	[86]	NSL-KDD	Anomaly-Based	DBN	-	90.40%	88.60%	95.30%
2017	[456]	KDD	Hybrid-Based	DT	-	99.85%	99.70%	98.10%
2017	[457]	KDD 99	-	K-NN	DoS, U2R, Probing, R2L	DoS 99.21%, U2R 99.62%, Probing 92.93%, R2L 99.01%	-	-
2017	[458]	KDD 99	-	K-NN, SVM, K-Means	DoS, U2R, Probing, R2L	-	99.68%	-
2017	[459]	KDD Cup 99	-	LSTM	-	97.54%	98.95%	-
2018	[406]	KDD	Misuse-Based	DT	-	99.96%		-
2018	[460]	KDD CUP99	Hybrid-Based	DT	-	92.87%	99.90%	-
2018	[406]	DARPA	Misuse-Based	ANN	-	99.82%		-
2018	[461]	UNSW-15	Anomaly-Based	ANN	DoS, U2R, Probing, R2L	92.40%	-	-
2018	[462]	KDD 99	-	NB, AdaBoost, RF	DoS, U2R, Probing, R2L	NB 91.03%, Adaboost 99.89%, RF 99.93%	-	-
2019	[463]	NSL-KDD	Anomaly-Based	SVM	-	89.70%	-	-
2019	[464]	KDD 99	-	SVM, NB, ANN	DoS, U2R, Probing, R2L	95.03%		95.23%
2019	[465]	NSL-KDD	Anomaly-Based	RF	-	95.10%	92.50%	
2019	[466]	NSL-KDD	Anomaly-Based	DBN	-	99.45%	99.20%	99.70%
2019	[467]	NSL-KDD	Anomaly-Based	ANN	-	94.50%		
2019	[410]	NSL-KDD	Hybrid-Based	RF	-	75.30%	81.40%	75.30%
2019	[468]	CICIDS	-	RF, Gradient Boosting Tree	DoS, DDoS	-	-	-
2019	[469]	ISCX 2012	-	SVM, MLP, PCA	DoS, U2R, Probing, R2L	SVM 87.02%, IBK 94.29%, MLP 82.42%	SVM 90.10%, IBK 91.40%, MLP 87.20%	SVM 87.00%, IBK 99.60%, MLP 82.40%
2019	[403]	KDD 99, NSL-KDD, UNSW-NB 15	-	NB, SVM, DR, RF	DoS, U2R, Probing, R2L, DDoS	NB 92.90%, SVM 80.10%, RF 78.40%	NB 99.90%, SVM 69.20%, RF 94.40%	NB 91.40%, SVM 96.90%, RF 72.50%
2019	[470]	NSL-KDD	-	DT, MLP, SVM, KNN	DoS, U2R, Probing, R2L	DT 97.14%, MLP 97.02%, SVM 97.42%, KNN 96.51%	-	DT 95.57%, MLP 95.80%, SVM 96.81%, KNN 94.79%
2020	[471]	Customized	-	AdaBoost, J48, SVM, NB	DDoS	Adaboost 93.40%, J48 90.30%, SVM 85.30%, NB 73.10%	-	Adaboost 93.40%, J48 90.20%, SVM 85.20%, NB 70.50%
2020	[472]	NSL-KDD	-	Deep Neural Network	DoS, U2R, Probing, R2L	95.40%	96.20%	93.50%
2020	[473]	KDD-99	-	NB, DT, RF	DoS, U2R, Probing, R2L	99.80%	99.80%	

rules, ANN, and KNN, were used to detect the anomaly-based detection techniques on a host [418]–[420]. For the hybrid-based intrusion, ANN and association rules were applied over the host [421], [422].

f: ML AND NETWORK-BASED ATTACKS

SVM and Decision Tree were applied to identify the misuse-based attacks on a network [474]–[479]. Random Forest and ANN were applied on the network for hybrid-based intrusion detection [480], [481]. Teodoro [482] used machine learning and knowledge-based approaches for anomaly-based intrusion detection.

Nguyen [483] presented the ML methods that classify the Internet traffic for any cyber data, and Internet Protocol (IP) flows. Others used Fuzzy Logic, ANN for their application in intrusion detection [484]. Case-based reasoning is an approach that provides the solution to new problems based on the solutions saved of previous similar problems. The solution of similar past problem cases is then used as a starting point for solving an existing problem [485]. Mansour [486] proposed a case-based reasoning approach for intrusion detection.

Apart from supervised ML techniques, various unsupervised and semi-supervised techniques were implemented to detect anomalies such as clustering algorithms in [390], SVM in [487], and neural networks in [488]. Others have applied deep learning models [489] to detect anomalies in airports and feature optimization techniques for intrusion detection system [490].

4) TOOLS

There are various tools available in the market for intrusion detection. Intrusion detection tools are developed to handle the intrusion either on the host or network. A network intrusion detection system (NIDS) is used to detect the intrusion on a network. Host intrusion detection system (HIDS) is used to detect the signature-based or anomaly-based intrusion on a host. Various ID tools are available for free. However, others are costly. McAfee NSP [491], Hillstone NIPS [492], Huawei NIP [493], Palo Alto [494], Dark Trace [495], and Cisco Firepower NGIPS [496] are popular commercial tools for ID. Free tools include Snort [497], Suricata [498], Samhain [499], Security Onion [500] and Sagan [501]. The usage of tools depends upon the operating system, detection type (HIDS, NIDS), or detection method (signature-based, anomaly-based). Trusted Automated eXchange of Intelligence Information (TAXII) is another tool to prevent and mitigate cyberattacks. TAXII uses Structured Threat Information eXpression (STIX), a language developed to describe the information of cyber threats, to define how the services and messages exchange become a mean of sharing threat information [502].

D. INTRUSION SYSTEM ON MOBILE DEVICES

1) BACKGROUND

Mobile devices are capable of performing many sophisticated tasks. Smart devices are also facing a growing number of threats every day [503], [504]. Currently, networks provide higher transmission rates from 100 Mbps to 10+ Gbps in wired networks. Due to this high volume of data, IDS could not effectively work to gather and analyse network traffic. Snort, a Deep Packet Inspection (DPI) can work properly on a wired network to handle data up to 1 Gbps and discard after 1.5 Gbps [505]. Replaying, traffic analysis, and spoofing are general examples of attacks on a mobile network [506].

2) TRENDS

ML techniques such as supervised ANN, Decision tree, MLP, and SVM are commonly used to detect the intrusion on a mobile network. Decision tree and deep learning approaches performed better than other classifiers. Machine learning techniques evolved to purpose new ways of intrusion detection due to the increase of bandwidth [73, 204].

3) TECHNIQUES AND METHODS

Attacks on the mobile network are classified into two major categories, namely, active attacks and passive attacks. An attack that involves information modification and disrupts the standard functionality of a network to get access and decrease network performance is called an active attack. In contrast, passive attacks do not disrupt the normal flow of the network but scan the network to get any valuable information [507].

a: ML AND ANOMALOUS BEHAVIOUR

Bayes decision rules were applied in [508] to increase the security in cellular networks. Authors in [509] used supervised ANN to detect malicious behaviour such as service fraud on mobile communication. ANN and probabilistic models were applied in [510] to identify the anomalies in usage with 69% TPR. ANN is further used in [511]–[513] to detect the anomalies in mobile network communication. The authors in [513] proposed a malware detection system called VirusMeter to identify the anomalous behaviours and compared their system with ANN and decision tree. Self-organizing maps and clustering techniques were applied to detect anomalous behaviour with the conclusion that both methods were suitable for network monitoring [514]. To observe the accuracy of detecting the misuse-based behaviour of users on mobile device, a comparison was made among the BN, KNN, and Random Forest techniques in [515].

Decision Tree, KNN, MLP, and SVM were applied to detect intrusion on mobile devices with decision tree outperformed with an accuracy of 97.04% [516]. SVM was used to detect intrusion on a mobile network and achieved similar performance as of system without intrusion [517]. A deep learning approach was proposed to detect cyberattacks with an accuracy of 90.99% [518].

4) TOOLS

There are various applications available in the market to protect the Android system. Some of them are free of charge to use, and other quality applications charge an annual fee. Bitdefender [519], Trend Micro [520], and BullGuard [521] are commercial apps available to protect the Android system by taking an annual fee. In contrast, Sophos [522], Trustlook [523], and PSafe [524] are examples of ID applications available for free to use but with limited features.

E. MALWARE DETECTION ON COMPUTER NETWORK

1) BACKGROUND

Malicious software, commonly termed as 'Malware', is a piece of code that is covertly inserted into a computer system or network with the intention to disrupt the user activities. Malware compromises and challenges the integrity, confidentiality, and availability of the victim's information on hardware or software. Malware is a combination of 'mal' from 'malicious' and 'ware' from 'software'. Viruses, Worms, Trojan Horses, Spyware, and Adware, are commonly taken examples of it [525], [526].

The objective of cybercriminals is to exploit the vulnerabilities of a computer system or network. Cybercriminals execute malicious code on the victim's device and propagate it into other devices or networks. The count of known malware samples crossed 800 million according to McAfee's technical report of 2019 [527]. Since the last few years, malware is increasing rapidly, creating financial loss from billons to trillion [528], [529]. Not only individuals are the main target of malware but also are the industries and military disrupted through trained hackers and customized malware [530]. Malware is considered as a top security risk for companies [531].

2) TRENDS

In the past, signature-based techniques were used to detect malware. These techniques do not perform well to detect zero-day or advanced malware attacks. Machine learning techniques are not only capable of identifying zero-day attacks but also outperform in detecting new or obfuscated malware attacks [70], [532]–[536]. SVM is the most studied ML classification approach used to detect malware with 29% usage, followed by a decision tree with 17% usage [514]. DBN, in combination with other semi-supervised learning techniques, also improved the accuracy of detection. Tables 10 gives the summary of performance evaluation results of ML models applied to detect malware over a decade.

3) TECHNIQUES AND METHODS

Malware is classified into two generations. In the first generation, malware has the same structure. Whereas in the second generation, it changes its structure and evolves into a new variant while the actions remain the same [537].

Encrypted, Oligormorphic, Polymorphic, and Metamorphic malware are the further classifications of the second generation based on the evolution of structure. Changes in the structure of malware are random and unpredictable [581].

a: ML AND FEATURE SELECTION

Feature selection provided better accuracy when using ML techniques. Authors in [544], [582]–[584] applied feature selection and claimed better accuracy in the detection of malware. Kolter [585] evaluated the datasets by applying a decision tree, TF-IDF, and support vector machine, with a decision tree outperformed. The decision tree was also used with a hierarchical feature extraction algorithm in [582]. Authors in [586] used the AdaBoostM1 and decision tree classification techniques and reported 90% malware detection accuracy. Authors in [587] claimed that there was no false alarm using their hyper-grams technique for malware detection. The semi-supervised method obtained an accuracy of 86% in [588], whereas others achieved 95.9% accuracy with SVM [544]. SVM was further used for malware detection in [589], [590]. Authors in [591] proposed a new method with an accuracy of 97.95% to detect unknown malware. Authors in [592] proposed a new dataset called CA and Mal2017 with 80 features and showed 87% recall for traffic classification for detection.

b: ML AND ZERO-DAY MALWARE

Pierra *et al.* proposed a technique to identify zero-day attacks [593]. Principal Component Analysis (PCA) and ANN were proposed to detect and classify AI-based cyber-attacks and successfully obtained an accuracy of 90% [594]. DBN was applied in [595] to detect malware. Other authors in [596] have combined DBN with semi-supervised techniques to achieve better accuracy.

c: ML AND ADVERSARIAL INPUTS

Adversarial malware samples can easily bypass the ML techniques that were applied to detect malware. Machine learning techniques were not primarily designed to work with cyber security so an evasion can easily fool the ML [597]–[599]. Research is going on to provide a solution by having adversarial training [600]–[604].

4) TOOLS

There are various tools available in the market for malware detection. However, choosing the right tool is critical. Some tools are available free of charge, and a few charge annual subscription fees. Avast Internet Security [605] is a mostly used anti-malware tool that has taken 15.21% of the total market size [606]. Other frequently used tools are Malwarebytes [607], Norton Power Eraser [608], AVG [609], and Bitdefender Antivirus [610].

F. MALWARE DETECTION ON MOBILE DEVICES

1) BACKGROUND

Due to the increasing use of E-commerce, mobile banking and mobile transactions, threats to mobile device are also increased. Hence, mobile devices are getting more vulnerable to threats than computers. Data values and banking details are as vulnerable on mobile device as on computer [611].

2) TRENDS

Authors in [554] provided a performance evaluation of supervised, semi-supervised, and unsupervised techniques. They concluded that unsupervised learning techniques had shown better accuracy to detect malware on Android devices. The authors had parallelly combined several classifiers and claimed to achieve better accuracy while combining classifiers. SVM, KNN, Random forest, decision tree are commonly used techniques to detect malware in mobile device and networks. Feature selection followed by a classification technique also helps to improve the accuracy of any classifier.

3) TECHNIQUES AND METHODS

Malware detection techniques for mobile devices can be categorized into three major groups, namely static, dynamic,

TABLE 10. **A comparision and summary of ML models for malware detection over a decade.**

Published Year	Ref.	Dataset	Sub-Domain	Learning Model	Attack Types	Results		
						Accuracy	Precision	Recall
2011	[538]	Customized	Hybrid	n-gram, Markov chain	-	94.41%	-	-
2011	[539]	Customized	Dynamic	-	Mobile Malware		-	-
2012	[540]	SMOTE	Static	DT	-	96.62%	-	-
2012	[540]	VX Heavens	Hybrid	ANN	-	88.89%	88.89%	-
2012	[540]	VX Heavens	Static	ANN	-	92.19%	-	-
2013	[543]	Malware Dataset	Dynamic	SVM	-	95%	-	-
2013	[544]	Malware Dataset	Static	DT	-	92.34%	-	93%
2013	[545]	Malware Dataset	Dynamic	DT	-	88.47%	-	-
2013	[546]	VX Heavens	Static	ANN	-	88.31%	-	-
2013	[547]	NSL-KDD	Hybrid	NB	-	99.50%	-	-
2014	[548]	Malware Dataset	Hybrid	NB	-		97.50%	67.40%
2014	[549]	Customized	Static	PART	Malicious Intend	95.8%	-	-
2014	[69]	Customized	Static	J48, NB, RF	Mobile	MLP :83%	-	-
2015	[550]	Malware Dataset	Dynamic	SVM	-	97.10%		
2015	[551]	KDD CUP99	Hybrid	DBN	-	91.40%		95.34%
2015	[552]	VX Heaven	Static	NB	-	88.80%		
2015	[553]	Malware Dataset	Hybrid	NB	-	95.90%	95.90%	95.90%
2016	[317]	Customized	Static	SVM	-	91%	84.74%	100%
2016	[554]	Customized	Static	DT	-	99.90%	99.40%	
2016	[555]	Customized	Static	DBN	-	89.03%	83%	98.18%
2016	[556]	Comodo	Static	ANN	-	92.02%	-	-
2016	[557]	Malware Dataset	Dynamic	RF	-	96.14%	-	-
2016	[558]	Drebin	Dynamic	RF, NB, SVM, LR	-	RF: 99.49%	-	-
2017	[559]	Malware Dataset	Static	SVM	-	94.37%	-	-
2017	[560]	Customized	Static	DT	-	84.7%	-	-
2017	[561]	Malware Dataset	Hybrid	RF	-	91.40%	89.80%	91.10%
2017	[562]	Moledroid Apps	Dynamic	RF	Information Theft	99.1%	-	-
2017	[563]	Contagio	Hybrid	CNN	API Calls	99.4%	-	-
2017	[564]	Comodo Cloud		DBN	API Calls	96.66%	-	-
2018	[565]	Customized	Static	SVM	-	89.91%	88.84%	
2018	[566]	Customized	Dynamic	SVM	-	96.27%	96.16%	93.71%
2018	[567]	SMOTE	Dynamic	DT	-	92.82%		
2018	[568]	Customized	Dynamic	ANN	-	82.79%		
2018	[569]	Drebin	Hybrid	RF	Mobile	99.07%	-	
2018	[570]	VirusShare	-	ANN			-	98.29%
2018	[571]	Drebin	Static	CNN	Code Analysis	95.4%	-	-
2018	[572]	Drebin	Dynamic/Static	DNN	System Calls	95%	-	-
2019	[573]	Customized	Static	SVM		95.17%	95.57%	95%
2019	[574]	Customized	-	KNN, DT, SVM, RF	Malicious Samples	KNN 94.68%, DT 99.37%	SVM 92%, RF 96%	KNN 95%, RF 96%
2019	[575]	Customized	-	J48, MLP	Hardware-Assisted	J48 93.2%, MLP 94.7%	-	-
2019	[576]	Contagio Dump, VirusShare	Static	Adaboost	Android Apps	99.11%	99.33%	99.36%
2020	[577]	Customized	-	J48,RF,Adaboost	Android Apps	J48 76.2%, RF 7.6%, Adaboost 75.4%	J48 76.8%, RF 73.5%, Adaboost 75.8%	J48 77.6%, RF 71.6%, Adaboost 75.9%
2020	[578]	Android Malware Dataset	-	LSTM	API Calls	97.22%	-	-
2020	[579]	Leopard Mobile dataset		Deep CNN	IoT Devices	-	98.79%	98.79%
2020	[580]	Drebin	Hybrid	Graph CN	Android Malware	99.69%	99.57%	99.82%

and hybrid groups. Static detection is a detection technique in which an application is observed for malicious patterns without execution. In contrast, dynamic detection is carried out by running the actual app to check the dynamic behaviour [612]–[614]. Hybrid malware technique is the malware detection technique that combines static and dynamic analysis to detect malware [615], [616].

a: ML AND FEATURE SELECTION

Others [617] have proposed a novel method to group the related flow behaviours into bags and then applied a supervised detection method and achieved a precision of 90%. Authors in [618] applied SVM to train their model with existing attacks and predicted future attacks. Decision Tree, KNN, and SVM were used on the model represented with opcode-sequence-frequency, achieving an accuracy of 90% [544]. Random Forest, SVM, Logistic Regression, and Naïve bays were used for malware detection, with Random Forest outperforming in the aspect of TPR/FPR [619].

b: ML AND ANDROID

Existing malware detection techniques performed excellently on Android fixed datasets but could not get a high detection rate with real-world problems. Using permission and API call, SVM, J48, and Bagging were used to detect malware in Android-based applications and obtained 96.39% accuracy with bagging [620]. Another author also used permission features and SVM to classify Android malware [621], [622].

Authors in [623] used Information Gain to identify the essential features. They applied C4.5 Decision Tree, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), and k-Nearest Neighbour techniques for malware classification. HOSBAD is a K-NN based Android malware detection system that is used to discriminate malicious and benign applications [624]. Naïve Bayes technique showed better accuracy than other classification models to detect malware in [625]–[627].

c: ML MODELS AND DETECTION TECHNIQUES

Authors in [628]–[630] categorized Android detection techniques for static and dynamic analysis and reviewed different methods. Authors in [631] extracted the critical features by performing static and dynamic analysis on the application and applied SVM with an accuracy of 95%. SVM was further used for malware detection in [583], [632]–[638].

DeepFlow, a deep learning model based on DBN architecture, was proposed to detect Android malware and achieved the highest F1 score comparing to other ML techniques [639]. Authors in [640] considered Android business and tool applications and identified malicious apps with a recall of 71% using the K-mean technique.

d: PARALLEL COMBINATION OF ML MODELS

Authors in [549] proposed a parallel combination of Decision Tree, Simple Logistic Regression, Naïve Bayes, PART, and RIDOR algorithms and claimed to achieve better accuracy than evaluating the classifiers individually. Authors in [555] used DBN architecture to construct a deep learning model and compared detection accuracy with SVM, C4.5, and Logistic Regression. The authors concluded that the deep learning model outperformed other machine learning models with an accuracy of 96.76%.

Ucci [80] presented a survey on malware analysis using different ML techniques and provided a relationship between the ML techniques used in the analysis procedure, the type of features extracted from samples, and the objective of the analysis. They stated that there was no sufficient dataset that was publically available and could be used for specific purposes. They emphasized that new proposed techniques should be tested on recent data. Otherwise, new methods would not be useful in real-world problems [80].

4) TOOLS

Kaspersky mobile antivirus [641], Norton Security and Antivirus [642], and Avira Antivirus Security [643] are considered as high-end mobile device malware detection tools.

V. CURRENT CHALLENGES OF USING ML TECHNIQUES FOR CYBER SECURITY

A. CHALLENGES FOR MACHINE LEARNING MODELS

Machine learning techniques are commonly used in the area of cyber security. However, there are various challenges in this direction. ML techniques need a considerable amount of high-performance resources and data while training the models. A solution is to use multiple GPUs, which is neither a power-efficient or cost-effective solution. Moreover, ML techniques are not designed to detect cybercrimes. Cyber security was not a focus of traditional ML techniques. There is a need to have powerful and robust ML techniques that are specifically designed to deal with security attacks and handle adversarial inputs. It should be pointed out that one ML model cannot perform well to detect various security attacks. There should be a particular ML model specially designed to deal with a specific type of cyberattack. Prevention of attack at an early stage is another challenging task. There should be capabilities in ML techniques to detect those real-time and zero-day attacks in a short interval.

Machine learning models were applied for decision making in terrorism detection or diagnosis of disease in the medical field. In these cases, prediction cannot be used to blind faith to avoid catastrophic consequences [644]. When machine learning techniques are used in life-critical or mission-critical applications (e.g., self-driving cars, cyber security, surgical robotics), it is crucial to ensure that they provide some high-level correctness guarantees instead of speed and accuracy [645]. Trustworthy machine learning is the secure use of machine learning techniques for cyberspace. The trustworthiness of a classifier can be elaborated in two ways: (1) trusting a prediction, i.e. whether a user trusts on a specific prediction model to take a particular action, and (2) trusting a model, i.e. whether the user will trust on a model deployed as a tool in rational ways.

Authors in [646] investigated the problem of dataset shift where the model was trained and tested with different datasets. Further, they have suggested that avoiding the dataset shift can be done by removing the leaked data or changing the training data. It helps to identify what must be done to convert an untrustworthy model into a trustworthy one. Classical linear/shallow learning tends to be more trustworthy but slower or less accurate. Deep learning is relatively opaque and complex, despite a rapidly developing theory.

The evolution of cell phones and the global positioning system provide opportunities for forensic science and epidemic control to identify the location information of specific moving objects. Nevertheless, due to the possibility of errors or tempered information on mobile devices, maintaining the trustworthiness of the particular object is a challenging task. Chenyun [647] proposed an approach to assess the similarity among the gathered information from multiple sources about the location of a particular object. The trustworthiness of location data gathered from the trajectories of moving objects always has the possibility of uncertainty. This uncertainty arises because the objects are moving their location, and due to the network delays [648]. Authors in [649] have proposed an approach of trust ontology to help the service providers and consumers for trustworthy interaction in an online web system.

Trustworthiness is also applied in natural language processing (NLP) for text classification, especially when a message is passed in life-critical missions. Evidentially, the trustworthiness should be incorporated where the text meaning is interpreted in both practical and semantic terms to achieve the best trustworthiness detection result [650]. Others have proposed a metric model to verify the trustworthiness of software [651].

ML techniques have applications in the energy sector in which power-aware strategies were designed to reduce the power consumption for data centers and companies [652]. An idle machine will be turned off dynamically to decrease the overall power consumption. The correct prediction of an idle machine will surely reduce energy consumption. The trustworthiness of the prediction model in scheduling which machine to turn off is very crucial. The sensitivity of detecting an alarm will lead to a higher false alarm rate is called alarm fatigue. The higher frequency of false alarms has left an adverse impact on security staff and resulted in missing the critical alarm or slow response time. This phenomenon is a challenging research question in cyber security [653], [654].

B. OTHER CHALLENGES OF USING ML FOR CYBER SECURITY

We have reviewed the state-of-the-art algorithms and techniques of machine learning that were used to tackle cybercrimes such as IDS, spam, and malware, as depicted in Table 2. Many other discrepancies and issues are exposed as well, making it a firm base for discussing more future challenges and trends. Some of these issues are discussed below.

1) DATASETS

We have provided an overview of the famous and commonly used datasets in Table 3. There is an issue uncovered in this direction, i.e. most of the datasets are outdated. The number of features and categories for each dataset is different. Most of the information related to data and attacks is redundant. Machine learning models perform better in case there is a large volume of data available for training, which is not the case for currently available datasets. There should be benchmarks and standard datasets that have a massive amount of data for training and testing purposes and have balanced and an equal number of attack categories. For a security system, data are collected from multiple sources of social media and traditional sources such as web or database access. The volume and heterogeneity of data sources collected from these numerous sources are also a challenge for ML models for cyber security. Due to privacy and security issues, most of the datasets that represent the latest attacks are private. Conversely, the publically available datasets are laboriously anonymized and suffer from various issues. In particular, these datasets do not typically exhibit real-world and latest attacks. Due to these issues, the exemplary and latest benchmark dataset yet to be discerned.

2) EVALUATIONS METRICS

We have provided different evaluation metrics in section II-C to evaluate a classifier. However, most of the researchers have used different parameters to evaluate a classification model and ignore another side of the picture, even on the same dataset. There is a need to consider an agreed standard set of metrics for model's comparison for further improvements.

3) DETECTION AND TIME COMPLEXITY OF VARIOUS TECHNIQUES

Little consideration of the real-time environment of attacks was made in the literature. The detection rate of an attack within a real-time environment and time complexity of an algorithm should also be considered. Cybercriminals evolve new attacks every day to expose the vulnerabilities of the network. The efficiency of the detection of an attack is a crucial point to consider. If there is a false positive in the system, security analysts will spend time investigating the activity that is not malicious. Security analysts will lack confidence in the system in case of more false alarms. The computational complexity of each ML model should also be considered. We have provided the time complexity of frequent ML models in Table 6. Moreover, improving the detection speed and computational cost by using advanced hardware through a distributed approach can be a future area of research.

4) ADVERSARIAL INPUTS TO ML MODELS

Authors in [655] described numerous challenges to test several machine learning models. In the military, quick action has to be taken against a message. An attacker can modify the sent message by adding adversarial text sequences. This modification can change the whole sense of message and lead to a disaster [656]. The training of the model in the adversarial setting is an essential factor that can be helpful to make an ML model more robust against adversarial inputs. A defensive mechanism DeepCloak was proposed to identify and remove unnecessary features in a deep neural network (DNN) model. DeepCloak limits the capacity of an attacker to generate adversarial samples and therefore increases the robustness of the model [657]. A model Goodfellow [658] was claimed to be robust against adversarial inputs. It is a common assumption that test data are from the same distribution as with the model is trained. This assumption is often violated. For instance, a camera that was used to take images for the model at training time might be different from the camera that was used to take images for the model at the testing time. Hence, the performance of the prediction model will suffer. Tony *et al.* in [659], [660] have described various adversarial attacks that can easily fool the learning process of ML models. Ibitoye *et al.* in [661] proposed a new model to identify the risk of adversarial attacks in network security. They have also provided the evaluation of different adversarial attacks to ML models applied in network security. Deep learning models that are considered robust to noise and adversarial examples for cyber security are imperative but remaining challenging.

5) ADVERSARIAL ATTACKS AND DEFENCES

In contrast, if the cyber attacker influences the data during deployment to fool the already trained model by manipulating the attack samples, then the attack is called an evasion attack [662]. There are various types of adversarial attacks including Fast Gradient Sign Method (FGSM), multi-step Bit Coordinate Ascent (BCAk), multi-step Bit Gradient Ascent (BGAk), Generative adversarial networks (GAN), Carlini & Wagner attack (C&W), to name a few. To counteract against the adversarial attacks, there are various defence strategies have also been proposed in the literature, namely Adversarial training [663], defensive distillation [664], feature squeezing [665], and Magnet [666]. In adversarial training, the adversarial examples are added during the training phase. It is easy to implement but requires retraining of the model. It is most useful where the attacks during the testing time for a deployed model are the same as during training.

Defensive distillation requires retrained of the model but most effective for most of the dataset. It requires the neural network distillation for the training of a new network model same as of the original one. Feature squeezing is considered a better approach on multiple image datasets (e.g. ImageNet, MNIST) to combat various adversarial attacks. This technique compresses (pixels in their case) by using multiple compression methods. In case the prediction of the original sample and the compressed sample is substantially different, then the compressed sample is considered as an adversarial sample. Magnet does not require the retraining of the model but uses the autoencoder to detect any adversarial sample.

6) GROWING AND NEW ATTACKS

With the progress of cyber security, the attack's evolution is growing at a rapid pace. There are two challenges in applying ML to handle such new attacks. Firstly, the ML models are applied to locate such activities that may not be previously seen [667]. Secondly, newer attacks are often technically different from older ones. Models are usually trained with more past features in a dataset. New attacks can have a different feature set. The latest attacks may evade from classifiers and generate a false alarm or reduce the detection rate.

7) CONFIDENTIALITY AND PROTECTION OF DATA

The security and privacy-related issues were elevated because the data are collected from both structured and non-structured sources. This leads to the problem of securing big data versus big data for security [668]. It is mandatory to assure the protection of data from adversarial attacks and being tempered by illegitimate users. Access to data should also be allowed to legitimate users.

VI. CONCLUSION

Cyber security has become a matter of concern globally in achieving enhancements in security measures to detect and react against cyberattacks. The previously used conventional security systems are no longer sufficient because those systems lack efficiency in detecting previously unseen and polymorphic attacks. Machine learning techniques are playing a vital role in numerous applications of cyber security systems. Our review here has revealed a rapidly growing interest in machine learning and cyber security in the academics and industry, which has resulted in a growing number of publications, particularly in the last decade. In this paper, we have bridged the gap between ML techniques and threats to computer networks and mobile communication by presenting a comprehensive survey of the crossovers between the two areas. This survey presents the literature review on machine learning techniques for intrusion detection, spam detection, and malware detection on computer networks and mobile device in the last decade.

This paper briefly presents the applications of machine learning models in the field of cyber security, mainly on the advancement of the last ten years. There are peculiarities of each cyber threat that make it difficult even for the state-of-the-art ML model in dealing with such cyberattacks. It is impossible to provide one recommendation for all the attacks, based on one model. Various criteria such as detection rate, time complexity, classification time to detect new and zero-day attacks, and accuracy of an ML model should be considered while selecting a particular model to detect a cyberattack. We have described the basics of cyber security such as the classification of cyberattacks on mobile device and computer networks. Due to the significance of ML, we have also described the foundations of machine learning, subtypes, and significant techniques for a beginner to get a better insight into this area. We are unaware of any work that discusses the applications of ML techniques in cyber security domain both on mobile device and computer networks in one paper. We have depicted a graphical summary of the attacks threatened to cyberspace and existing ML techniques to fight against these cybercrimes. We have presented an overview of several popular ML tools. We have also given the evaluation metrics to evaluate the working of any classifier.

Dataset is very crucial for the training and testing of ML models. We have presented a description of commonly used security datasets. There is the unavailability of representative and benchmark datasets for each threat domain. Machine learning techniques were not primarily designed to work with cyber security. Evasion can easily fool the ML model by giving adversarial inputs. Trustworthy machine learning is the secure use of machine learning techniques for cyberspace to provide some high-level correctness guarantees instead of speed and accuracy of the model. We have also briefly summarized some of the significant challenges of using machine learning techniques in cyber security as well as given an extensive bibliography in this area. The mentioned challenges are worthy of attention for future research.

APPENDIX

In this appendix, we have explicitly provided analysis to show the trends of machine learning and cyber security using the Scopus database. The search string was "Machine learning" and ("Cyber security" or "Cybersecurity"). A total number of 1200 documents were retrieved from 595 multiple sources, including 325 journals articles, 8 books, 34 book chapters, 744 conference papers, 52 conference reviews articles,

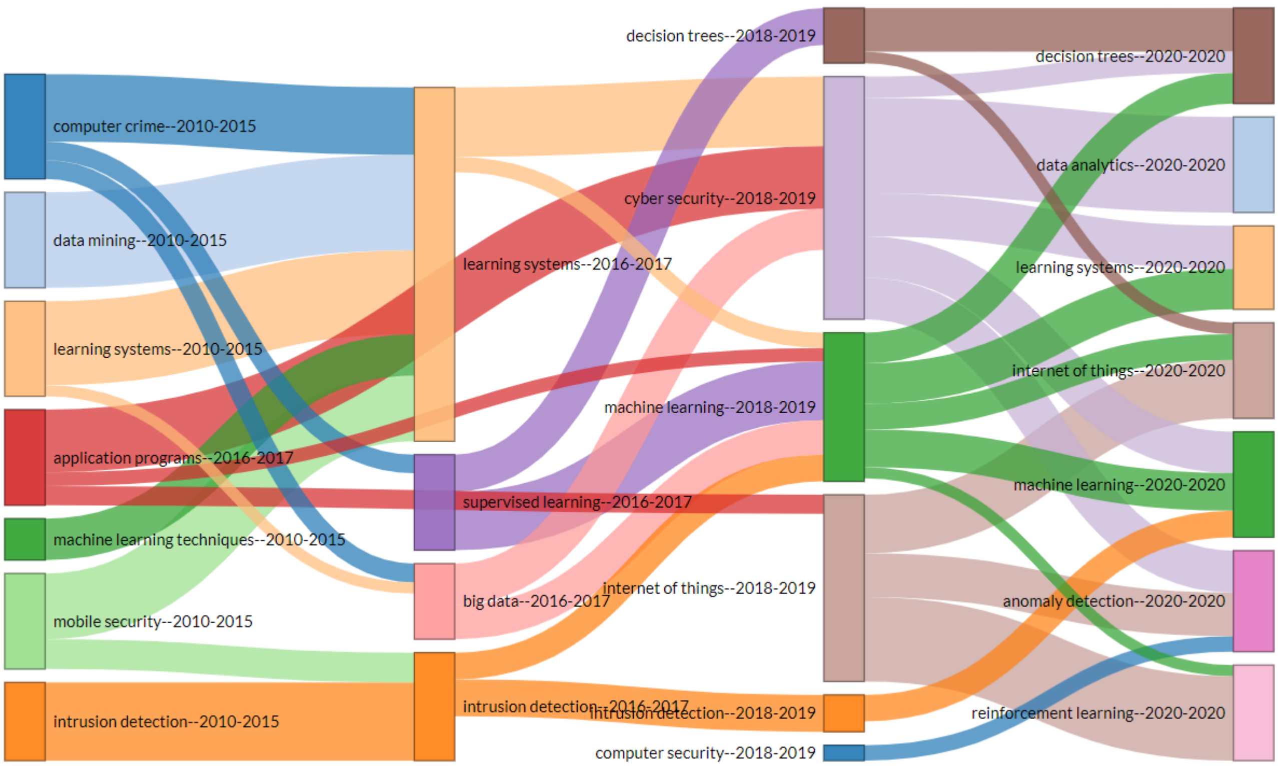


FIGURE 11. Thematic Evolution in the Four Segments.

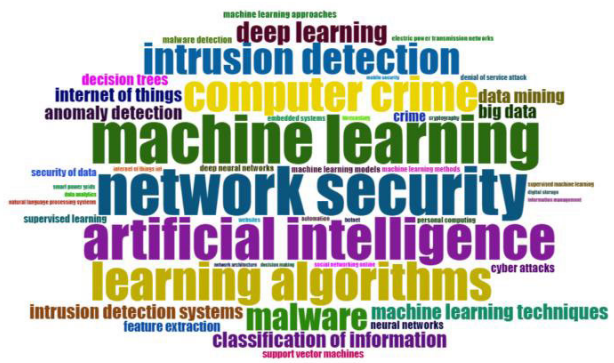


FIGURE 12. Word Cloud.

TABLE 11. The country affiliation of corresponding authors.

Country	Articles	Freq	SCP	MCP	MCP_Ratio
USA	255	0.35815	240	15	0.0588
INDIA	88	0.1236	80	8	0.0909
CANADA	35	0.04916	33	2	0.0571
UK	30	0.04213	22	8	0.2667
CHINA	24	0.03371	14	10	0.4167
AUSTRALIA	21	0.02949	17	4	0.1905
KOREA	20	0.02809	16	4	0.2
JAPAN	18	0.02528	17	1	0.0556
MALAYSIA	13	0.01826	11	2	0.1538
NORWAY	12	0.01685	9	3	0.25

34 journal review articles, and 3 editorials. There were total of 3182 distinct authors who contributed towards these 1200 articles, including 130 single-author documents. The collaboration index of these authors was 2.8.

Figure 11 illustrates a word cloud of commonly used words in the literature of machine learning and cyber security. The size of a word corresponds to the occurrence of the word it has been found in the documents. Figure 12 provides the thematic evolution of how this field evolves over the last decade. We have divided the timeline into four segments. The first segment is from 2010 to 2015, the second segment is from 2016 to 2017, the third segment is 2018 to 2019, and the fourth one is in 2020. As depicted in Figure 12, from 2010 to 2015, the literature was about computer crime, data mining, learning systems, machine learning techniques, mobile device security, and intrusion detection. From 2016 to 2017, research in learning systems got increased, and topics related to computer crime, mobile device security, learning system, and machine learning techniques evolved into learning systems. Currently, the literature is evolved in multiple topics including, decision trees, data analytics, machine learning, anomaly detection, IoT, and reinforcement learning.

Table 11 provides the list of top 10 corresponding authors' countries. The USA is on the top with 255 articles from which the 240 are single country publication (SCP), and 15 are multiple country publication (MCP). MCP involves at least a foreign author. India ranked on the second position with 88 articles including 80 articles as SCP, and 8 with MCP.

Amrita School of Engineering, India is the most relevant affiliation with 28 articles followed by the Swinburne University of Technology, Australia with 15 articles. IEEE Access is the most cited source with more than 400 documents, followed by the Computer Security journal with 210 documents. Machine learning is the most common word with the occurrence of 553, cyber security with 601, artificial intelligence with 307, computer crime with 257, and learning algorithms with 278.

REFERENCES

[1] *ICT Fact and Figures 2017*. Accessed: Jun. 1, 2020. [Online]. Available: https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2017.pdf

[2] ICT Facts and Figures, International Telecommunication Union. (2017). *Telecommunication Development Bureau*. [Online]. Available: https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx (accessed Oct. 09, 2019).

[3] D. K. Bhattacharyya and J. K. Kalita, *Network Anomaly Detection: A Machine Learning Perspective*. London, U.K.: Chapman & Hall, 2013.

[4] V. Ambalavanan, "Cyber threats detection and mitigation using machine learning," in *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*. Hershey, PA, USA: IGI Global, 2020, pp. 132–149.

[5] T. Thomas, A. P. Vijayaraghavan, and S. Emmanuel, "Machine learning and cybersecurity," in *Machine Learning Approaches in Cyber Security Analytics*. Singapore: Springer, 2020, pp. 37–47. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-15-1706-8_3

[6] *The Comprehensive National Cybersecurity Initiative*. Accessed: Jun. 1, 2020. [Online]. Available: https://obamawhitehouse.archives.gov/issues/foreign-policy/cybersecurity/national-initiative

[7] *The White House, Remarks by APHSCT Lisa O. Monaco at the International Conference on Cyber Security*. Accessed: Oct. 17, 2019. [Online]. Available: https://obamawhitehouse.archives.gov/the-press-office/2016/07/26/remarks-aphsct-lisa-o-monaco-international-conference-cyber-security

[8] *10 Years After the Landmark Attack on Estonia, Is the World Better Prepared for Cyber Threats?* Accessed: Jun. 1, 2020. [Online]. Available: https://foreignpolicy.com/2017/04/27/10-years-after-the-landmark-attack-on-estonia-is-the-world-better-prepared-for-cyber-threats/

[9] North Atlantic Treaty Organization. (Apr. 3, 2008). *Bucharest Summit Declaration. Issued by the Heads of State and Government Participating in the Meeting of the North Atlantic Council in Bucharest*. Accessed: Oct. 9, 2019. [Online]. Available: https://www.nato.int/cps/en/natolive/official_texts_8443.htm

[10] F. Farahmand, S. B. Navathe, P. H. Enslow, and G. P. Sharp, "Managing vulnerabilities of information systems to security incidents," in *Proc. 5th Int. Conf. Electron. Commerce (ICEC)*, 2003, pp. 348–354.

[11] D. Craigen, N. Diakun-Thibault, and R. Purse, "Defining cybersecurity," *Technol. Innov. Manage. Rev.*, vol. 4, no. 10, pp. 13–21, Oct. 2014.

[12] P. Szor, *The Art of Computer Virus Research and Defense: ART COMP VIRUS RES DEFENSE_p1*. London, U.K.: Pearson, 2005.

[13] I. Firdausi, C. Lim, A. Erwin, and A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection," in *Proc. 2nd Int. Conf. Adv. Comput., Control, Telecommun. Technol.*, Dec. 2010, pp. 201–203.

[14] S. Gu, B. T. Kelly, and D. Xiu, "Empirical asset pricing via machine learning," in *Proc. 31st Australas. Finance Banking Conf.*, Chicago Booth Res. Paper 18-04, Yale ICF Working Paper 2018-09, Sep. 2019. [Online]. Available: https://ssrn.com/abstract=3159577

[15] P. Mathur, "Overview of machine learning in finance," in *Machine Learning Applications Using Python*. Berkeley, CA, USA: Apress, 2019, pp. 259–270. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4842-3787-8_13

[16] S. Emerson, R. Kennedy, L. O'Shea, and J. O'Brien, "Trends and applications of machine learning in quantitative finance," in *Proc. 8th Int. Conf. Econ. Finance Res. (ICEFR)*, 2019, pp. 1–9.

[17] K. Shaukat *et al.*, "Student's performance in the context of data mining," in *Proc. 19th Int. Multi-Topic Conf. (INMIC)*, 2016.

[18] S. Jha and E. J. Topol, "Adapting to artificial intelligence: Radiologists and pathologists as information specialists," *Jama*, vol. 316, no. 22, pp. 2353–2354, 2016.

[19] A. I. Tekkesin, "Artificial intelligence in healthcare: Past, present and future," *Anatolian J. Cardiol.*, vol. 2, no. 4, pp. 230–243, 2019.

[20] K. Shaukat, N. Masood, A. Bin Shafaat, K. Jabbar, H. Shabbir, and S. Shabbir, "Dengue fever in perspective of clustering algorithms," 2015, *arXiv:1511.07353*. [Online]. Available: http://arxiv.org/abs/1511.07353

[21] K. S. Dar and S. M. U. Azmeen, "Dengue fever prediction: A data mining problem," *J. Data Mining Genomics Proteomics*, vol. 6, no. 3, pp. 1–5, 2015.

[22] B.-H. Li, B.-C. Hou, W.-T. Yu, X.-B. Lu, and C.-W. Yang, "Applications of artificial intelligence in intelligent manufacturing: A review," *Frontiers Inf. Technol. Electron. Eng.*, vol. 18, no. 1, pp. 86–96, 2017.

[23] C. Virmani, T. Choudhary, A. Pillai, and M. Rani, "Applications of machine learning in cyber security," in *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*. Hershey, PA, USA: IGI Global, 2020, pp. 83–103.

[24] R. Calderon, "The benefits of artificial intelligence in cybersecurity," La Salle Univ., Philadelphia, PA, USA, Tech. Rep. Winter 1-15-2019, 2019.

[25] M. Taddeo, "Three ethical challenges of applications of artificial intelligence in cybersecurity," *Minds Mach.*, vol. 29, no. 2, pp. 1–5, 2019.

[26] K. Shaukat, A. Rubab, I. Shehzadi, and R. Iqbal, "A socio-technological analysis of cyber crime and cyber security in Pakistan," *Transylvanian Rev.*, vol. 1, no. 3, pp. 4187–4190, 2017.

[27] R. Sagar, R. Jhaveri, and C. Borrego, "Applications in security and evasions in machine learning: A survey," *Electronics*, vol. 9, no. 1, p. 97, Jan. 2020.

[28] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, S. Chen, D. Liu, and J. Li, "Performance comparison and current challenges of using machine learning techniques in cybersecurity," *Energies*, vol. 13, no. 10, p. 2509, May 2020.

[29] S. Selvaraj, "Applying of machine learning for spam classification," M.S. thesis, Instytut Telekomunikacji, 2019. [Online]. Available: http://repo.bg.pw.edu.pl/index.php/en/r#/info/master/WUT4e00628d97f64f24b0935f599fbdf4b6/?affil=IT&r=diploma&tab=&lang=en

[30] A. A. Alurkar, S. B. Ranade, S. V. Joshi, S. S. Ranade, G. R. Shinde, P. A. Sonewar, and P. N. Mahalle, "A comparative analysis and discussion of email spam classification methods using machine learning techniques," in *Applied Machine Learning for Smart Data Analysis*. Boca Raton, FL, USA: CRC Press, May 2019, p. 225. [Online]. Available: https://books.google.com.au/books?id=iKWaDwAAQBAJ&dq=A+comparative+analysis+and+discussion+of+email+spam+classi%3Fcation+methods+using+machine+learning+tech-+2083+niques,&lr=&source=gbs_navlinks_s

[31] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: Review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, Jun. 2019, Art. no. e01802.

[32] A. K. Jain, D. Goel, S. Agarwal, Y. Singh, and G. Bajaj, "Predicting spam messages using back propagation neural network," *Wireless Pers. Commun.*, vol. 110, no. 1, pp. 403–422, Jan. 2020.

[33] D. Prusti, S. S. Padmanabhuni, and S. K. Rath, "Credit card fraud detection by implementing machine learning techniques," Tech. Rep., 2019, pp. 1–10. [Online]. Available: http://dspace.nitrkl.ac.in/dspace/handle/2080/3273

[34] H. A. Shukur and S. Kurnaz, "Credit card fraud detection using machine learning methodology," *Int. J. Comput. Sci. Mobile Comput.*, vol. 8, no. 3, pp. 257–260, Mar. 2019.

[35] M. Lokanan, V. Tran, and N. H. Vuong, "Detecting anomalies in financial statements using machine learning algorithm," *Asian J. Accounting Res.*, vol. 4, no. 2, pp. 181–201, Oct. 2019.

[36] A. M. Mubalaike and E. Adali, "Deep learning approach for intelligent financial fraud detection system," in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2018, pp. 598–603.

[37] Z. Ma, H. Ge, Y. Liu, M. Zhao, and J. Ma, "A combination method for Android malware detection based on control flow graphs and machine learning algorithms," *IEEE Access*, vol. 7, pp. 21235–21245, 2019.

[38] S. Saad, W. Briguglio, and H. Elmiligi, "The curious case of machine learning in malware detection," 2019, *arXiv:1905.07573*. [Online]. Available: http://arxiv.org/abs/1905.07573

[39] P. Jain, "Machine learning versus deep learning for malware detection," Master's Projects, 2019, p. 704. [Online]. Available: https://scholarworks.sjsu.edu/etd_projects/704/, doi: 10.31979/etd.56y7-b74e.

[40] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," 2017, *arXiv:1701.07179*. [Online]. Available: http://arxiv.org/abs/1701.07179

[41] R. S. Rao and A. R. Pais, "Detection of phishing Websites using an efficient feature-based machine learning framework," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3851–3873, Aug. 2019.

[42] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Syst. Appl.*, vol. 117, pp. 345–357, Mar. 2019.

[43] M. Alauthman, A. Almomani, M. Alweshah, W. Omoushd, and K. Alieyane, "Machine learning for phishing detection and mitigation," in *Machine Learning for Computer and Cyber Security: Principle, Algorithms, and Practices*. New York, NY, USA: CRC Press, 2019, p. 26.

[44] M. Almukaynizi, A. Grimm, E. Nunes, J. Shakarian, and P. Shakarian, "Predicting cyber threats through the dynamics of user connectivity in darkweb and deepweb forums," *ACM Comput. Social Sci.*, 2017. [Online]. Available: https://usc-isi-i2.github.io/papers/kristina02.pdf

[45] M. Almukaynizi, A. Grimm, E. Nunes, J. Shakarian, and P. Shakarian, "Predicting cyber threats through hacker social networks in darkweb and deepweb forums," in *Proc. Int. Conf. Comput. Social Sci. Soc. Americas (CSS)*, 2017, pp. 1–7.

[46] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Comput. Netw.*, vol. 151, pp. 147–157, Mar. 2019.

[47] S. Dey, Q. Ye, and S. Sampalli, "A machine learning based intrusion detection scheme for data fusion in mobile clouds involving heterogeneous client networks," *Inf. Fusion*, vol. 49, pp. 205–215, Sep. 2019.

[48] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419.

[49] M. Pradhan, C. K. Nayak, and S. K. Pradhan, "Intrusion detection system (IDS) and their types," in *Securing the Internet of Things: Concepts, Methodologies, Tools, and Applications*. Hershey, PA, USA: IGI Global, 2020, pp. 481–497.

[50] S. Sarkar, M. Almukaynizi, J. Shakarian, and P. Shakarian, "Predicting enterprise cyber incidents using social network analysis on dark Web hacker forums," *Cyber Defense Rev.*, pp. 87–102, Mar. 2019.

[51] A. Zenebe, M. Shumba, A. Carillo, and S. Cuenca, "Cyber threat discovery from dark Web," in *Proc. 28th Int. Conf.*, vol. 64, 2019, pp. 174–183.

[52] E. Nunes, A. Diab, A. Gunn, E. Marin, V. Mishra, V. Paliath, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian, "Darknet and deepnet mining for proactive cybersecurity threat intelligence," in *Proc. IEEE Conf. Intell. Secur. Informat. (ISI)*, Sep. 2016, pp. 7–12.

[53] M. Kadoguchi, S. Hayashi, M. Hashimoto, and A. Otsuka, "Exploring the dark Web for cyber threat intelligence using machine leaning," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Jul. 2019, pp. 200–202.

[54] X. Zhang and K. Chow, "A framework for dark Web threat intelligence analysis," in *Cyber Warfare and Terrorism: Concepts, Methodologies, Tools, and Applications*. Hershey, PA, USA: IGI Global, 2020, pp. 266–276.

[55] R. Prasad and V. Rohokale, "Artificial intelligence and machine learning in cyber security," in *Cyber Security: The Lifeline of Information and Communication Technology*. Cham, Switzerland: Springer, 2020, pp. 231–247.

[56] *Security Information and Event Management (SIEM)*. Accessed: May 27, 2020. [Online]. Available: https://www.esecurityplanet.com/products/top-siem-products.html

[57] *Top Intrusion Detection and Prevention Systems: Guide to IDPS*. Accessed: May 27, 2020. [Online]. Available: https://www.esecurityplanet.com/products/top-intrusion-detection-prevention-systems.html

[58] *Unified Threat Management*. Accessed: May 27, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Unified_threat_management

[59] B. Geluvaraj, P. Satwik, and T. A. Kumar, "The future of cybersecurity: Major role of artificial intelligence, machine learning, and deep learning in cyberspace," in *Proc. Int. Conf. Comput. Netw. Commun. Technol.* Singapore: Springer, 2019, pp. 739–747.

[60] *How Artificial Intelligence is Transforming Cybersecurity*. Accessed: May 28, 2020. [Online]. Available: https://www.plugandplaytechcenter.com/resources/how-artificial-intelligence-transforming-cybersecurity/

[61] A. Sharma, Z. Kalbarczyk, J. Barlow, and R. Iyer, "Analysis of security data from a large computing organization," in *Proc. IEEE/IFIP 41st Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2011, pp. 506–517.

[62] B. Arslan, S. Gunduz, and S. Sagiroglu, "A review on mobile threats and machine learning based detection approaches," in *Proc. 4th Int. Symp. Digit. Forensic Secur. (ISDFS)*, Apr. 2016, pp. 7–13.

[63] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," 2019, *arXiv:1906.05799*. [Online]. Available: http://arxiv.org/abs/1906.05799

[64] K. Geis, "Machine learning: Cybersecurity that can meet the demands of today as well as the demands of tomorrow," Ph.D. dissertation, Master Sci. Cybersecur., Utica College, Utica, NY, USA, 2019.

[65] M. Thangavel, A. S. TGR, P. Priyadharshini, and T. Saranya, "Review on machine and deep learning applications for cyber security," in *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*. Hershey, PA, USA: IGI Global, 2020, pp. 42–63.

[66] R. G. M. Helali, "Data mining based network intrusion detection system: A survey," in *Novel Algorithms and Techniques in Telecommunications and Networking*. Dordrecht, The Netherlands: Springer, 2010, pp. 501–505. [Online]. Available: https://link.springer.com/chapter/10.1007/978-90-481-3662-9_86

[67] D. P. Vinchurkar and A. Reshamwala, "A review of intrusion detection system using neural network and machine learning," *Int. J. Eng. Sci. Innov. Technol.*, vol. 1, no. 2, pp. 54–63, 2012. [Online]. Available: https://www.researchgate.net/profile/Alpa_Reshamwala/publication/233943805_A_Review_of_Intrusion_Detection_System_Using_Neural_Network_and_Machine_Learning_Technique/links/02bfe50d2f409cda-fa000000.pdf

[68] J. Singh and M. J. Nene, "A survey on machine learning techniques for intrusion detection systems," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 2, no. 11, pp. 4349–4355, 2013.

[69] M. Z. Mas'ud, S. Sahib, M. F. Abdollah, S. R. Selamat, and R. Yusof, "Analysis of features selection and machine learning classifier in Android malware detection," in *Proc. Int. Conf. Inf. Sci. Appl. (ICISA)*, May 2014, pp. 1–5.

[70] E. Gandotra, D. Bansal, and S. Sofat, "Malware analysis and classification: A survey," *J. Inf. Secur.*, vol. 5, no. 2, p. 56, 2014.

[71] B. Dharamkar and R. R. Singh, "A review of cyber attack classification technique based on data mining and neural network approach," *Int. J. Comput. Trends Technol.*, vol. 7, no. 2, pp. 100–105, Jan. 2014.

[72] V. Ford and A. Siraj, "Applications of machine learning in cyber security," in *Proc. 27th Int. Conf. Comput. Appl. Ind. Eng.*, 2014, pp. 1–6.

[73] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[74] H. Jiang, J. Nagra, and P. Ahammad, "SoK: Applying machine learning in security—A survey," 2016, *arXiv:1611.03186*. [Online]. Available: http://arxiv.org/abs/1611.03186

[75] L. S. Wijesinghe, L. N. B. De Silva, G. T. A. Abhayaratne, P. Krithika, S. M. D. R. Priyashan, and D. Dhammearatchi, "Combating cyber crime using artificial agent systems," *Int. J. Sci. Res. Publications*, vol. 6, no. 4, pp. 265–271, Apr. 2016. [Online]. Available: http://www.ijsrp.org/research-paper-0416/ijsrp-p5241.pdf.

[76] R. Das and T. H. Morris, "Machine learning and cyber security," in *Proc. Int. Conf. Comput., Electr. Commun. Eng. (ICCECE)*, 2017, pp. 1–7.

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

[78] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[79] J.-H. Li, "Cyber security meets artificial intelligence: A survey," *Frontiers Inf. Technol. Electron. Eng.*, vol. 19, no. 12, pp. 1462–1474, Dec. 2018.

[80] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," 2017, *arXiv:1710.08189*. [Online]. Available: http://arxiv.org/abs/1710.08189

[81] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, ''On the effectiveness of machine and deep learning for cyber security,'' in *Proc. 10th Int. Conf. Cyber Conflict (CyCon)*, May 2018, pp. 371–390.
[82] A. P. Veiga, ''Applications of artificial intelligence to network security,'' 2018, *arXiv:1803.09992*. [Online]. Available: http://arxiv.org/abs/1803.09992
[83] Z. Guan, L. Bian, T. Shang, and J. Liu, ''When machine learning meets security issues: A survey,'' in *Proc. IEEE Int. Conf. Intell. Saf. Robot. (ISR)*, Aug. 2018, pp. 158–165.
[84] A. Shalaginov, S. Banin, A. Dehghantanha, and K. Franke, ''Machine learning aided static malware analysis: A survey and tutorial,'' in *Cyber Threat Intelligence*. Cham, Switzerland: Springer, 2018, pp. 7–45.
[85] B. S. Sagar, S. Niranjan, N. Kashyap, and D. N. Sachin, ''Providing cyber security using artificial intelligence—A survey,'' in *Proc. 3rd Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Mar. 2019, pp. 717–720.
[86] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, ''A survey of deep learning-based network anomaly detection,'' *Cluster Comput.*, vol. 22, pp. 1–13, Jan. 2017.
[87] D. Berman, A. Buczak, J. Chavis, and C. Corbett, ''A survey of deep learning methods for cyber security,'' *Information*, vol. 10, no. 4, p. 122, Apr. 2019.
[88] J. M. Torres, C. I. Comesaña, and P. J. García-Nieto, ''Machine learning techniques applied to cybersecurity,'' *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 10, pp. 2823–2836, 2019.
[89] A. Handa, A. Sharma, and S. K. Shukla, ''Machine learning in cybersecurity: A review,'' *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 9, no. 4, p. e1306, 2019.
[90] D. Gumusbas, T. Yldrm, A. Genovese, and F. Scotti, ''A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems,'' *IEEE Syst. J.*, early access, May 26, 2020, doi: 10.1109/JSYST.2020.2992966.
[91] B. M. Coronado, U. Mori, A. Mendiburu, and J. Miguel-Alonso, ''Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process,'' *IEEE Trans. Netw. Service Manage.*, early access, Aug. 12, 2020, doi: 10.1109/TNSM.2020.3016246.
[92] *Difference Between Threat and Attack*. Accessed: Jun. 3, 2020. [Online]. Available: https://www.geeksforgeeks.org/difference-between-threat-and-attack/
[93] *What is Cyber-Security?* Accessed: Jan. 11, 2020. [Online]. Available: https://www.kaspersky.com.au/resource-center/definitions/what-is-cyber-security
[94] J. A. Lewis, ''National perceptions of cyber threats,'' *Strategic Anal.*, vol. 38, no. 4, pp. 566–576, Jul. 2014.
[95] K.-F. Cheung and M. G. H. Bell, ''Attacker–defender model against quantal response adversaries for cyber security in logistics management: An introductory study,'' *Eur. J. Oper. Res.*, pp. 1–11, Oct. 2019, doi: 10.1016/j.ejor.2019.10.019.
[96] (2018). *Cisco 2018 Annual Cybersecurity Report*. Accessed: Dec. 25, 2019. [Online]. Available: https://www.cisco.com/c/m/en_au/products/security/offers/annual-cybersecurity-report-2018.html
[97] E. A. Fischer, ''Creating a national framework for cybersecurity: An analysis of issues and options,'' Library Congr., Washington, DC, USA, Tech. Rep. RL32777, 2005.
[98] S. Purkait, ''Phishing counter measures and their effectiveness–literature review,'' *Inf. Manage. Comput. Secur.*, vol. 20, no. 5, pp. 382–420, Nov. 2012.
[99] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, ''Social phishing,'' *Commun. ACM*, vol. 50, no. 10, pp. 94–100, 2007.
[100] R. M. Mohammad, F. Thabtah, and L. McCluskey, ''Tutorial and critical analysis of phishing Websites methods,'' *Comput. Sci. Rev.*, vol. 17, pp. 1–24, Aug. 2015.
[101] E. H. Spafford, ''Computer viruses as artificial life,'' *Artif. Life*, vol. 1, no. 3, pp. 249–265, Apr. 1994.
[102] G. B. Shelly and M. E. Vermaat, *Discovering Computers-Fundamentals 2011 Edition*. 2010. [Online]. Available: https://dl.acm.org/doi/book/10.5555/1841059
[103] G. B. Shelly, T. J. Cashman, and M. E. Vermaat, *Discovering Computers*. Jakarta, Indonesia: Salemba Infotek, 2012.
[104] H. Drucker, D. Wu, and V. N. Vapnik, ''Support vector machines for spam categorization,'' *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, Sep. 1999.
[105] N. Jindal and B. Liu, ''Review spam detection,'' in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 1189–1190.
[106] S. M. Abdulhamid, M. S. Abd Latiff, H. Chiroma, O. Osho, G. Abdul-Salaam, A. I. Abubakar, and T. Herawan, ''A review on mobile SMS spam filtering techniques,'' *IEEE Access*, vol. 5, pp. 15650–15666, 2017.
[107] D. D. Arifin and M. A. Bijaksana, ''Enhancing spam detection on mobile phone short message service (SMS) performance using FP-growth and naive Bayes classifier,'' in *Proc. IEEE Asia Pacific Conf. Wireless Mobile (APWiMob)*, Sep. 2016, pp. 80–84.
[108] J. Raiyn, ''A survey of cyber attack detection strategies,'' *Int. J. Secur. Appl.*, vol. 8, no. 1, pp. 247–256, Jan. 2014.
[109] P. Ganapathi, ''A review of machine learning methods applied for handling zero-day attacks in the cloud environment,'' in *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*. Hershey, PA, USA: IGI Global, 2020, pp. 364–387.
[110] K. Kotapati, P. Liu, Y. Sun, and T. F. LaPorta, ''A taxonomy of cyber attacks on 3G networks,'' in *Proc. Int. Conf. Intell. Secur. Inform.* Berlin, Germany: Springer, 2005, pp. 631–633.
[111] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta, and Q. Wu, ''AVOIDIT: A cyber attack taxonomy,'' in *Proc. 9th Annu. Symp. Inf. Assurance (ASIA)*, 2014, pp. 2–12.
[112] B. Narwal, A. K. Mohapatra, and K. A. Usmani, ''Towards a taxonomy of cyber threats against target applications,'' *J. Statist. Manage. Syst.*, vol. 22, no. 2, pp. 301–325, Feb. 2019.
[113] P. Radmand, A. Talevski, S. Petersen, and S. Carlsen, ''Taxonomy of wireless sensor network cyber security attacks in the oil and gas industries,'' in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 949–957.
[114] A. Sari and U. C. Atasoy, ''Taxonomy of cyber attack weapons, defense strategies, and cyber war incidents,'' in *Applying Methods of Scientific Inquiry Into Intelligence, Security, and Counterterrorism*. Hershey, PA, USA: IGI Global, 2019, pp. 1–45.
[115] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, and G. Loukas, ''A taxonomy and survey of attacks against machine learning,'' *Comput. Sci. Rev.*, vol. 34, Nov. 2019, Art. no. 100199.
[116] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, ''DDoS attacks in cloud computing: Issues, taxonomy, and future directions,'' *Comput. Commun.*, vol. 107, pp. 30–48, Jul. 2017.
[117] C. Haruna, M. Abdulhamid, Y. Abdulsalam, M. Ali, and U. Timothy, ''Academic community cyber cafes—A Perpetration point for cyber crimes in Nigeria,'' *Int. J. Inf. Sci. Comput. Eng.*, vol. 2, no. 2, pp. 7–13, 2011.
[118] S. I. M. Abdulhamid, C. Haruna, and A. Abubakar, ''Cybercrimes and the nigerian academic institution networks,'' *IUP J. Inf. Technol.*, vol. 7, no. 1, pp. 47–57, Mar. 2011.
[119] A. Özgür and H. Erdem, ''A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015,'' *PeerJ Preprints*, vol. 4, Apr. 2016, Art. no. e1954v1.
[120] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, ''A detailed analysis of the KDD CUP 99 data set,'' in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.
[121] H. Saxena and V. Richariya, ''Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain,'' *Int. J. Comput. Appl.*, vol. 98, no. 6, pp. 25–29, Jul. 2014.
[122] S. Revathi and A. Malathi, ''A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection,'' *Int. J. Eng. Res. Technol.*, vol. 2, no. 12, pp. 1848–1853, 2013.
[123] B. Ingre and A. Yadav, ''Performance analysis of NSL-KDD dataset using ANN,'' in *Proc. Int. Conf. Signal Process. Commun. Eng. Syst.*, Jan. 2015, pp. 92–96.
[124] M. Xie, J. Hu, and J. Slay, ''Evaluating host-based anomaly detection systems: Application of the one-class SVM algorithm to ADFA-LD,'' in *Proc. 11th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Xiamen, China, 2014. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6980972
[125] M. Xie, J. Hu, and J. Slay, ''Evaluating host-based anomaly detection systems: Application of the one-class SVM algorithm to ADFA-LD,'' in *Proc. 11th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Aug. 2014, pp. 978–982.
[126] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, ''An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection,'' *Pattern Recognit. Lett.*, vol. 51, pp. 1–7, Jan. 2015.

[127] D. Bhamare, T. Salman, M. Samaka, A. Erbad, and R. Jain, "Feasibility of supervised machine learning for cloud security," in *Proc. Int. Conf. Inf. Sci. Secur. (ICISS)*, Dec. 2016, pp. 1–5.

[128] P. Nevavuori and T. Kokkonen, "Requirements for training and evaluation dataset of network and host intrusion detection system," in *Proc. World Conf. Inf. Syst. Technol.* Cham, Switzerland: Springer, 2019, pp. 534–546.

[129] A. Aldribi, I. Traore, and B. Moa, "Data sources and datasets for cloud intrusion detection modeling and evaluation," in *Cloud Computing for Optimization: Foundations, Applications, and Challenges*. Cham, Switzerland: Springer, 2018, pp. 333–366.

[130] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation," in *Proc. DARPA Inf. Survivability Conf. Expo. (DISCEX)*, 2000, pp. 12–26.

[131] A. Chahal and R. Nagpal, "Performance of snort on Darpa dataset and diferent false alert reduction techniques," in *Proc. 3rd Int. Conf. Elect., Electron., Eng. Trends, Commun., Optim. Sci. (EEECOS)*, 2016, pp. 1–8. [Online]. Available: https://pdfs. semanticscholar.org/9634/2f678949bcae35eabda3cfafeb0d0abe1d32.pdf

[132] C. Brown, A. Cowperthwaite, A. Hijazi, and A. Somayaji, "Analysis of the 1999 DARPA/Lincoln laboratory IDS evaluation data with NetADHICT," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–7.

[133] K. Kato and V. Klyuev, "An intelligent ddos attack detection system using packet analysis and support vector machine," *IJICR*, vol. 14, no. 5, pp. 478–485, 2014.

[134] M. Małowidzki, P. Berezinski, and M. Mazur, "Network intrusion detection: Half a kingdom for a good dataset," in *Proc. NATO STO SAS Workshop*, Lisbon, Portugal, 2015, pp. 1–6.

[135] S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H. Medal, M. Marufuzzaman, and L. Bian, "Botnet detection using graph-based feature clustering," *J. Big Data*, vol. 4, no. 1, p. 14, Dec. 2017.

[136] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, Sep. 2014.

[137] C. T. Giménez, A. P. Villegas, and G. Marañón, "HTTP data set CSIC 2010," Inf. Secur. Inst. CSIC, Spanish Res. Nat. Council, New Delhi, India, 2010. [Online]. Available: https://www.isi.csic.es/dataset/

[138] D. Atienza, Á. Herrero, and E. Corchado, "Neural analysis of http traffic for Web attack detection," in *Proc. Comput. Intell. Secur. Inf. Syst. Conf.* Cham, Switzerland: Springer, 2015, pp. 201–212.

[139] M. A. M. Hasan, M. Nasser, S. Ahmad, and K. I. Molla, "Feature selection for intrusion detection using random forest," *J. Inf. Secur.*, vol. 7, no. 3, p. 129, 2016.

[140] B. Gallagher and T. Eliassi-Rad, "Classification of http attacks: A study on the ECML/PKDD 2007 discovery challenge," Lawrence Livermore Nat. Lab., Livermore, CA, USA, Tech. Rep. LLNL-TR-414570, 2009.

[141] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[142] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J. A, Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Apr. 2016.

[143] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," in *Proc. IEEE 26th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2017, pp. 1881–1886.

[144] D. G. Mogal, S. R. Ghungrad, and B. B. Bhusare, "NIDS using machine learning classifiers on UNSW-NB15 and KDDCUP99 datasets," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 6, no. 4, pp. 533–537, Apr. 2017.

[145] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Multi-classification of UNSW-NB15 dataset for network anomaly detection system," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 15, pp. 5094–5104, Aug. 2018.

[146] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems," *Int. J. Eng. Technol.*, vol. 7, no. 3.24, pp. 479–482, 2018.

[147] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving AdaBoost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset," *J. Phys., Conf. Ser.*, vol. 1192, Mar. 2019, Art. no. 012018.

[148] D. Stiawan, M. Y. Bin Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020.

[149] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[150] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for Internet of Things in smart city," *Future Gener. Comput. Syst.*, vol. 107, pp. 433–442, Jun. 2020.

[151] Y. Rizk, N. Hajj, N. Mitri, and M. Awad, "Deep belief networks and cortical algorithms: A comparative study for supervised classification," *Appl. Comput. Informat.*, vol. 15, no. 2, pp. 81–93, Jul. 2019.

[152] R. Karthika and P. Visalakshi, "A hybrid ACO based feature selection method for email spam classification," *WSEAS Trans. Comput*, vol. 14, no. 2015, pp. 171–177, Jan. 2015.

[153] W. A. Awad, "Machine learning methods for spam E-Mail classification," *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 1, pp. 173–184, Feb. 2011.

[154] M. N. Asim, M. Wasim, M. S. Ali, and A. Rehman, "Comparison of feature selection methods in text classification on highly skewed datasets," in *Proc. 1st Int. Conf. Latest Trends Electr. Eng. Comput. Technol. (INTELLECT)*, Nov. 2017, pp. 1–8.

[155] P. Azunre, C. Corcoran, N. Dhamani, J. Gleason, G. Honke, D. Sullivan, R. Ruppel, S. Verma, and J. Morgan, "Semantic classification of tabular datasets via character-level convolutional neural networks," 2019, *arXiv:1901.08456*. [Online]. Available: http://arxiv.org/abs/1901.08456

[156] S. K. Trivedi and S. Dey, "A combining classifiers approach for detecting email spams," in *Proc. 30th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Mar. 2016, pp. 355–360.

[157] T. Zaki, M. S. Uddin, M. M. Hasan, and M. N. Islam, "Security threats for big data: A study on enron e-mail dataset," in *Proc. Int. Conf. Res. Innov. Inf. Syst. (ICRIIS)*, Jul. 2017, pp. 1–6.

[158] *SMS Spam Collection Dataset*. Accessed: Aug. 10, 2020. [Online]. Available: https://www.kaggle.com/uciml/sms-spam-collection-dataset

[159] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: New collection and results," in *Proc. 11th ACM Symp. Document Eng. (DocEng)*, 2011, pp. 259–262.

[160] *Email Spam Dataset*. Accessed: Aug. 10, 2020. [Online]. Available: https://www.kaggle.com/veleon/ham-and-spam-dataset

[161] S. O. Olatunji, "Improved email spam detection model based on support vector machines," *Neural Comput. Appl.*, vol. 31, no. 3, pp. 691–699, Mar. 2019.

[162] S. R. Ayala. (2017). *An Automated Behaviour-Based Malware Analysis Method Based on Free Open Source Software*. [Online]. Available: http://openaccess.uoc.edu/webapps/o2/handle/10609/66365

[163] H. Kim, T. Cho, G.-J. Ahn, and J. H. Yi, "Risk assessment of mobile applications based on machine learned malware dataset," *Multimedia Tools Appl.*, vol. 77, no. 4, pp. 5027–5042, Feb. 2018.

[164] H.-J. Zhu, Z.-H. You, Z.-X. Zhu, W.-L. Shi, X. Chen, and L. Cheng, "DroidDet: Effective and robust detection of Android malware using static analysis along with rotation forest model," *Neurocomputing*, vol. 272, pp. 638–646, Jan. 2018.

[165] M. S. I. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, "Detecting malicious urls using lexical analysis," in *Proc. Int. Conf. Netw. Syst. Secur.* Cham, Switzerland: Springer, 2016, pp. 467–482.

[166] J. Saxe and K. Berlin, "EXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys," 2017, *arXiv:1702.08568*. [Online]. Available: http://arxiv.org/abs/1702.08568

[167] A. H. Lashkari, A. F. A. Kadir, L. Taheri, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark Android malware datasets and classification," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2018, pp. 1–7.

[168] F. Noorbehbahani, F. Rasouli, and M. Saberi, "Analysis of machine learning techniques for ransomware detection," in *Proc. 16th Int. ISC (Iranian Soc. Cryptology), Conf. Inf. Secur. Cryptol. (ISCISC)*, Aug. 2019, pp. 128–133.

[169] E. C. Bayazit, O. Koray Sahingoz, and B. Dogan, "Malware detection in Android systems with traditional machine learning models: A survey," in *Proc. Int. Congr. Hum.-Comput. Interact., Optim. Robotic Appl. (HORA)*, Jun. 2020, pp. 1–8.

[170] X. Wang, S. Zhu, D. Zhou, and Y. Yang, "Droid-AntiRM: Taming control flow anti-analysis to support automated dynamic analysis of Android malware," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, Dec. 2017, pp. 350–361.

[171] N. Kiss, J.-F. Lalande, M. Leslous, and V. V. T. Tong, "Kharon dataset: Android malware under a microscope," in *Proc. LASER Workshop, Learn. Authoritative Secur. Exp. Results (LASER)*, 2016, pp. 1–12.
[172] H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, "Droidkin: Lightweight detection of Android apps similarity," in *Proc. Int. Conf. Secur. Privacy Commun. Netw.* Cham, Switzerland: Springer, 2014, pp. 436–453.
[173] S. Y. Yerima and M. K. Alzaylaee, "Mobile botnet detection: A deep learning approach using convolutional neural networks," in *Proc. Int. Conf. Cyber Situational Awareness, Data Analytics Assessment (CyberSA)*, Jun. 2020, pp. 1–8.
[174] C. Torrano-Gimenez, A. Pérez-Villegas, G. Álvarez, E. Fernández-Medina, M. Malek, and J. Hernando, "An anomaly-based Web application firewall," in *Proc. SECRYPT*, 2009, pp. 23–28.
[175] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.
[176] *The BoT-IoT Dataset*. Accessed: Aug. 8, 2020. [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php
[177] Spambase Dataset. *Center for Machine Learning and Intelligent Systems at UC Irvine*. Assessed: Oct. 29, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Spambase
[178] B. Klimt and Y. Yang, "Introducing the Enron corpus," in *Proc. CEAS*, 2004, pp. 1–2.
[179] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos, "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages," in *Proc. 23rd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2000, pp. 160–167.
[180] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos, "A memory-based approach to anti-spam filtering for mailing lists," *Inf. Retr.*, vol. 6, no. 1, pp. 49–73, 2003.
[181] C. Tagg, "A corpus linguistics study of SMS text messaging," Ph.D. dissertation, Dept. English Literature, Univ. Birmingham, Birmingham, U.K., 2009. [Online]. Available: http://etheses.bham.ac.uk/253/1/Tagg09PhD.pdf
[182] *VirusShare*. Accessed: Nov. 15, 2020. [Online]. Available: https://virusshare.com/
[183] *Android Malware Dataset (CICAndMal2017)*. Accessed: Aug. 8, 2020. [Online]. Available: https://www.unb.ca/cic/datasets/andmal2017.html
[184] *Kharon Malware Dataset*. Accessed: Aug. 8, 2020. [Online]. Available: http://kharon.gforge.inria.fr/dataset/
[185] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," *IEEE Access*, vol. 8, pp. 104650–104675, 2020.
[186] *Android Adware and General Malware Dataset*. Accessed: Aug. 8, 2020. [Online]. Available: https://www.unb.ca/cic/datasets/android-adware.html
[187] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Inf. Sci.*, vols. 340–341, pp. 250–261, May 2016.
[188] D. Michie, D. J. Spiegelhalter, and C. Taylor, "Machine learning," *Neural Stat. Classification*, vol. 13, 1994.
[189] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. New York, NY, USA: Auerbach, 2016.
[190] S. Angra and S. Ahuja, "Machine learning and its applications: A review," in *Proc. Int. Conf. Big Data Anal. Comput. Intell. (ICBDAC)*, 2017, pp. 57–60.
[191] T. M. Alam, K. Shaukat, M. Mushtaq, Y. Ali, M. Khushi, S. Luo, and A. Wahab, "Corporate bankruptcy prediction: An approach towards better corporate world," *Comput. J.*, pp. 1–16, Jun. 2020.
[192] J. B. Fraley and J. Cannady, "The promise of machine learning in cybersecurity," in *Proc. SoutheastCon*, Mar. 2017, pp. 1–6.
[193] A. Kulkarni and L. L. Brown, III, "Phishing websites detection using machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 7, pp. 8–13, 2019, doi: 10.14569/IJACSA.2019.0100702.
[194] M. Islam and N. K. Chowdhury, "Phishing Websites detection using machine learning based classification techniques," in *Proc. 1st Int. Conf. Adv. Inf. Commun. Technol.*, 2016, pp. 1–4.
[195] S. Marsland, *Machine Learning: An Algorithmic Perspective*. Boca Raton, FL, USA: CRC Press, 2014.
[196] S. R. Granter, A. H. Beck, and D. J. Papke, "AlphaGo, deep learning, and the future of the human microscopist," *Arch. Pathol. Lab. Med.*, vol. 141, no. 5, pp. 619–621, May 2017.
[197] J. X. Chen, "The evolution of computing: AlphaGo," *Comput. Sci. Eng.*, vol. 18, no. 4, pp. 4–7, Jul. 2016.
[198] M. H. Ling, K.-L.-A. Yau, J. Qadir, G. S. Poh, and Q. Ni, "Application of reinforcement learning for security enhancement in cognitive radio networks," *Appl. Soft Comput.*, vol. 37, pp. 809–829, Dec. 2015.
[199] Y. Wang, Z. Ye, P. Wan, and J. Zhao, "A survey of dynamic spectrum allocation based on reinforcement learning algorithms in cognitive radio networks," *Artif. Intell. Rev.*, vol. 51, no. 3, pp. 493–506, Mar. 2019.
[200] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [Exploratory DSP]," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 145–154, Jan. 2011.
[201] Y. Bengio, "Learning deep architectures for AI," in *Foundations Trends Machine Learning*, vol. 2, no. 1. Boston, MA, USA: Now, 2009.
[202] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12 pp. 2493–2537, Aug. 2011.
[203] P. Le Callet, C. Viard-Gaudin, and D. Barba, "A convolutional neural network approach for objective video quality assessment," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1316–1327, Sep. 2006.
[204] L. F. Maimo, A. L. P. Gomez, F. J. G. Clemente, M. G. Perez, and G. M. Perez, "A self-adaptive deep learning-based system for anomaly detection in 5G networks," *IEEE Access*, vol. 6, pp. 7700–7712, 2018.
[205] A. Abeshu and N. Chilamkurti, "Deep learning: The frontier for distributed attack detection in fog-to-things computing," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 169–175, Feb. 2018.
[206] T. M. Kebede, O. Djaneye-Boundjou, B. N. Narayanan, A. Ralescu, and D. Kapp, "Classification of malware programs using autoencoders based deep learning architecture and its application to the microsoft malware classification challenge (BIG 2015) dataset," in *Proc. IEEE Nat. Aerosp. Electron. Conf. (NAECON)*, Jun. 2017, pp. 70–75.
[207] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," *J. Biomed. Informat.*, vol. 83, pp. 112–134, Jul. 2018.
[208] S. Feng, H. Zhou, and H. Dong, "Using deep neural network with small dataset to predict material defects," *Mater. Des.*, vol. 162, pp. 300–310, Jan. 2019.
[209] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2617–2634, Oct. 2005.
[210] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 582–588.
[211] A. L. Prodromidis and S. J. Stolfo, "Cost complexity-based pruning of ensemble classifiers," *Knowl. Inf. Syst.*, vol. 3, no. 4, pp. 449–469, Nov. 2001.
[212] J. R. Quinlan, *C4. 5: Programs for Machine Learning*. Amsterdam, The Netherlands: Elsevier, 2014.
[213] V. H. Garcia, R. Monroy, and M. Quintana, "Web attack detection using ID3," in *Proc. IFIP World Comput. Congr. (TC)*, vol. 12. Boston, MA, USA: Springer, 2006, pp. 323–332.
[214] *WEKA Packages*. Accessed: Nov. 16, 2020. [Online]. Available: https://weka.sourceforge.io/packageMetaData/
[215] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.
[216] A. A. Aburomman and M. B. Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Appl. Soft Comput.*, vol. 38, pp. 360–372, Jan. 2016.
[217] S. He, G. M. Lee, S. Han, and A. B. Whinston, "How would information disclosure influence organizations' outbound spam volume? Evidence from a field experiment," *J. Cybersecurity*, vol. 2, no. 1, pp. 99–118, Dec. 2016.
[218] S. T. Miller and C. Busby-Earle, "Multi-perspective machine learning a classifier ensemble method for intrusion detection," in *Proc. Int. Conf. Mach. Learn. Soft Comput. (ICMLSC)*, 2017, pp. 7–12.
[219] V. Narayan and D. Shanmugapriya, "Big data analytics with machine learning and deep learning methods for detection of anomalies in network traffic," in *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*. Hershey, PA, USA: IGI Global, 2020, pp. 317–346.
[220] L. Jiang, H. Zhang, and Z. Cai, "A novel Bayes model: Hidden naive Bayes," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 10, pp. 1361–1371, Oct. 2009.

[221] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
[222] E. Frank and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2011.
[223] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. 11th Int. Conf. Data Eng.*, 1995, pp. 3–14.
[224] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, Mar. 1996.
[225] Q. J. Ross, *C4. 5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.
[226] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.
[227] Q. Tian, D. Han, K.-C. Li, X. Liu, L. Duan, and A. Castiglione, "An intrusion detection approach based on improved deep belief network," *Int. J. Speech Technol.*, vol. 50, no. 10, pp. 3162–3178, Oct. 2020.
[228] S. Sathasivam and W. A. T. W. Abdullah, "Logic learning in hopfield networks," 2008, *arXiv:0804.4075*. [Online]. Available: http://arxiv.org/abs/0804.4075
[229] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5353–5360.
[230] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, "Multinomial naive Bayes for text categorization revisited," in *Proc. Australas. Joint Conf. Artif. Intell.* Berlin, Germany: Springer, 2004, pp. 488–499.
[231] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *Proc. Workshop Learn. Text Categorization (AAAI)*, vol. 752, no. 1. Madison, WI, USA: Citeseer, 1998, pp. 41–48. [Online]. Available: http://www.cs.cmu.edu/~mccallum/textcat.html
[232] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proc. 11th Conf. Uncertainty Artif. Intell.* San Mateo, CA, USA: Morgan Kaufmann, 1995, pp. 338–345.
[233] A. Jagota, "Novelty detection on a very large number of memories stored in a hopfield-style network," in *Proc. Seattle Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 2, 1991, p. 905.
[234] M. F. Augusteijn and B. A. Folkert, "Neural network classification and novelty detection," *Int. J. Remote Sens.*, vol. 23, no. 14, pp. 2891–2902, Jan. 2002.
[235] D. Martinez, "Neural tree density estimation for novelty detection," *IEEE Trans. Neural Netw.*, vol. 9, no. 2, pp. 330–338, Mar. 1998.
[236] P. Taveras and L. Hernandez, "Supervised machine learning techniques, cybersecurity habits and human generated password entropy for hacking prediction," in *Proc. MWAIS*, vol. 38, 2018, pp. 1–6. [Online]. Available: http://aisel.aisnet.org/mwais2018/38
[237] J. M. Gómez Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García, "Content based SMS spam filtering," in *Proc. ACM Symp. Document Eng. (DocEng)*, 2006, pp. 107–114.
[238] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Netw.*, vol. 1, no. 2, pp. 119–130, Jan. 1988.
[239] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.
[240] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[241] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[242] S. Lawrence, C. L. Giles, A. Chung Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.
[243] I. Wallach, M. Dzamba, and A. Heifets, "AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery," 2015, *arXiv:1510.02855*. [Online]. Available: http://arxiv.org/abs/1510.02855
[244] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at microsoft," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 3009–3017.
[245] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1222–1228.
[246] R. U. Khan, X. Zhang, M. Alazab, and R. Kumar, "An improved convolutional neural network model for intrusion detection in networks," in *Proc. Cybersecurity Cyberforensics Conf. (CCC)*, May 2019, pp. 74–77.
[247] K. Millar, A. Cheng, H. G. Chew, and C.-C. Lim, "Using convolutional neural networks for classifying malicious network traffic," in *Deep Learning Applications for Cyber Security*. Cham, Switzerland: Springer, 2019, pp. 103–126.
[248] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2017, pp. 712–717.
[249] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Jul. 2017, pp. 43–48.
[250] Y.-J. Zheng, W.-G. Sheng, X.-M. Sun, and S.-Y. Chen, "Airline passenger profiling based on fuzzy deep machine learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 2911–2923, Dec. 2017.
[251] F. He, Y. Zhang, D. Liu, Y. Dong, C. Liu, and C. Wu, "Mixed wavelet-based neural network model for cyber security situation prediction using MODWT and Hurst exponent analysis," in *Proc. Int. Conf. Netw. Syst. Secur.* Cham, Switzerland: Springer, 2017, pp. 99–111.
[252] G. Tzortzis and A. Likas, "Deep belief networks for spam filtering," in *Proc. 19th IEEE Int. Conf. Tools Artif. Intell. (ICTAI)*, vol. 2, Oct. 2007, pp. 306–309.
[253] X. Xu and T. Xie, "A reinforcement learning approach for host-based intrusion detection using sequences of system calls," in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2005, pp. 995–1003.
[254] X. Xu, Y. Sun, and Z. Huang, "Defending DDoS attacks using hidden Markov models and cooperative reinforcement learning," in *Proc. Pacific–Asia Workshop Intell. Secur. Inform.* Berlin, Germany: Springer, 2007, pp. 196–207.
[255] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decis. Support Syst.*, vol. 107, pp. 88–102, Mar. 2018.
[256] M. Feng and H. Xu, "Deep reinforecement learning based optimal defense for cyber-physical system in presence of unknown cyber-attack," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–8.
[257] *WEKA*. Accessed: Dec. 28, 2019. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/
[258] *Caffe*. Accessed: Dec. 28, 2019. [Online]. Available: http://caffe.berkeleyvision.org/
[259] *Torch*. Accessed: Dec. 28, 2019. [Online]. Available: http://torch.ch/
[260] *Keras*. Accessed: Dec. 28, 2019. [Online]. Available: https://github.com/EderSantana/keras
[261] *TensorFlow*. Accessed: Dec. 28, 2019. [Online]. Available: https://www.tensorflow.org/
[262] *Theano*. Accessed: Dec. 28, 2019. [Online]. Available: http://deeplearning.net/software/theano/
[263] *Shogun*. Accessed: Dec. 28, 2019. [Online]. Available: https://www.shogun-toolbox.org/
[264] *Accord.Net*. Accessed: Dec. 28, 2019. [Online]. Available: http://accord-framework.net/
[265] *MXNET An Efficient Library for Deep Learning*. Accessed: Aug. 13, 2020. [Online]. Available: https://mxnet.apache.org/versions/1.6/
[266] *Lasagne*. Accessed: Aug. 13, 2020. [Online]. Available: https://github.com/Lasagne
[267] *Blocks*. Accessed: Aug. 13, 2020. [Online]. Available: https://github.com/mila-udem/blocks
[268] *Deeplearning4j*. Accessed: Aug. 13, 2020. [Online]. Available: http://deeplearning4j.org
[269] *CNKT*. Accessed: Aug. 13, 2020. [Online]. Available: https://www.microsoft.com/en-us/cognitive-toolkit/
[270] *Core ML*. Accessed: Aug. 13, 2020. [Online]. Available: https://developer.apple.com/documentation/coreml
[271] *NCNN*. Accessed: Aug. 13, 2020. [Online]. Available: https://github.com/Tencent/ncnn
[272] *DeepSence*. Accessed: Aug. 13, 2020. [Online]. Available: https://deepsense.ai/
[273] S. M. Lee, D. S. Kim, J. H. Kim, and J. S. Park, "Spam detection using feature selection and parameters optimization," in *Proc. Int. Conf. Complex, Intell. Softw. Intensive Syst.*, Feb. 2010, pp. 883–888.

[274] N. F. Shah and P. Kumar, "A comparative analysis of various spam classifications," in *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*. Singapore: Springer, 2018, pp. 265–271.

[275] I. Alsmadi and I. Alhami, "Clustering and classification of email contents," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 27, no. 1, pp. 46–57, Jan. 2015.

[276] A. A. A. Abdelrahim, A. A. E. Elhadi, H. Ibrahim, and N. Elmisbah, "Feature selection and similarity coefficient based method for email spam filtering," in *Proc. Int. Conf. Comput., Electr. Electron. Eng. (ICCEEE)*, Aug. 2013, pp. 630–633.

[277] D. C. Chandrasekar, "Classification techniques using spam filtering email," *Int. J. Adv. Res. Comput. Sci.*, vol. 9, no. 2, pp. 402–410, Apr. 2018.

[278] L. H. Gomes, C. Cazita, J. M. Almeida, V. Almeida, and W. Meira, "Characterizing a spam traffic," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas. (IMC)*, 2004, pp. 356–369.

[279] M. Chakraborty, S. Pal, R. Pramanik, and C. R. Chowdary, "Recent developments in social spam detection and combating techniques: A survey," *Inf. Process. Manage.*, vol. 52, no. 6, pp. 1053–1073, Nov. 2016.

[280] A. S. Manisha and M. D. R. Jain, "Data pre-processing in spam detection," *Int. J. Sci. Technol. Eng.*, vol. 1, no. 11, p. 5, May 2015.

[281] A. A. Elhadi, M. A. Maarof, and A. H. Osman, "Malware detection based on hybrid signature behaviour application programming interface call graph," *Amer. J. Appl. Sci.*, vol. 9, no. 3, p. 283, 2012.

[282] C. Castillo and B. D. Davison, "Adversarial Web search," *Found. Trends Inf. Retr.*, vol. 4, no. 5, pp. 377–486, 2011.

[283] H. Stern, "A survey of modern spam tools," in *Proc. CEAS*, 2008, p. 136.

[284] B. Mehta, T. Hofmann, and P. Fankhauser, "Lies and propaganda: Detecting spam users in collaborative filtering," in *Proc. 12th Int. Conf. Intell. Interfaces (IUI)*, 2007, pp. 14–21.

[285] D. DeBarr and H. Wechsler, "Spam detection using clustering, random forests, and active learning," in *Proc. 6th Conf. Email Anti-Spam*. Mountain View, CA, USA: Citeseer, 2009, pp. 1–6.

[286] A. Ramachandran, N. Feamster, and S. Vempala, "Filtering spam with behavioral blacklisting," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 342–351.

[287] H. Ponnapalli, D. Herts, and J. Pablo, "Analysis and detection of modern spam techniques on social networking sites," in *Proc. 3rd Int. Conf. Services Emerg. Markets*, 2012, pp. 147–152.

[288] S. R. Kiran, "Spam or not spam–that is the question," SemanticScholar, Tech. Rep., 2009. [Online]. Available: https://courses.cs.washington.edu/courses/cse573/04au/Project/mini2/Ravi&Indri/spamfilter_ravi_indri.pdf

[289] S. Sharma and A. Arora, "Adaptive approach for spam detection," *Int. J. Comput. Sci. Issues*, vol. 10, no. 4, p. 23, 2013.

[290] M. A. A. M. Foqaha, "Email spam classification using hybrid approach of RBF neural network and particle swarm optimization," *Int. J. Netw. Secur. Appl.*, vol. 8, no. 4, pp. 17–28, 2016.

[291] A. Sharaff, N. K. Nagwani, and A. Dhadse, "Comparative study of classification algorithms for spam email detection," in *Emerging Research in Computing, Information, Communication and Applications*. New Delhi, India: Springer, 2016, pp. 237–244.

[292] A. R. Behjat, A. Mustapha, H. Nezamabadi-pour, M. N. Sulaiman, and N. Mustapha, "GA-based feature subset selection in a spam/non-spam detection system," in *Proc. Int. Conf. Comput. Commun. Eng. (ICCCE)*, Jul. 2012, pp. 675–679.

[293] A. Araúzo-Azofra and J. M. Benítez, "Empirical study of feature selection methods in classification," in *Proc. 8th Int. Conf. Hybrid Intell. Syst.*, Sep. 2008, pp. 584–589.

[294] S. Nizamani, N. Memon, U. K. Wiil, and P. Karampelas, "Modeling suspicious email detection using enhanced feature selection," 2013, *arXiv:1312.1971*. [Online]. Available: http://arxiv.org/abs/1312.1971

[295] S. K. Trivedi and P. K. Panigrahi, "Spam classification: A comparative analysis of different boosted decision tree approaches," *J. Syst. Inf. Technol.*, vol. 20, no. 3, pp. 105–298, Aug. 2018.

[296] S. Y. Bhat, M. Abulaish, and A. A. Mirza, "Spammer classification using ensemble methods over structural social network features," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. (WI) Intell. Agent Technol. (IAT)*, vol. 2, Aug. 2014, pp. 454–458.

[297] M. Scholar, "Supervised learning approach for spam classification analysis using data mining tools," *Organization*, vol. 2, no. 8, pp. 2760–2766, 2010.

[298] S. Youn and D. McLeod, "A comparative study for email classification," in *Proc. Adv. Innov. Syst., Comput. Sci. Softw. Eng.* Dordrecht, The Netherlands: Springer, 2007, pp. 387–391.

[299] R. M. Silva, A. Yamakami, and T. A. Almeida, "An analysis of machine learning methods for spam host detection," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, vol. 2, Dec. 2012, pp. 227–232.

[300] J. C. Gomez and M.-F. Moens, "PCA document reconstruction for email classification," *Comput. Statist. Data Anal.*, vol. 56, no. 3, pp. 741–751, Mar. 2012.

[301] A. Kantchelian, J. Ma, L. Huang, S. Afroz, A. Joseph, and J. D. Tygar, "Robust detection of comment spam using entropy rate," in *Proc. 5th ACM Workshop Secur. Artif. Intell. (AISec)*, 2012, pp. 59–70.

[302] T. Subramaniam, H. A. Jalab, and A. Y. Taqa, "Overview of textual anti-spam filtering techniques," *Int. J. Phys. Sci.*, vol. 5, no. 12, pp. 1869–1882, 2010.

[303] P. Chhabra, R. Wadhvani, and S. Shukla, "Spam filtering using support vector machine," *Int. J. Comput. Commun. Technol.*, vol. 1, no. 2, pp. 256–261, Oct. 2010.

[304] M. Soranamageswari and C. Meena, "A novel approach towards image spam classification," *Int. J. Comput. Eng.*, vol. 3, no. 1, p. 84, 2011.

[305] M. Mccord and M. Chuah, "Spam detection on Twitter using traditional classifiers," in *Proc. Int. Conf. Autonomic Trusted Comput.* Berlin, Germany: Springer, 2011, pp. 175–186.

[306] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time URL spam filtering service," in *Proc. IEEE Symp. Secur. Privacy*, May 2011, pp. 447–462.

[307] S. Sharma and A. Arora, "Adaptive approach for spam detection," *Int. J. Comput. Sci. Issues*, vol. 10, no. 4, p. 23, 2013.

[308] A. Arram, H. Mousa, and A. Zainal, "Spam detection using hybrid artificial neural network and genetic algorithm," in *Proc. 13th Int. Conf. Intell. Syst. Design Appl.*, Dec. 2013, pp. 336–340.

[309] M. Rathi and V. Pareek, "Spam mail detection through data mining-A comparative performance analysis," *Int. J. Mod. Educ. Comput. Sci.*, vol. 5, no. 12, p. 31, 2013.

[310] S. Mohammed, O. Mohammed, J. Fiaidhi, S. Fong, and T. H. Kim, "Classifying unsolicited bulk email (UBE) using python machine learning techniques," *Int. J. Hybrid Inf. Technol.*, vol. 6, no. 1, pp. 43–56, 2013.

[311] H. Najadat, N. Abdulla, R. Abooraig, and S. Nawasrah, "Mobile sms spam filtering based on mixing classifiers," *Int. J. Adv. Comput. Res.*, vol. 1, pp. 1–7, 2014.

[312] S. A. Saab, N. Mitri, and M. Awad, "Ham or spam? A comparative study for some content-based classification algorithms for email filtering," in *Proc. 17th IEEE Medit. Electrotechn. Conf. (MELECON)*, Apr. 2014, pp. 339–343.

[313] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowl.-Based Syst.*, vol. 64, pp. 22–31, Jul. 2014.

[314] C. Chen, J. Zhang, Y. Xie, Y. Xiang, W. Zhou, M. M. Hassan, A. AlElaiwi, and M. Alrubaian, "A performance evaluation of machine learning-based streaming spam tweets detection," *IEEE Trans. Comput. Social Syst.*, vol. 2, no. 3, pp. 65–76, Sep. 2015.

[315] D. K. Renuka, P. Visalakshi, and T. Sankar, "Improving E-mail spam classification using ant colony optimization algorithm," *Int. J. Comput. Appl.*, pp. 22–26, Jan. 2015.

[316] T. Vyas, P. Prajapati, and S. Gadhwal, "A survey and evaluation of supervised machine learning techniques for spam e-mail filtering," in *Proc. IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Mar. 2015, pp. 1–7.

[317] Z. Khan and U. Qamar, "Text mining approach to detect spam in emails," in *Proc. Int. Conf. Innov. Intell. Syst. Comput. Technol. (ICIISCT)*, 2016, p. 45.

[318] I. J. Alkaht and B. Al Khatib, "Filtering SPAM using several stages neural networks," *Int. Rev. Comput. Softw.*, vol. 11, no. 2, p. 123, Feb. 2016.

[319] A. Tyagi, "Content based spam classification—A deep learning approach," M.S. thesis, Dept. Comput. Sci., Univ. Calgary, Calgary, AB, Canada, 2016.

[320] H. Xu, W. Sun, and A. Javaid, "Efficient spam detection across online social networks," in *Proc. IEEE Int. Conf. Big Data Anal. (ICBDA)*, Mar. 2016, pp. 1–6.

[321] E.-X. Shang and H.-G. Zhang, "Image spam classification based on convolutional neural network," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, vol. 1, Jul. 2016, pp. 398–403.

[322] G. Jain, M. Sharma, and B. Agarwal, "Spam detection on social media using semantic convolutional neural network," *Int. J. Knowl. Discovery Bioinf.*, vol. 8, no. 1, pp. 12–26, Jan. 2018.

[323] M. Bassiouni, M. Ali, and E. A. El-Dahshan, "Ham and spam e-mails classification using machine learning techniques," *J. Appl. Secur. Res.*, vol. 13, no. 3, pp. 315–331, Jul. 2018.

[324] A. Annadatha and M. Stamp, "Image spam analysis and detection," *J. Comput. Virol. Hacking Techn.*, vol. 14, no. 1, pp. 39–52, 2018.
[325] R. Vinayakumar, M. Alazab, A. Jolfaei, K. P. Soman, and P. Poornachandran, "Ransomware triage using deep learning: Twitter as a case study," in *Proc. Cybersecurity Cyberforensics Conf. (CCC)*, May 2019, pp. 67–73.
[326] M. Diale, T. Celik, and C. Van Der Walt, "Unsupervised feature learning for spam email filtering," *Comput. Electr. Eng.*, vol. 74, pp. 89–104, Mar. 2019.
[327] G. Jain, M. Sharma, and B. Agarwal, "Optimizing semantic LSTM for spam detection," *Int. J. Inf. Technol.*, vol. 11, no. 2, pp. 239–250, Jun. 2019.
[328] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep learning to filter SMS spam," *Future Gener. Comput. Syst.*, vol. 102, pp. 524–533, Jan. 2020.
[329] T. Gangavarapu, C. Jaidhar, and B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: Review and approaches," *Artif. Intell. Rev.*, vol. 53, pp. 1–63, Feb. 2020.
[330] L. GuangJun, S. Nazir, H. U. Khan, and A. U. Haq, "Spam detection approach for secure mobile message communication using machine learning algorithms," *Secur. Commun. Netw.*, vol. 2020, pp. 1–6, Jul. 2020.
[331] Z. Chu, I. Widjaja, and H. Wang, "Detecting social spam campaigns on Twitter," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2012, pp. 455–472.
[332] A. Aggarwal, J. Almeida, and P. Kumaraguru, "Detection of spam tipping behaviour on foursquare," in *Proc. 22nd Int. Conf. World Wide Web (WWW Companion)*, 2013, pp. 641–648.
[333] C.-C. Lai, "An empirical study of three machine learning methods for spam filtering," *Knowl.-Based Syst.*, vol. 20, no. 3, pp. 249–254, Apr. 2007.
[334] M. Bassiouni, M. Ali, and E. A. El-Dahshan, "Ham and spam E-Mails classification using machine learning techniques," *J. Appl. Secur. Res.*, vol. 13, no. 3, pp. 315–331, Jul. 2018.
[335] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10206–10222, Sep. 2009.
[336] A. Gupta, C. Singhal, and S. Aggarwal, "An improved anti spam filter based on content, low level features and noise," in *Proc. Int. Conf. Comput. Sci. Inf. Technol.* Berlin, Germany: Springer, 2012, pp. 563–572.
[337] P. Li, H. Yan, G. Cui, and Y. Du, "Integration of local and global features for image spam filtering," *J. Comput. Inf. Syst.*, vol. 8, no. 2, pp. 779–789, 2012.
[338] B. Biggio, G. Fumera, I. Pillai, and F. Roli, "A survey and experimental evaluation of image spam filtering techniques," *Pattern Recognit. Lett.*, vol. 32, no. 10, pp. 1436–1446, Jul. 2011.
[339] Z. Al Muataz and N. A. Aziz, "A new efficient text detection method for image spam filtering," *Int. Rev. Comput. Softw.*, vol. 10, no. 1, pp. 1–8, 2015.
[340] T.-J. Liu, C.-N. Wu, C.-L. Lee, and C.-W. Chen, "A self-adaptable image spam filtering system," *J. Chin. Inst. Eng.*, vol. 37, no. 4, pp. 517–528, May 2014.
[341] A. S. Manek, D. K. Shamini, V. H. Bhat, P. D. Shenoy, M. C. Mohan, K. R. Venugopal, and L. M. Patnaik, "ReP-ETD: A repetitive preprocessing technique for embedded text detection from images in spam emails," in *Proc. IEEE Int. Advance Comput. Conf. (IACC)*, Feb. 2014, pp. 568–573.
[342] S. Wakade, K. J. Liszka, and C.-C. Chan, "Application of learning algorithms to image spam evolution," in *Emerging Paradigms in Machine Learning*. Berlin, Germany: Springer, 2013, pp. 471–495.
[343] A. Attar, R. M. Rad, and R. E. Atani, "A survey of image spamming and filtering techniques," *Artif. Intell. Rev.*, vol. 40, no. 1, pp. 71–105, Jun. 2013.
[344] C. Romero, M. Garcia-Valdez, and A. Alanis, "A comparative study of blog comments spam filtering with machine learning techniques," in *Soft Computing for Recognition Based on Biometrics*. Berlin, Germany: Springer, 2010, pp. 57–72.
[345] S. Abu-Nimeh and T. Chen, "Proliferation and detection of blog spam," *IEEE Secur. Privacy Mag.*, vol. 8, no. 5, pp. 42–47, Sep. 2010.
[346] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi, "Detecting spam blogs: A machine learning approach," in *Proc. Nat. Conf. Artif. Intell.*, vol. 21, no. 2. Cambridge, MA, USA: AAAI Press, 2006, p. 1351.
[347] T. Yoshinaka, S. Ishii, T. Fukuhara, H. Masuda, and H. Nakagawa, "A user-oriented splog filtering based on a machine learning," in *Recent Trends and Developments in Social Software*. Berlin, Germany: Springer, 2008, pp. 88–99.
[348] D. Sculley and G. M. Wachman, "Relaxed online SVMs for spam filtering," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2007, pp. 415–422.
[349] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, C. Zhang, and K. Ross, "Identifying video spammers in online social networks," in *Proc. 4th Int. Workshop Adversarial Inf. Retr. Web (AIRWeb)*, 2008, pp. 45–52.
[350] K. Indira and E. C. Joy, "Prevention of spammers and Promoters in Video Social Networks using SVM-KNN," *Int. J. Eng. Technol.*, vol. 6, no. 5, pp. 2024–2030, Oct./Nov. 2014.
[351] Y. Zhang, S. Wang, and L. Wu, "Spam detection via feature selection and decision tree," *Adv. Sci. Lett.*, vol. 5, no. 2, pp. 726–730, Feb. 2012.
[352] R. K. Kumar, G. Poonkuzhali, and P. Sudhakar, "Comparative study on email spam classifier using data mining techniques," in *Proc. Int. MultiConf. Eng. Comput. Scientists*, vol. 1, 2012, pp. 14–16.
[353] J. Datta, N. Kataria, and N. Hubballi, "Network traffic classification in encrypted environment: A case study of Google hangout," in *Proc. 21st Nat. Conf. Commun. (NCC)*, Feb. 2015, pp. 1–6.
[354] S. Goyal, R. K. Chauhan, and S. Parveen, "Spam detection using KNN and decision tree mechanism in social network," in *Proc. 4th Int. Conf. Parallel, Distrib. Grid Comput. (PDGC)*, 2016, pp. 522–526.
[355] M. Z. Gashti, "Detection of spam email by combining harmony search algorithm and decision tree," *Eng., Technol. Appl. Sci. Res.*, vol. 7, no. 3, pp. 1713–1718, Jun. 2017.
[356] K. R. Dhanaraj and V. Palaniswami, "Firefly and Bayes classifier for email spam classification in a distributed environment," *Aust. J. Basic Appl. Sci.*, vol. 8, no. 17, pp. 118–130, 2014.
[357] Z. Jorgensen, Y. Zhou, and M. Inge, "A multiple instance learning strategy for combating good word attacks on spam filters," *J. Mach. Learn. Res.*, vol. 9, pp. 1115–1146, Jun. 2008.
[358] P. P. K. Chan, C. Yang, D. S. Yeung, and W. W. Y. Ng, "Spam filtering for short messages in adversarial environment," *Neurocomputing*, vol. 155, pp. 167–176, May 2015.
[359] N. Choudhary and A. K. Jain, "Towards filtering of SMS spam messages using machine learning based technique," in *Proc. Int. Conf. Adv. Informat. Comput. Res.* Singapore: Springer, 2017, pp. 18–30.
[360] M. K. Tai, "Recognize and evaluate security framework of classifier under attack," *Int. J. Res. Anal. Rev.*, vol. 6, no. 2, pp. 167–170, 2019.
[361] *SolarWinds MSP Mail Assure*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.solarwindsmsp.com/products/mail
[362] *SpamTitan*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.spamtitan.com/
[363] *SPAMfighter*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.spamfighter.com/SPAMfighter/Product_Info.asp
[364] *ZEROSPAM*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.zerospam.ca/en/home/
[365] E.-S.-M. El-Alfy and A. A. AlHasan, "Spam filtering framework for multimodal mobile communication based on dendritic cell algorithm," *Future Gener. Comput. Syst.*, vol. 64, pp. 98–107, Nov. 2016.
[366] S. J. Delany, M. Buckley, and D. Greene, "SMS spam filtering: Methods and data," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9899–9908, Aug. 2012.
[367] I. Ahmed, D. Guan, and T. C. Chung, "SMS classification based on naive Bayes classifier and Apriori algorithm frequent itemset," *Int. J. Mach. Learn. Comput.*, vol. 4, no. 2, p. 183, 2014.
[368] R. Taheri and R. Javidan, "Spam filtering in SMS using recurrent neural networks," in *Proc. Artif. Intell. Signal Process. Conf. (AISP)*, Oct. 2017, pp. 331–336.
[369] A. Karami and L. Zhou, "Improving static SMS spam detection by using new content-based features," in *Proc. 12th Amer. Conf. Inf. Syst.*, Savannah, GA, USA, 2014, pp. 1–9. [Online]. Available: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1205&context=amcis2014
[370] H.-Y. Zhang and W. Wang, "Application of Bayesian method to spam SMS filtering," in *Proc. Int. Conf. Inf. Eng. Comput. Sci.*, Dec. 2009, pp. 1–3.
[371] W. Liu and T. Wang, "Index-based online text classification for SMS spam filtering," *J. Comput.*, vol. 5, no. 6, pp. 844–851, Jun. 2010.
[372] L. Duan, N. Li, and L. Huang, "A new spam short message classification," in *Proc. 1st Int. Workshop Edu. Technol. Comput. Sci.*, vol. 2, 2009, pp. 168–171.
[373] K. Mathew and B. Issac, "Intelligent spam classification for mobile text message," in *Proc. Int. Conf. Comput. Sci. Netw. Technol.*, vol. 1, Dec. 2011, pp. 101–105.
[374] R. K. Solanki, K. Verma, and R. Kumar, "Spam filtering using hybrid local-global naive Bayes classifier," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Aug. 2015, pp. 829–833.

[375] D. Gunawan, R. F. Rahmat, A. Putra, and M. F. Pasha, "Filtering spam text messages by using Twitter-LDA algorithm," in *Proc. IEEE Int. Conf. Commun., Netw. Satell. (Comnetsat)*, Nov. 2018, pp. 1–6.

[376] R. K. Kaliyar, P. Narang, and A. Goswami, "SMS spam filtering on multiple background datasets using machine learning techniques: A novel approach," in *Proc. IEEE 8th Int. Advance Comput. Conf. (IACC)*, Dec. 2018, pp. 59–65.

[377] B. Biggio, G. Fumera, I. Pillai, and F. Roli, "Image spam filtering using visual information," in *Proc. 14th Int. Conf. Image Anal. Process. (ICIAP)*, Sep. 2007, pp. 105–110.

[378] A. D. Kumar and S. Kp, "DeepImageSpam: Deep learning based image spam detection," 2018, *arXiv:1810.03977*. [Online]. Available: http://arxiv.org/abs/1810.03977

[379] A. Bhowmick and S. M. Hazarika, "Machine learning for E-mail spam filtering: Review,Techniques and trends," 2016, *arXiv:1606.01042*. [Online]. Available: http://arxiv.org/abs/1606.01042

[380] F. Aiwan and Y. Zhaofeng, "Image spam filtering using convolutional neural networks," *Pers. Ubiquitous Comput.*, vol. 22, nos. 5–6, pp. 1029–1037, Oct. 2018.

[381] S. Patil, T. Angre, H. Bhanushali, and Y. Tank, "Classification and spam image detection in smartphones," in *Proc. 2nd Int. Conf. Adv. Sci. Technol.*, 2019, pp. 1–7.

[382] A. Sachdeva, R. Kapoor, A. Sharma, and A. Mishra, "Categorical classification and deletion of spam images on smartphones using image processing and machine learning," in *Proc. Int. Conf. Mach. Learn. Data Sci. (MLDS)*, Dec. 2017, pp. 23–30.

[383] H. Li, X. Xu, C. Liu, T. Ren, K. Wu, X. Cao, W. Zhang, Y. Yu, and D. Song, "A machine learning approach to prevent malicious calls over telephony networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 53–69.

[384] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, "Practical real-time intrusion detection using machine learning approaches," *Comput. Commun.*, vol. 34, no. 18, pp. 2227–2235, Dec. 2011.

[385] F. Amiri, M. R. Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1184–1199, Jul. 2011.

[386] S.-W. Lin, K.-C. Ying, C.-Y. Lee, and Z.-J. Lee, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection," *Appl. Soft Comput.*, vol. 12, no. 10, pp. 3285–3290, Oct. 2012.

[387] S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, and F. Herrera, "On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 193–202, Jan. 2015.

[388] H. Gharaee and H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM," in *Proc. 8th Int. Symp. Telecommun. (IST)*, Sep. 2016, pp. 139–144.

[389] D. Boughaci, M. D. E. Kadi, and M. Kada, "Fuzzy particle swarm optimization for intrusion detection," in *Proc. Int. Conf. Neural Inf. Process.* Berlin, Germany: Springer, 2012, pp. 541–548.

[390] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," *Comput. Commun.*, vol. 35, no. 7, pp. 772–783, Apr. 2012.

[391] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowl.-Based Syst.*, vol. 78, pp. 13–21, Apr. 2015.

[392] T. Lane and C. E. Brodley, "An application of machine learning to anomaly detection," in *Proc. 20th Nat. Inf. Syst. Secur. Conf.*, Baltimore, MD, USA, vol. 377, 1997, pp. 366–380.

[393] L. Khan, M. Awad, and B. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," *VLDB J.*, vol. 16, no. 4, pp. 507–521, Aug. 2007.

[394] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6225–6232, Sep. 2010.

[395] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, and C. D. Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 306–313, 2011.

[396] A. M. Chandrasekhar and K. Raghuveer, "Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers," in *Proc. Int. Conf. Comput. Commun. Informat.*, Jan. 2013, pp. 1–7.

[397] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1690–1700, Mar. 2014.

[398] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 303–336, 1st Quart., 2014.

[399] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 686–728, 1st Quart., 2019.

[400] M. D. Rich, "Evaluating machine learning classifiers for hybrid network intrusion detection systems," M.S. thesis, Wright-Patterson AFB OH Graduate School Eng. Manage., Air Force Inst. Technol., Wright-Patterson AFB, OH, USA, 2015.

[401] A. B. Akhi, E. J. Kanon, A. Kabir, and A. Banu, "Network intrusion classification employing machine learning: A survey," Ph.D. dissertation, Dept. Comput. Sci. Eng., United Int. Univ., Dhaka, Bangladesh, 2019.

[402] R. Jamadar, S. Ingale, A. Panhalkar, A. Kakade, and M. Shinde, "Survey of deep learning based intrusion detection systems for cyber security," *Int. J. Res. Anal. Rev.*, vol. 6, no. 2, pp. 257–261, 2019.

[403] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[404] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci.*, vol. 378, pp. 484–497, Feb. 2017.

[405] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *Proc. IEEE 15th Int. Symp. Intell. Syst. Informat. (SISY)*, Sep. 2017, pp. 000277–000282.

[406] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: http://arxiv.org/abs/1901.03407

[407] A. Torkaman, G. Javadzadeh, and M. Bahrololum, "A hybrid intelligent HIDS model using two-layer genetic algorithm and neural network," in *Proc. 5th Conf. Inf. Knowl. Technol.*, May 2013, pp. 92–96.

[408] R. Puzis, M. D. Klippel, Y. Elovici, and S. Dolev, "Optimization of NIDS placement for protection of intercommunicating critical infrastructures," in *Proc. Eur. Conf. Intell. Secur. Inform.* Berlin, Germany: Springer, 2008, pp. 191–203.

[409] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive bayes vs decision trees in intrusion detection systems," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2004, pp. 420–424.

[410] Y. Bouzida and F. Cuppens, "Neural networks vs. decision trees for intrusion detection," in *Proc. IEEE/IST Workshop Monitor., Attack Detection Mitigation (MonAM)*, vol. 28. Citeseer, 2006, p. 29.

[411] D. S. Kim and J. S. Park, "Network-based intrusion detection with support vector machines," in *Proc. Int. Conf. Inf. Netw.* Berlin, Germany: Springer, 2003, pp. 747–756.

[412] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, "Intrusion detection using fuzzy association rules," *Appl. Soft Comput.*, vol. 9, no. 2, pp. 462–469, Mar. 2009.

[413] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proc. IEEE Symp. Secur. Privacy*, May 1999, pp. 120–132.

[414] A. K. Ghosh and A. Schwartzbard, "A study in using neural networks for anomaly and misuse detection," in *Proc. USENIX Secur. Symp.*, vol. 99, 1999, p. 12.

[415] J. Cannady, "Artificial neural networks for misuse detection," in *Proc. Nat. Inf. Syst. Secur. Conf.*, Baltimore, MD, USA, vol. 26, 1998, pp. 443–456.

[416] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Comput. Secur.*, vol. 24, no. 4, pp. 295–307, Jun. 2005.

[417] M. G. Schultz, E. Eskin, F. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in *Proc. IEEE Symp. Secur. Privacy. (S P)*, May 2000, pp. 38–49.

[418] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to DDoS attack detection and response," in *Proc. DARPA Inf. Survivability Conf. Exposit.*, vol. 1, 2003, pp. 303–314.

[419] M.-Y. Su, G.-J. Yu, and C.-Y. Lin, "A real-time network intrusion detection system for large-scale attacks based on an incremental mining approach," *Comput. Secur.*, vol. 28, no. 5, pp. 301–309, Jul. 2009.

[420] Y. Liao and V. R. Vemuri, "Use of K-Nearest neighbor classifier for intrusion detection," *Comput. Secur.*, vol. 21, no. 5, pp. 439–448, Oct. 2002.

[421] D. Endler, "Applying machine learning to Solaris audit data," in *Proc. Annu. Comput. Secur. Appl. Conf.*, 1998, pp. 268–279.
[422] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 227–261, Nov. 2000.
[423] D. M. Farid, N. Harbi, and M. Z. Rahman, "Combining naive Bayes and decision tree for adaptive intrusion detection," 2010, *arXiv:1005.4496*. [Online]. Available: https://arxiv.org/abs/1005.4496
[424] M. S. Mok, S. Y. Sohn, and Y. H. Ju, "Random effects logistic regression model for anomaly detection," *Expert Syst. Appl.*, vol. 37, no. 10, pp. 7162–7166, Oct. 2010.
[425] M. M. T. Jawhar and M. Mehrotra, "Design network intrusion detection system using hybrid fuzzy-neural network," *Int. J. Comput. Sci. Secur.*, vol. 4, no. 3, pp. 285–294, 2010.
[426] M. Yan and Z. Liu, "A new method of transductive SVM-based network intrusion detection," in *Proc. Int. Conf. Comput. Comput. Technol. Agricult.* Berlin, Germany: Springer, 2010, pp. 87–95.
[427] C. Wagner, J. François, and T. Engel, "Machine learning approach for ip-flow record anomaly detection," in *Proc. Int. Conf. Res. Netw.* Berlin, Germany: Springer, 2011, pp. 28–39.
[428] C. M. Rahman, D. M. Farid, and M. Z. Rahman, "Adaptive intrusion detection based on boosting and naive Bayesian classifier," Faculty Paper, 2011. [Online]. Available: http://103.109.52.4:8080/handle/52243/74
[429] M.-Y. Su, "Real-time anomaly detection systems for denial-of-service attacks by weighted k-nearest-neighbor classifiers," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3492–3498, Apr. 2011.
[430] S. Lee, G. Kim, and S. Kim, "Self-adaptive and dynamic clustering for online anomaly detection," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14891–14898, Nov. 2011.
[431] M. Sheikhan, Z. Jadidi, and A. Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping," *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1185–1190, Sep. 2012.
[432] S. K. Sharma, P. Pandey, S. K. Tiwari, and M. S. Sisodia, "An improved network intrusion detection technique based on k-means clustering via Naïve Bayes classification," in *Proc. IEEE Int. Conf. Adv. Eng., Sci. Manage. (ICAESM)*, Mar. 2012, pp. 417–422.
[433] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a hidden Naïve Bayes multiclass classifier," *Expert Syst. Appl.*, vol. 39, no. 18, pp. 13492–13500, Dec. 2012.
[434] S. S. Sivatha Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 129–141, Jan. 2012.
[435] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 424–430, Jan. 2012.
[436] A. Chandrashekhar and K. Raghuveer, "Fortification of hybrid intrusion detection system using variants of neural networks and support vector machines," *Int. J. Netw. Secur. Appl.*, vol. 5, no. 1, p. 71, 2013.
[437] M. M. Lisehroodi, Z. Muda, and W. Yassin, "A hybrid framework based on neural network MLP and K-means clustering for intrusion detection system," in *Proc. 4th Int. Conf. Comput. Inform. (ICOCI)*, 2013, pp. 1–7.
[438] S. Devaraju and S. Ramakrishnan, "Detection of accuracy for intrusion detection system using neural network classifier," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 1, pp. 338–345, 2013.
[439] W. Yassin, N. I. Udzir, Z. Muda, and M. N. Sulaiman, "Anomaly-based intrusion detection through k-means clustering and Naives Bayes classification," in *Proc. 4th Int. Conf. Comput. Inform. (ICOCI)*, vol. 49, 2013, pp. 1–6.
[440] Z. A. Baig, S. M. Sait, and A. Shaheen, "GMDH-based networks for intelligent intrusion detection," *Eng. Appl. Artif. Intell.*, vol. 26, no. 7, pp. 1731–1740, Aug. 2013.
[441] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," in *Proc. 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Dec. 2014, pp. 1–6.
[442] R. T. Kokila, S. T. Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in *Proc. 6th Int. Conf. Adv. Comput. (ICoAC)*, Dec. 2014, pp. 205–210.
[443] A. K. Shrivas and A. K. Dewangan, "An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set," *Int. J. Comput. Appl.*, vol. 99, no. 15, pp. 8–13, Aug. 2014.
[444] R. Ranjan and G. Sahoo, "A new clustering approach for anomaly intrusion detection," 2014, *arXiv:1404.2772*. [Online]. Available: http://arxiv.org/abs/1404.2772
[445] W. Feng, Q. Zhang, G. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Generat. Comput. Syst.*, vol. 37, pp. 127–140, Jul. 2014.
[446] M. V. Kotpalliwar and R. Wajgi, "Classification of attacks using support vector machine (SVM) on KDDCUP'99 IDS database," in *Proc. 5th Int. Conf. Commun. Syst. Netw. Technol.*, Apr. 2015, pp. 987–990.
[447] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2670–2679, Apr. 2015.
[448] B. W. Masduki, K. Ramli, F. A. Saputra, and D. Sugiarto, "Study on implementation of machine learning methods combination for improving attacks detection accuracy on intrusion detection system (IDS)," in *Proc. Int. Conf. Qual. Res. (QiR)*, Aug. 2015, pp. 56–64.
[449] N. G. Relan and D. R. Patil, "Implementation of network intrusion detection system using variant of decision tree algorithm," in *Proc. Int. Conf. Nascent Technol. Eng. Field (ICNTE)*, Jan. 2015, pp. 1–5.
[450] G. Kim, H. Yi, J. Lee, Y. Paek, and S. Yoon, "LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems," 2016, *arXiv:1611.01726*. [Online]. Available: http://arxiv.org/abs/1611.01726
[451] A. Hadri, K. Chougdali, and R. Touahni, "Intrusion detection system using PCA and fuzzy PCA techniques," in *Proc. Int. Conf. Adv. Commun. Syst. Inf. Secur. (ACOSIS)*, Oct. 2016, pp. 1–7.
[452] B. Subba, S. Biswas, and S. Karmakar, "Enhancing performance of anomaly based intrusion detection systems through dimensionality reduction using principal component analysis," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Nov. 2016, pp. 1–6.
[453] P. Vivek, S. Nandan, and R. Tiwari, "Enhanced method for intrusion detection over KDD cup 99 dataset," *Int. J. Current Trends Eng. Technol.*, vol. 2, no. 2, pp. 218–224, 2016. [Online]. Available: https://www.semanticscholar.org/paper/Enhanced-Method-for-Intrusion-Detection-over-KDD-99-Tiwari-Rathore/9b2ed9b997d35839e761f8dbe2b87a0c45cb88e6
[454] R. K. Sharma, H. K. Kalita, and P. Borah, "Analysis of machine learning techniques based intrusion detection systems," in *Proc. 3rd Int. Conf. Adv. Comput., Netw. Informat.* New Delhi, India: Springer, 2016, pp. 485–493.
[455] P. U. K. P. chandra and P. N. A. lilhore, "Network intrusion detection system based on modified Random forest classifiers for KDD cup-99 and NSL-KDD dataset," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 4, no. 8, pp. 786–791, 2017.
[456] J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Comput. Appl.*, vol. 28, no. S1, pp. 1051–1058, Dec. 2017.
[457] A. R. Syarif and W. Gata, "Intrusion detection system using hybrid binary PSO and K-nearest neighborhood algorithm," in *Proc. 11th Int. Conf. Inf. Commun. Technol. Syst. (ICTS)*, Oct. 2017, pp. 181–186.
[458] B. Xu, S. Chen, H. Zhang, and T. Wu, "Incremental k-NN SVM method in intrusion detection," in *Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Nov. 2017, pp. 712–717.
[459] T.-T.-H. Le, J. Kim, and H. Kim, "An effective intrusion detection classifier using long short-term memory with gradient descent optimization," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Feb. 2017, pp. 1–6.
[460] A. J. Malik and F. A. Khan, "A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection," *Cluster Comput.*, vol. 21, no. 1, pp. 667–680, Mar. 2018.
[461] M. Al-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial Internet of Things based on deep learning models," *J. Inf. Secur. Appl.*, vol. 41, pp. 1–11, Aug. 2018.
[462] Q. Zhang, Y. Qu, and A. Deng, "Network intrusion detection using kernel-based fuzzy-rough feature selection," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2018, pp. 1–6.
[463] J. Lee, J. Kim, I. Kim, and K. Han, "Cyber threat detection based on artificial neural networks using event profiles," *IEEE Access*, vol. 7, pp. 165607–165626, 2019.
[464] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, Feb. 2019.
[465] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," 2019, *arXiv:1904.01352*. [Online]. Available: http://arxiv.org/abs/1904.01352
[466] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, pp. 31711–31722, 2019.

[467] A.-U.-H. Qureshi, H. Larijani, N. Mtetwa, A. Javed, and J. Ahmad, "RNN-ABC: A new swarm optimization based technique for anomaly detection," *Computers*, vol. 8, no. 3, p. 59, Aug. 2019.
[468] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," in *Proc. ACM Southeast Conf. (ZZZ ACM SE)*, 2019, pp. 86–93.
[469] F. Salo, A. B. Nassif, and A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection," *Comput. Netw.*, vol. 148, pp. 164–175, Jan. 2019.
[470] J. Ghasemi, J. Esmaily, and R. Moradinezhad, "Intrusion detection system using an optimized kernel extreme learning machine and efficient features," *Sadhana*, vol. 45, no. 1, pp. 1–9, Dec. 2020.
[471] S. Sen, K. D. Gupta, and M. M. Ahsan, "Leveraging machine learning approach to setup software-defined network (SDN) controller rules during DDoS attack," in *Proc. Int. Joint Conf. Comput. Intell.* Singapore: Springer, 2020, pp. 49–60.
[472] Z. Liu, Y. Zhu, X. Yan, L. Wang, Z. Jiang, and J. Luo, "Deep learning approach for IDS," in *Proc. 4th Int. Congr. Inf. Commun. Technol.* Singapore: Springer, 2020, pp. 471–479.
[473] M. Sarnovsky and J. Paralic, "Hierarchical intrusion detection using machine learning and knowledge model," *Symmetry*, vol. 12, no. 2, p. 203, Feb. 2020.
[474] C. Kruegel and T. Toth, "Using decision trees to improve signature-based intrusion detection," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Berlin, Germany: Springer, 2003, pp. 173–191.
[475] A. Abraham, C. Grosan, and C. Martin-Vide, "Evolutionary design of intrusion detection programs," *IJ Netw. Secur.*, vol. 4, no. 3, pp. 328–339, 2007.
[476] A. Abraham, R. Jain, J. Thomas, and S. Y. Han, "D-SCIDS: Distributed soft computing intrusion detection system," *J. Netw. Comput. Appl.*, vol. 30, no. 1, pp. 81–98, Jan. 2007.
[477] S. Mukkamala and A. H. Sung, "A comparative study of techniques for intrusion detection," in *Proc. 15th IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2003, pp. 570–577.
[478] R. Mohan, V. Vaidehi, M. Mahalakshmi, and S. S. Chakkaravarthy, "Complex event processing based hybrid intrusion detection system," in *Proc. 3rd Int. Conf. Signal Process., Commun. Netw. (ICSCN)*, Mar. 2015, pp. 1–6.
[479] S. H. Vasudeo, P. Patil, and R. V. Kumar, "IMMIX-intrusion detection and prevention system," in *Proc. Int. Conf. Smart Technol. Manage. Comput., Commun., Controls, Energy Mater. (ICSTM)*, 2015, pp. 96–101.
[480] A. K. Ghosh, A. Schwartzbard, and M. Schatz, "Learning program behavior profiles for intrusion detection," in *Proc. Workshop Intrusion Detection Netw. Monitor.*, vol. 51462, 1999, pp. 1–13.
[481] J. Zhang and M. Zulkernine, "A hybrid network intrusion detection technique using random forests," in *Proc. 1st Int. Conf. Availability, Rel. Secur. (ARES)*, 2006, p. 269.
[482] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, nos. 1–2, pp. 18–28, Feb. 2009.
[483] T. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.
[484] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 1–35, Jan. 2010.
[485] K.-D. Althoff, R. Bergmann, S. Wess, M. Manago, E. Auriol, O. I. Larichev, A. Bolotov, Y. I. Zhuravlev, and S. I. Gurov, "Case-based reasoning for medical decision support tasks: The inreca approach," *Artif. Intell. Med.*, vol. 12, no. 1, pp. 25–41, Jan. 1998.
[486] M. Esmaili, B. Balachandran, R. Safavi-Naini, and J. Pieprzyk, "Case-based reasoning for intrusion detection," in *Proc. 12th Annu. Comput. Secur. Appl. Conf.*, 1996, pp. 214–223.
[487] J. Yang, T. Deng, and R. Sui, "An adaptive weighted one-class svm for robust outlier detection," in *Proc. Chin. Intell. Syst. Conf.* Berlin, Germany: Springer, 2016, pp. 475–484.
[488] G. Kumar and K. Kumar, "A multi-objective genetic algorithm based approach for effective intrusion detection using neural networks," in *Intelligent Methods for Cyber Warfare*. Cham, Switzerland: Springer, 2015, pp. 173–200.
[489] B. Sezari, D. P. F. Moller, and A. Deutschmann, "Anomaly-based network intrusion detection model using deep learning in airports," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun., 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1725–1729.
[490] M. Aljanabi, M. A. Ismail, and V. Mezhuyev, "Improved TLBO-JAYA algorithm for subset feature selection and parameter optimisation in intrusion detection system," *Complexity*, vol. 2020, pp. 1–18, May 2020.
[491] *McAfee Network Security Platform*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.mcafee.com/enterprise/en-au/products/network-security-platform.html
[492] *Hillstone S-Series Network Intrusion Prevention System (NIPS)*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.hillstonenet.com/products/network-intrusion-prevention-system-s-series/
[493] *NIP2000/5000 Intrusion Prevention System*. Accessed: Feb. 16, 2020. [Online]. Available: https://e.huawei.com/en/related-page/products/enterprise-network/security/application-gateway/nip-ips/security_nip2000_5000_ips_v2_en
[494] *Palo Alto Networks Completes Acquisition of The Crypsis Group*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.paloaltonetworks.com/
[495] *Dark Trace—Cyber AI Security*. Accessed: Oct. 10, 2020. [Online]. Available: https://www.darktrace.com/en/
[496] *Next-Generation Intrusion Prevention System (NGIPS)*. Accessed: Feb. 14, 2020. [Online]. Available: https://www.cisco.com/c/en_au/products/security/ngips/index.html
[497] *Snort 2.9.15.1*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.snort.org/
[498] *Suricata | Open Source IDS / IPS / NSM Engine*. Accessed: Feb. 16, 2020. [Online]. Available: https://suricata-ids.org/
[499] *The SAMHAIN File Integrity / Host-Based Intrusion Detection System*. Accessed: Feb. 16, 2020. [Online]. Available: https://la-samhna.de/samhain/
[500] *Security Onion*. Accessed: Feb. 16, 2020. [Online]. Available: https://securityonion.net/
[501] *THE SAGAN LOG ANALYSIS ENGINE*. Accessed: Feb. 16, 2020. [Online]. Available: https://quadrantsec.com/sagan_log_analysis_engine/
[502] *What Are STIX/TAXII*. Accessed: Nov. 16, 2020. [Online]. Available: https://www.anomali.com/resources/what-are-stix-taxii
[503] B. Sun, Z. Chen, R. Wang, F. Yu, and V. C. M. Leung, "Towards adaptive anomaly detection in cellular mobile networks," in *Proc. 3rd IEEE Consum. Commun. Netw. Conf. (CCNC)*, vol. 2, Jan. 2006, pp. 666–670.
[504] B. Sun, Y. Xiao, and K. Wu, "Intrusion detection in cellular mobile networks," in *Wireless Network Security*. Boston, MA, USA: Springer, 2007, pp. 183–210.
[505] V. Richariya, U. P. Singh, and R. Mishra, "Distributed approach of intrusion detection system: Survey," *Int. J. Adv. Comput. Res.*, vol. 2, no. 4, p. 358, 2012.
[506] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in mobile ad hoc networks," in *Wireless Network Security*. Boston, MA, USA: Springer, 2007, pp. 103–135.
[507] A. A. Korba, M. Nafaa, and G. Salim, "Survey of routing attacks and countermeasures in mobile ad hoc networks," in *Proc. 15th Int. Conf. Comput. Modeling Simulation (UKSim)*, Apr. 2013, pp. 693–698.
[508] R. Buschkes, D. Kesdogan, and P. Reichl, "How to increase security in mobile networks by anomaly detection," in *Proc. 14th Annu. Comput. Secur. Appl. Conf.*, 1998, pp. 3–12.
[509] Y. Moreau, H. Verrelst, and J. Vandewalle, "Detection of mobile phone fraud using supervised neural networks: A first prototype," in *Proc. Int. Conf. Artif. Neural Netw.* Berlin, Germany: Springer, 1997, pp. 1065–1070.
[510] J. Hollmén, *User Profiling and Classification For Fraud Detection in Mobile Communications Networks*. Espoo, Finland: Helsinki Univ. of Technology, 2000.
[511] P. Burge and J. Shawe-Taylor, "An unsupervised neural network approach to profiling the behavior of mobile phone users for use in fraud detection," *J. Parallel Distrib. Comput.*, vol. 61, no. 7, pp. 915–925, Jul. 2001.
[512] A. Boukerche and M. S. M. A. Notare, "Behavior-based intrusion detection in mobile phone systems," *J. Parallel Distrib. Comput.*, vol. 62, no. 9, pp. 1476–1490, Sep. 2002.
[513] L. Liu, G. Yan, X. Zhang, and S. Chen, "Virusmeter: Preventing your cellphone from spies," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Berlin, Germany: Springer, 2009, pp. 244–264.

[514] P. Kumpulainen and K. Hätönen, "Anomaly detection algorithm test bench for mobile network management," Tampere Univ. Technol., Tampere, Finland, 2008. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.618.5065&rep=rep1&type=pdf

[515] D. Damopoulos, S. A. Menesidou, G. Kambourakis, M. Papadaki, N. Clarke, and S. Gritzalis, "Evaluation of anomaly-based IDS for mobile devices using machine learning classifiers," *Secur. Commun. Netw.*, vol. 5, no. 1, pp. 3–14, Jan. 2012.

[516] A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina, and S. Shamshirband, "A study of machine learning classifiers for anomaly-based mobile botnet detection," *Malaysian J. Comput. Sci.*, vol. 26, no. 4, pp. 251–265, Dec. 2013.

[517] E. A. Shams and A. Rizaner, "A novel support vector machine based intrusion detection system for mobile ad hoc networks," *Wireless Netw.*, vol. 24, no. 5, pp. 1821–1829, Jul. 2018.

[518] K. K. Nguyen, D. T. Hoang, D. Niyato, P. Wang, D. Nguyen, and E. Dutkiewicz, "Cyberattack detection in mobile cloud computing: A deep learning approach," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[519] *Bitdefender*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.bitdefender.com.au/

[520] *Trend Micro*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.trendmicro.com/en_au/forHome.html

[521] *BullGuard*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.bullguard.com/

[522] *Sophos*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.sophos.com/en-us.aspx

[523] *Trustlook*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.trustlook.com/

[524] *Psafe Electronic Security Systems*. Accessed: Feb. 16, 2020. [Online]. Available: http://www.esuppliersindia.com/psafe-electronic-security-systems/intrusion-alarm-system-pr3537476-sFP-swf.html

[525] P. Pathak, "Cybercrime: A global threat to cybercommunity," *Int. J. Comput. Sci. Eng. Technol.*, vol. 7, no. 3, pp. 46–49, 2016.

[526] H. Guo, H. K. Cheng, and K. Kelley, "Impact of network structure on malware propagation: A growth curve perspective," *J. Manage. Inf. Syst.*, vol. 33, no. 1, pp. 296–325, Jan. 2016.

[527] M. R. Reports. *McAfee Labs Threats Report August 2019*. Accessed: Nov. 15, 2020. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-aug-2019.pdf

[528] *The 2012 Norton Cybercrime Report*, Symantec, Mountain View, CA, USA, 2012.

[529] *Malware Report: The Economic Impact of Viruses, Spyware, Adware, Botnets, and Other Malicious Code*, Comput. Econ., Irvine, CA, USA, 2007.

[530] R. Stone, "A call to cyber arms," Amer. Assoc. Advancement Sci., Washington, DC, USA, Tech. Rep., 2013. [Online]. Available: https://science.sciencemag.org/content/339/6123/1026/tab-pdf, doi: 10.1126/science.339.6123.1026.

[531] R. Richardson, "CSI computer crime and security survey," *Comput. Secur. Inst.*, vol. 1, pp. 1–30, Nov. 2008.

[532] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning," *J. Comput. Secur.*, vol. 19, no. 4, pp. 639–668, Jun. 2011.

[533] K. Allix, T. F. Bissyandé, Q. Jérome, J. Klein, R. State, and Y. L. Traon, "Large-scale machine learning-based malware detection: Confronting the '10-fold cross validation' scheme with reality," in *Proc. 4th ACM Conf. Data Appl. Secur. Privacy (CODASPY)*, 2014, pp. 163–166.

[534] H. V. Nath and B. M. Mehtre, "Static malware analysis using machine learning methods," in *Proc. Int. Conf. Secur. Comput. Netw. Distrib. Syst.* Berlin, Germany: Springer, 2014, pp. 440–450.

[535] X. Sun, J. Dai, P. Liu, A. Singhal, and J. Yen, "Using Bayesian networks for probabilistic identification of zero-day attack paths," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2506–2521, Oct. 2018.

[536] Y. Afek, A. Bremler-Barr, and S. L. Feibish, "Zero-day signature extraction for high-volume attacks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 691–706, Apr. 2019.

[537] A. Govindaraju, "Exhaustive statistical analysis for detection of metamorphic malware," Master's Projects, 2010, p. 66. [Online]. Available: https://scholarworks.sjsu.edu/etd_projects/66/, doi: 10.31979/etd.ucv9-qd8t.

[538] B. Anderson, D. Quist, J. Neil, C. Storlie, and T. Lane, "Graph-based malware detection using dynamic analysis," *J. Comput. Virol.*, vol. 7, no. 4, pp. 247–258, Nov. 2011.

[539] T. Isohara, K. Takemori, and A. Kubota, "Kernel-based behavior analysis for Android malware detection," in *Proc. 7th Int. Conf. Comput. Intell. Secur.*, Dec. 2011, pp. 1011–1015.

[540] T. Kavzoglu and I. Colkesen, "The effects of training set size for performance of support vector machines and decision trees," in *Proc. 10th Int. Symp. Spatial Accuracy Assessment Natural Resour. Environ. Sci.*, 2012, p. 1013.

[541] Y. Chen, A. Narayanan, S. Pang, and B. Tao, "Multiple sequence alignment and artificial neural networks for malicious software detection," in *Proc. 8th Int. Conf. Natural Comput.*, May 2012, pp. 261–265.

[542] A. Shabtai, R. Moskovitch, C. Feher, S. Dolev, and Y. Elovici, "Detecting unknown malicious code by applying classification techniques on OpCode patterns," *Secur. Informat.*, vol. 1, no. 1, p. 1, Dec. 2012.

[543] A. Mohaisen and O. Alrawi, "Unveiling zeus: Automated classification of malware samples," in *Proc. 22nd Int. Conf. World Wide Web (WWW Companion)*, 2013, pp. 829–832.

[544] I. Santos, F. Brezo, X. Ugarte-Pedrero, and P. G. Bringas, "Opcode sequences as representation of executables for data-mining-based unknown malware detection," *Inf. Sci.*, vol. 231, pp. 64–82, May 2013.

[545] R. Islam, R. Tian, L. M. Batten, and S. Versteeg, "Classification of malware based on integrated static and dynamic features," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 646–656, Mar. 2013.

[546] C. Liangboonprakong and O. Sornil, "Classification of malware families based on N-grams sequential pattern features," in *Proc. IEEE 8th Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2013, pp. 777–782.

[547] A. H. Bhat, S. Patra, and D. Jena, "Machine learning approach for intrusion detection on cloud virtual machines," *Int. J. Appl. Innov. Eng. Manage.*, vol. 2, no. 6, pp. 56–66, 2013.

[548] Z. Salehi, A. Sami, and M. Ghiasi, "Using feature generation from API calls for malware detection," *Comput. Fraud Secur.*, vol. 2014, no. 9, pp. 9–18, Sep. 2014.

[549] S. Y. Yerima, S. Sezer, and I. Muttik, "Android malware detection using parallel machine learning classifiers," in *Proc. 8th Int. Conf. Next Gener. Mobile Apps, Services Technol.*, Sep. 2014, pp. 37–42.

[550] P. V. Shijo and A. Salim, "Integrated static and dynamic analysis for malware detection," *Procedia Comput. Sci.*, vol. 46, pp. 804–811, Jan. 2015.

[551] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *Int. J. Secur. Appl.*, vol. 9, no. 5, pp. 205–216, May 2015.

[552] B. M. Khammas, A. Monemi, J. S. Bassi, I. Ismail, S. M. Nor, and M. N. Marsono, "Feature selection and machine learning classification for malware detection," *J. Teknologi*, vol. 77, no. 1, pp. 1–9, Oct. 2015.

[553] .-I. Fan, H.-W. Hsiao, C.-H. Chou, and Y.-F. Tseng, "Malware detection systems based on API log data mining," in *Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf.*, vol. 3, Jul. 2015, pp. 255–260.

[554] Q. Jamil and M. A. Shah, "Analysis of machine learning solutions to detect malware in android," in *Proc. 6th Int. Conf. Innov. Comput. Technol. (INTECH)*, Aug. 2016, pp. 226–232.

[555] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: Android malware characterization and detection using deep learning," *Tsinghua Sci. Technol.*, vol. 21, no. 1, pp. 114–123, Feb. 2016.

[556] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "DL4MD: A deep learning framework for intelligent malware detection," in *Proc. Int. Conf. Data Mining, Steering Committee World Congr. Comput. Sci. (DMIN)*, 2016, p. 61.

[557] H. S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection," *J. Comput. Virol. Hacking Techn.*, vol. 12, no. 2, pp. 59–67, May 2016.

[558] A. Karim, R. Salleh, and M. K. Khan, "SMARTbot: A behavioral analysis framework augmented with machine learning to identify mobile botnet applications," *PLoS ONE*, vol. 11, no. 3, Mar. 2016, Art. no. e0150077.

[559] Y. Cheng, W. Fan, W. Huang, and J. An, "A shellcode detection method based on full native API sequence and support vector machine," in *Proc. IOP Conf., Mater. Sci. Eng.*, vol. 242, no. 1. Bristol, U.K.: IOP Publishing, 2017, Art. no. 012124.

[560] D. Moon, H. Im, I. Kim, and J. H. Park, "DTB-IDS: An intrusion detection system based on decision tree using behavior analysis for preventing APT attacks," *J. Supercomput.*, vol. 73, no. 7, pp. 2881–2895, Jul. 2017.

[561] R. Mosli, R. Li, B. Yuan, and Y. Pan, "A behavior-based approach for malware detection," in *Proc. IFIP Int. Conf. Digit. Forensics*. Cham, Switzerland: Springer, 2017, pp. 187–201.

[562] Z. Cheng, X. Chen, Y. Zhang, S. Li, and Y. Sang, "Detecting information theft based on mobile network flows for Android users," in *Proc. Int. Conf. Netw., Archit., Storage (NAS)*, Aug. 2017, pp. 1–10.

[563] R. Nix and J. Zhang, "Classification of Android apps and malware using deep neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1871–1878.
[564] S. Hou, A. Saas, L. Chen, Y. Ye, and T. Bourlai, "Deep neural networks for automatic Android malware detection," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Jul. 2017, pp. 803–810.
[565] H.-J. Zhu, T.-H. Jiang, B. Ma, Z.-H. You, W.-L. Shi, and L. Cheng, "HEMD: A highly efficient random forest-based malware detection framework for android," *Neural Comput. Appl.*, vol. 30, no. 11, pp. 3353–3361, Dec. 2018.
[566] P. Feng, J. Ma, C. Sun, X. Xu, and Y. Ma, "A novel dynamic Android malware detection system with ensemble learning," *IEEE Access*, vol. 6, pp. 30996–31011, 2018.
[567] P. Yan and Z. J. S. Q. J. Yan, "A survey on dynamic mobile malware detection," *Softw. Qual. J.*, vol. 26, no. 3, pp. 891–919, 2018.
[568] T. D. Phan and N. Zincir-Heywood, "User identification via neural network based language models," *Int. J. Netw. Manage.*, vol. 29, no. 3, p. e2049, May 2019.
[569] S. Arshad, M. A. Shah, A. Wahid, A. Mehmood, H. Song, and H. Yu, "SAMADroid: A novel 3-Level hybrid malware detection model for Android operating system," *IEEE Access*, vol. 6, pp. 4321–4339, 2018.
[570] E. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, "MalDozer: Automatic framework for Android malware detection using deep learning," *Digit. Invest.*, vol. 24, pp. S48–S59, Mar. 2018.
[571] C. Hasegawa and H. Iyatomi, "One-dimensional convolutional neural networks for Android malware detection," in *Proc. IEEE 14th Int. Colloq. Signal Process. Appl. (CSPA)*, Mar. 2018, pp. 99–102.
[572] H. Alshahrani, H. Mansourt, S. Thorn, A. Alshehri, A. Alzahrani, and H. Fu, "DDefender: Android application threat detection using static and dynamic analysis," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–6.
[573] S. Naz and D. K. Singh, "Review of machine learning methods for windows malware detection," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–6.
[574] K. Sethi, R. Kumar, L. Sethi, P. Bera, and P. K. Patra, "A novel machine learning based malware detection and classification framework," in *Proc. Int. Conf. Cyber Secur. Protection Digit. Services (Cyber Secur.)*, Jun. 2019, pp. 1–4.
[575] H. Sayadi, H. M. Makrani, S. M. Pudukotai Dinakarrao, T. Mohsenin, A. Sasan, S. Rafatirad, and H. Homayoun, "2SMaRT: A two-stage machine learning-based approach for run-time specialized hardware-assisted malware detection," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 728–733.
[576] A. Mehtab, W. B. Shahid, T. Yaqoob, M. F. Amjad, H. Abbas, H. Afzal, and M. N. Saqib, "AdDroid: Rule-based machine learning framework for Android malware analysis," *Mobile Netw. Appl.*, vol. 25, no. 1, pp. 180–192, Feb. 2020.
[577] F. Mercaldo and A. Santone, "Deep learning for image-based mobile malware detection," *J. Comput. Virol. Hacking Techn.*, vol. 16, pp. 1–15, Jan. 2020.
[578] Z. Ma, H. Ge, Z. Wang, Y. Liu, and X. Liu, "Droidetec: Android malware detection and malicious code localization through deep learning," 2020, *arXiv:2002.03594*. [Online]. Available: http://arxiv.org/abs/2002.03594
[579] H. Naeem, F. Ullah, M. R. Naeem, S. Khalid, D. Vasan, S. Jabbar, and S. Saeed, "Malware detection in industrial Internet of Things based on hybrid image visualization and deep learning model," *Ad Hoc Netw.*, vol. 105, Aug. 2020, Art. no. 102154.
[580] X. Pei, L. Yu, and S. Tian, "AMalNet: A deep learning framework based on graph convolutional networks for malware detection," *Comput. Secur.*, vol. 93, Jun. 2020, Art. no. 101792.
[581] A. Sharma and S. K. Sahay, "Evolution and detection of polymorphic and metamorphic malwares: A survey," 2014, *arXiv:1406.7061*. [Online]. Available: http://arxiv.org/abs/1406.7061
[582] O. Henchiri and N. Japkowicz, "A feature selection and evaluation scheme for computer virus detection," in *Proc. 6th Int. Conf. Data Mining (ICDM)*, Dec. 2006, pp. 891–895.
[583] M. Siddiqui, M. C. Wang, and J. Lee, "Detecting Internet worms using data mining techniques," *J. Systemics, Cybern. Inform.*, vol. 6, no. 6, pp. 48–53, 2009.
[584] S. B. Mehdi, A. K. Tanwani, and M. Farooq, "IMAD: In-execution malware analysis and detection," in *Proc. 11th Annu. Conf. Genetic Evol. Comput. (GECCO)*, 2009, pp. 1553–1560.
[585] J. Z. Kolter and M. A. Maloof, "Learning to detect malicious executables in the wild," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2004, pp. 470–478.
[586] S. M. Tabish, M. Z. Shafiq, and M. Farooq, "Malware detection using statistical analysis of byte-level file content," in *Proc. ACM SIGKDD Workshop CyberSecurity Intell. Informat. (CSI-KDD)*, 2009, pp. 23–31.
[587] B. Mehdi, F. Ahmed, S. A. Khayyam, and M. Farooq, "Towards a theory of generalizing system call representation for in-execution malware detection," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.
[588] I. Santos, J. Nieves, and P. G. Bringas, "Semi-supervised learning for unknown malware detection," in *Proc. Int. Symp. Distrib. Comput. Artif. Intell.* Springer, 2011, pp. 415–422.
[589] O. E. David and N. S. Netanyahu, "DeepSign: Deep learning for automatic malware signature generation and classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.
[590] M. Yeo, Y. Koo, Y. Yoon, T. Hwang, J. Ryu, J. Song, and C. Park, "Flow-based malware detection using convolutional neural network," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 910–913.
[591] A. Sharma and S. K. Sahay, "An effective approach for classification of advanced malware with high accuracy," 2016, *arXiv:1606.06897*. [Online]. Available: http://arxiv.org/abs/1606.06897
[592] A. H. Lashkari, A. F. A. Kadir, H. Gonzalez, K. F. Mbah, and A. A. Ghorbani, "Towards a network-based framework for Android malware detection and characterization," in *Proc. 15th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2017, pp. 233–23309.
[593] P. Parrend, J. Navarro, F. Guigou, A. Deruyver, and P. Collet, "Foundations and applications of artificial intelligence for zero-day and multi-step attack detection," *EURASIP J. Inf. Secur.*, vol. 2018, no. 1, p. 4, Dec. 2018.
[594] D. Arivudainambi, V. K. Ka, and P. Visu, "Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance," *Comput. Commun.*, vol. 147, pp. 50–57, Nov. 2019.
[595] Y. Ding, S. Chen, and J. Xu, "Application of deep belief networks for opcode based malware detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 3901–3908.
[596] M. Nadeem, O. Marshall, S. Singh, X. Fang, and X. Yuan, "Semi-supervised deep neural network for network intrusion detection," in *Proc. KSU Cybersecur. Educ., Res. Pract.*, vol. 2, 2016. [Online]. Available: https://digitalcommons.kennesaw.edu/ccerp/2016/Practice/2
[597] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. K. Nicholas, "Malware detection by eating a whole exe," in *Proc. Workshops 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–13.
[598] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial examples for malware detection," in *Proc. Eur. Symp. Res. Comput. Secur.* Cham, Switzerland: Springer, 2017, pp. 62–79.
[599] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, and F. Roli, "Adversarial malware binaries: Evading deep learning for malware detection in executables," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 533–537.
[600] W. Yang, D. Kong, T. Xie, and C. A. Gunter, "Malware detection in adversarial settings: Exploiting feature evolutions and confusions in Android apps," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, Dec. 2017, pp. 288–302.
[601] A. Al-Dujaili, A. Huang, E. Hemberg, and U.-M. O'Reilly, "Adversarial deep learning for robust detection of binary encoded malware," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 76–82.
[602] T. S. John and T. Thomas, "Adversarial attacks and defenses in malware detection classifiers," in *Handbook of Research on Cloud Computing and Big Data Applications in IoT*. Hershey, PA, USA: IGI Global, 2019, pp. 127–150.
[603] R. Podschwadt and H. Takabi, "Effectiveness of adversarial examples and defenses for malware classification," 2019, *arXiv:1909.04778*. [Online]. Available: http://arxiv.org/abs/1909.04778
[604] A. Y. Huang, *Towards Robust Malware Detection*. Cambridge, MA, USA: MIT, 2018.
[605] *Avast Internet Security*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.avast.com/en-au/internet-security
[606] *10 BEST Free Malware Removal Software Of 2020*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.softwaretestinghelp.com/best-malware-removal/
[607] *Malwarebytes*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.malwarebytes.com/
[608] *Norton Power Eraser*. Accessed: Feb. 16, 2020. [Online]. Available: https://us.norton.com/support/tools/npe.html
[609] *AVG*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.avg.com/en-ww/homepage#pc

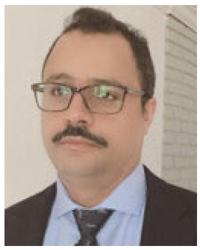
[610] *Bitdefender Antivirus*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.bitdefender.com/
[611] Q. Su, J. Tian, X. Chen, and X. Yang, "A fingerprint authentication system based on mobile phone," in *Proc. Int. Conf. Audio Video-Based Biometric Person Authentication*. Berlin, Germany: Springer, 2005, pp. 151–159.
[612] A. Arora, S. Garg, and S. K. Peddoju, "Malware detection using network traffic analysis in Android based mobile devices," in *Proc. 8th Int. Conf. Next Gener. Mobile Apps, Services Technol.*, Sep. 2014, pp. 66–71.
[613] L. Chen, "Deep transfer learning for static malware classification," 2018, *arXiv:1812.07606*. [Online]. Available: http://arxiv.org/abs/1812.07606
[614] A. Jadhav, D. Vidyarthi, and M. Hemavathy, "Evolution of evasive malwares: A survey," in *Proc. Int. Conf. Comput. Techn. Inf. Commun. Technol. (ICCTICT)*, Mar. 2016, pp. 641–646.
[615] F. Martinelli, F. Mercaldo, A. Saracino, and C. A. Visaggio, "I find your behavior disturbing: Static and dynamic app behavioral analysis for detection of Android malware," in *Proc. 14th Annu. Conf. Privacy, Secur. Trust (PST)*, 2016, pp. 129–136.c.
[616] A. De Paola, S. Gaglio, G. L. Re, and M. Morana, "A hybrid system for malware detection on big data," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 45–50.
[617] K. Bartos, M. Sofka, and V. Franc, "Optimized invariant representation of network traffic for detecting unseen malware variants," in *Proc. 25th USENIX Secur. Symp. (USENIX Security)*, 2016, pp. 807–822.
[618] P. Wang and Y.-S. Wang, "Malware behavioural detection and vaccine development by using a support vector model classifier," *J. Comput. Syst. Sci.*, vol. 81, no. 6, pp. 1012–1026, Sep. 2015.
[619] H.-S. Ham and M.-J. Choi, "Analysis of Android malware detection performance using machine learning classifiers," in *Proc. Int. Conf. ICT Converg. (ICTC)*, Oct. 2013, pp. 490–495.
[620] N. Peiravian and X. Zhu, "Machine learning for Android malware detection using permission and api calls," in *Proc. IEEE 25th Int. Conf. Tools Artif. Intell.*, Nov. 2013, pp. 300–305.c.
[621] A. Bhattacharya and R. T. Goswami, "DMDAM: Data mining based detection of Android malware," in *Proc. 1st Int. Conf. Intell. Comput. Commun.* Singapore: Springer, 2017, pp. 187–194.
[622] D. O. Sahin, O. E. Kural, S. Akleylek, and E. Kilic, "New results on permission based static analysis for Android malware," in *Proc. 6th Int. Symp. Digit. Forensic Secur. (ISDFS)*, Mar. 2018, pp. 1–4.
[623] G. Tam and A. Hunter, "Machine learning to identify Android malware," in *Proc. 9th IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Nov. 2018, pp. 1–5.
[624] L. Vaishanav, S. Chauhan, H. Vaishanav, M. S. Sankhla, and R. Kumar, "Behavioural analysis of Android malware using machine learning," *Int. J. Eng. Comput. Sci.*, vol. 6, no. 5, pp. 1–12, 2017.
[625] C. S. Gates, N. Li, H. Peng, B. Sarma, Y. Qi, R. Potharaju, C. Nita-Rotaru, and I. Molloy, "Generating summary risk scores for mobile applications," *IEEE Trans. dependable secure Comput.*, vol. 11, no. 3, pp. 238–251, Jan. 2014.
[626] Y. Lu, P. Zulie, L. Jingju, and S. Yi, "Android malware detection technology based on improved Bayesian classification," in *Proc. 3rd Int. Conf. Instrum., Meas., Comput., Commun. Control*, Sep. 2013, pp. 1338–1341.
[627] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, and P. G. Bringas, "On the automatic categorisation of Android applications," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2012, pp. 149–153.
[628] M. Odusami, O. Abayomi-Alli, S. Misra, O. Shobayo, R. Damasevicius, and R. Maskeliunas, "Android malware detection: A survey," in *Proc. Int. Conf. Appl. Informat.* Cham, Switzerland: Springer, 2018, pp. 255–266.
[629] O. Shobayo, R. Damasevicius, and R. Maskeliunas, "Android malware detection: A survey," in *Proc. 1st Int. Conf. Appl. Inform. (ICAI)*, vol. 942. Bogotá, Colombia: Springer, Nov. 2018, p. 255.
[630] R. Zachariah, K. Akash, M. S. Yousef, and A. M. Chacko, "Android malware detection a survey," in *Proc. IEEE Int. Conf. Circuits Syst. (ICCS)*, Dec. 2017, pp. 238–244.
[631] L. Wen and H. Yu, "An Android malware detection system based on machine learning," in *Proc. AIP Conf.*, vol. 1864, no. 1. Melville, NY, USA: AIP Publishing, 2017, Art. no. 020136.
[632] S. Gunalakshmii and P. Ezhumalai, "Mobile keylogger detection using machine learning technique," in *Proc. IEEE Int. Conf. Comput. Commun. Syst. (ICCCS)*, Feb. 2014, pp. 51–56.
[633] J. Sahs and L. Khan, "A machine learning approach to Android malware detection," in *Proc. Eur. Intell. Secur. Informat. Conf.*, Aug. 2012, pp. 141–147.
[634] W. Li, J. Ge, and G. Dai, "Detecting malware for Android platform: An SVM-based approach," in *Proc. IEEE 2nd Int. Conf. Cyber Secur. Cloud Comput.*, Nov. 2015, pp. 464–469.
[635] S. Sheen, R. Anitha, and V. Natarajan, "Android based malware detection using a multifeature collaborative decision fusion approach," *Neurocomputing*, vol. 151, pp. 905–912, Mar. 2015.
[636] H.-S. Ham, H.-H. Kim, M.-S. Kim, and M.-J. Choi, "Linear SVM-based Android malware detection," in *Frontier and Innovation in Future Computing and Communications*. Dordrecht, The Netherlands: Springer, 2014, pp. 575–585.
[637] A. Narayanan, L. Chen, and C. K. Chan, "AdDetect: Automated detection of Android ad libraries using semantic analysis," in *Proc. IEEE 9th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process. (ISSNIP)*, Apr. 2014, pp. 1–6.
[638] M. Spreitzenbarth, T. Schreck, F. Echtler, D. Arp, and J. Hoffmann, "Mobile-sandbox: Combining static and dynamic analysis with machine-learning techniques," *Int. J. Inf. Secur.*, vol. 14, no. 2, pp. 141–153, Apr. 2015.
[639] D. Zhu, H. Jin, Y. Yang, D. Wu, and W. Chen, "DeepFlow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 438–443.
[640] A. A. A. Samra and O. A. Ghanem, "Analysis of clustering technique in Android malware detection," in *Proc. 7th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, Jul. 2013, pp. 729–733.
[641] *Kaspersky Mobile Antivirus*. Accessed: Feb. 16, 2020. [Online]. Available: https://usa.kaspersky.com/android-security
[642] *Norton Security and Antivirus*. Accessed: Feb. 16, 2020. [Online]. Available: https://my.norton.com/mobile/home
[643] *Avira Antivirus Security*. Accessed: Feb. 16, 2020. [Online]. Available: https://www.avira.com/
[644] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.
[645] S. Ghosh, P. Lincoln, A. Tiwari, and X. Zhu, "Trusted machine learning: Model repair and data repair for probabilistic models," in *Proc. Workshops 31st AAAI Conf. Artif. Intell.*, 2017.
[646] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.
[647] C. Dai, H.-S. Lim, E. Bertino, and Y.-S. Moon, "Assessing the trustworthiness of location data based on provenance," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, 2009, pp. 276–285
[648] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain, "Managing uncertainty in moving objects databases," *ACM Trans. Database Syst.*, vol. 29, no. 3, pp. 463–507, Sep. 2004.
[649] M. Zhu and Z. Jin, "A trust measurement mechanism for service agents," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, Sep. 2009, pp. 375–382.
[650] Q. Su, C.-R. Huang, and H. K.-Y. Chen, "Evidentiality for text trustworthiness detection," in *Proc. Workshop NLP Linguistics, Finding Common Ground, Assoc. Comput. Linguistics*, 2010, pp. 10–17.
[651] H. Tao and Y. Chen, "A metric model for trustworthiness of softwares," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, Sep. 2009, pp. 69–72.
[652] J. L. Berral, Ì. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, and J. Torres, "Towards energy-aware scheduling in data centers using machine learning," in *Proc. 1st Int. Conf. Energy-Efficient Comput. Netw. (E-Energy)*, 2010, pp. 215–224.
[653] X. Wang, Y. Gao, J. Lin, H. Rangwala, and R. Mittu, "A machine learning approach to false alarm detection for critical arrhythmia alarms," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 202–207.
[654] L. M. Eerikainen, J. Vanschoren, M. J. Rooijakkers, R. Vullings, and R. M. Aarts, "Decreasing the false alarm rate of arrhythmias in intensive care using a machine learning approach," in *Proc. Comput. Cardiol. Conf. (CinC)*, Sep. 2015, pp. 293–296.
[655] S. Huang, E.-H. Liu, Z.-W. Hui, S.-Q. Tang, and S.-J. Zhang, "Challenges of testing machine learning applications," *Int. J. Performability Eng.*, vol. 14, no. 6, pp. 1–8, 2018.
[656] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 50–56.
[657] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi, "DeepCloak: Masking deep neural network models for robustness against adversarial samples," 2017, *arXiv:1702.06763*. [Online]. Available: http://arxiv.org/abs/1702.06763

[658] I. Goodfellow, P. McDaniel, and N. Papernot, "Making machine learning robust against adversarial inputs," *Commun. ACM*, vol. 61, no. 7, pp. 56–66, Jun. 2018.
[659] T. Thomas, A. P. Vijayaraghavan, and S. Emmanuel, "Adversarial machine learning in cybersecurity," in *Machine Learning Approaches in Cyber Security Analytics*. Singapore: Springer, 2020, pp. 185–200.
[660] P. Dasgupta and J. B. Collins, "A survey of game theoretic approaches for adversarial machine learning in cybersecurity tasks," 2019, *arXiv:1912.02258*. [Online]. Available: http://arxiv.org/abs/1912.02258
[661] O. Ibitoye, R. Abou-Khamis, A. Matrawy, and M. O. Shafiq, "The threat of adversarial attacks on machine learning in network security—A survey," 2019, *arXiv:1911.02621*. [Online]. Available: http://arxiv.org/abs/1911.02621
[662] F. Zhang, P. P. K. Chan, B. Biggio, D. S. Yeung, and F. Roli, "Adversarial feature selection against evasion attacks," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 766–777, Mar. 2016.
[663] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*. [Online]. Available: http://arxiv.org/abs/1412.6572
[664] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.
[665] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," 2017, *arXiv:1704.01155*. [Online]. Available: http://arxiv.org/abs/1704.01155
[666] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. Conf. Comput. Commun. Secur. (ACM SIGSAC)*, Oct. 2017, pp. 135–147.
[667] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 305–316.
[668] A. A. Cárdenas, P. K. Manadhata, and S. P. Rajan, "Big data analytics for security," *IEEE Security Privacy*, vol. 11, no. 6, pp. 74–76, Nov./Dec. 2013.

KAMRAN SHAUKAT received the M.Sc. degree in computer science from Mohammad Ali Jinnah University, Pakistan, and obtained a Gold Medal. He is currently pursuing the Ph.D. degree with The University of Newcastle, Australia. He is the author of many papers in the area of machine learning, databases, and cyber security. He has served the University of the Punjab, Pakistan, for seven years as a Lecturer. He has served as a Reviewer for many journals, including IEEE Access. He has attended several international conferences in the USA, U.K., Thailand, Turkey, and Pakistan.

SUHUAI LUO received the bachelor's and master's degrees from the Nanjing University of Posts and Telecommunications, and the Ph.D. degree from The University of Sydney, all in electrical engineering. He is currently an Associate Professor in information technology with The University of Newcastle. His main research interests include image processing, computer vision, machine learning, cybersecurity, and media data mining. His diverse research focus has led him to conduct studies in areas ranging from medical imaging for computer-aided diagnoses, to computer vision for intelligent driving systems, and machine learning for enhancing cybersecurity. He has lectured and developed curricula for courses in computer science, electrical engineering, and information engineering.

VIJAY VARADHARAJAN (Senior Member, IEEE) is currently the Global Innovation Chair Professor with The University of Newcastle, Australia, and the Director of the Advanced Cyber Security Research Centre. He has published over 380 papers in international journals and conferences, ten books on information technology, security, networks, and distributed systems, and has held three patents. He has been/is on the Editorial Board of several journals, including the *ACM Transactions on Information and System Security*, the IEEE Transactions on Dependable and Secure Computing, the IEEE Transactions on Information Forensics and Security, and the IEEE Transactions on Cloud Computing.



IBRAHIM A. HAMEED (Senior Member, IEEE) received the Ph.D. degree in industrial systems and information engineering from Korea University, Seoul, South Korea, and the Ph.D. degree in mechanical engineering from Aarhus University, Aarhus, Denmark. He is currently a Professor with the Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology (NTNU), Norway, where he is the Deputy Head of Research and Innovation within the same department. His current research interests include artificial intelligence, machine learning, optimization, and robotics. He is an elected chair of the IEEE Computational Intelligence Society (CIS) Norway Section.

MIN XU (Member, IEEE) received the bachelor's degree in engineering from the University of Science and Technology of China, the master's degree in science (computing) from the National University of Singapore, and the Ph.D. degree in IT from The University of Newcastle, Australia. She is currently an Associate Professor with the School of Electrical and Data Engineering, University of Technology Sydney, Australia. Her expertise is in multimedia data (video, audio, and text) analytics and computer vision. She has proposed several innovative methods for 1) multimedia affective/semantic content analysis, 2) multi-modality information analysis and fusion. Recently, she is focusing on applying machine learning algorithms for multimedia applications, including affective computing, image caption, and action recognition. She has published over 100 research papers in high-quality international journals and conferences.