Journal of Cybersecurity and Privacy

MDPI

Review

Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review

Mostofa Ahsan [1,*], Kendall E. Nygard [1], Rahul Gomes [2,*], Md Minhaz Chowdhury [3], Nafiz Rifat [1] and Jayden F Connolly [2]

[1] Department of Computer Science, North Dakota State University, Fargo, ND 58102, USA; kendall.nygard@ndsu.edu (K.E.N.); nafiz.rifat@ndsu.edu (N.R.)
[2] Department of Computer Science, University of Wisconsin-Eau Claire, Eau Claire, WI 54701, USA; flippcjt5179@uwec.edu
[3] Department of Computer Science, East Stroudsburg University of Pennsylvania, East Stroudsburg, PA 18301, USA; mchowdhur1@esu.edu
* Correspondence: mostofa.ahsan@ndsu.edu (M.A.); gomesr@uwec.edu (R.G.)

Abstract: Machine learning is of rising importance in cybersecurity. The primary objective of applying machine learning in cybersecurity is to make the process of malware detection more actionable, scalable and effective than traditional approaches, which require human intervention. The cybersecurity domain involves machine learning challenges that require efficient methodical and theoretical handling. Several machine learning and statistical methods, such as deep learning, support vector machines and Bayesian classification, among others, have proven effective in mitigating cyber-attacks. The detection of hidden trends and insights from network data and building of a corresponding data-driven machine learning model to prevent these attacks is vital to design intelligent security systems. In this survey, the focus is on the machine learning techniques that have been implemented on cybersecurity data to make these systems secure. Existing cybersecurity threats and how machine learning techniques have been used to mitigate these threats have been discussed. The shortcomings of these state-of-the-art models and how attack patterns have evolved over the past decade have also been presented. Our goal is to assess how effective these machine learning techniques are against the ever-increasing threat of malware that plagues our online community.

Keywords: cybersecurity; machine learning; neural networks; classification; clustering; intrusion detection

Citation: Ahsan, M.; Nygard, K.E.; Gomes, R.; Chowdhury, M.M.; Rifat, N.; Connolly, J.F. Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review. *J. Cybersecur. Priv.* **2022**, 2, 527–555. https://doi.org/10.3390/jcp2030027

Academic Editor: Danda B. Rawat

Received: 15 June 2022
Accepted: 7 July 2022
Published: 10 July 2022

1. Introduction

With the rapidly increasing prominence of information technology in recent decades, various types of security incidents, such as unauthorized access [1], denial of service (DoS) [2], malware attacka [3], zero-day attacks [4], data breaches [5], social engineering or phishing [6], etc., have increased at an exponential rate in the last decade. In 2010, the security community documented less than 50 million distinct malware executables. In the year 2012, this reported number doubled to around 100 million. From the record according to AV-TEST statistics, the security industry detected over 900 million malicious executables in 2019, and this number is rising [7]. Cybercrime and network attacks can result in significant financial losses for businesses and people. For example, according to estimates, an average data breach costs USD 3.9 million in the United States and USD 8.19 million globally [8], and cybercrime costs the world economy USD 400 billion per year [9]. The security community estimates [10], over the next five years, that the number of records broken will nearly quadruple. As a result, to minimize further losses, businesses must create and implement a comprehensive cybersecurity strategy. The most recent socioeconomic studies show that [11] the nation's security is dependent on governments, people with access to data, applications and tools that require high security clearance.

It is also dependent on businesses that give access to their employees, who possess the capacity and knowledge to identify such cyber-threats quickly and effectively. As a result, the primary concern that must be addressed immediately is to intelligently identify various cyber occurrences, whether previously known or unseen, and safeguard critical systems from such cyber-attacks adequately.

Cybersecurity refers to technologies and techniques that protect programs, networks, computers and data from being damaged, attacked or accessed by unauthorized people [12]. Cybersecurity covers various situations, from corporate to mobile computing, and can be divided into several areas. These are: (i) network security, which focuses on preventing cyber-attackers or intruders from gaining access to a computer network; (ii) application security, which considers keeping devices and software free of risks or cyber-threats; (iii) information security, which primarily considers the security and privacy of relevant data; and (iv) operational security refers to the procedures for handling and safeguarding data assets. Traditional cybersecurity solutions include a firewall, antivirus software or an intrusion detection system in network and computer security systems. Data science is driving the transformation, where machine learning, an essential aspect of "Artificial Intelligence", can play a vital role in discovering hidden patterns from data. Data science is pioneering a new scientific paradigm, and machine learning has substantially impacted the cybersecurity landscape [13,14]. As discussed in the article [15], with the advancement of technologies pertinent to launching cyber threats, attackers are becoming more efficient, giving rise to an increasing number of connected technologies. The graph in Figure 1 depicts timestamp data in terms of a specific date, with the *x*-axis representing the matching popularity and the *y*-axis representing the corresponding popularity in the range of 0 (minimum) to 100 (maximum). It is observed that the popularity values of cybersecurity and machine learning areas were less than 30 in 2015, and they exceeded 70 in 2022, i.e., more than double in terms of increased popularity. In this study, we focus on machine learning in cybersecurity, which is closely related to these areas in terms of security, intelligent decision making and the data processing techniques to deploy in real-world applications. Overall, this research is concerned with security data, using machine learning algorithms to estimate cyber-hazards and optimize cybersecurity processes. This project is also useful for academic and industrial researchers interested in studying and developing data-driven smart cybersecurity models using machine learning approaches.

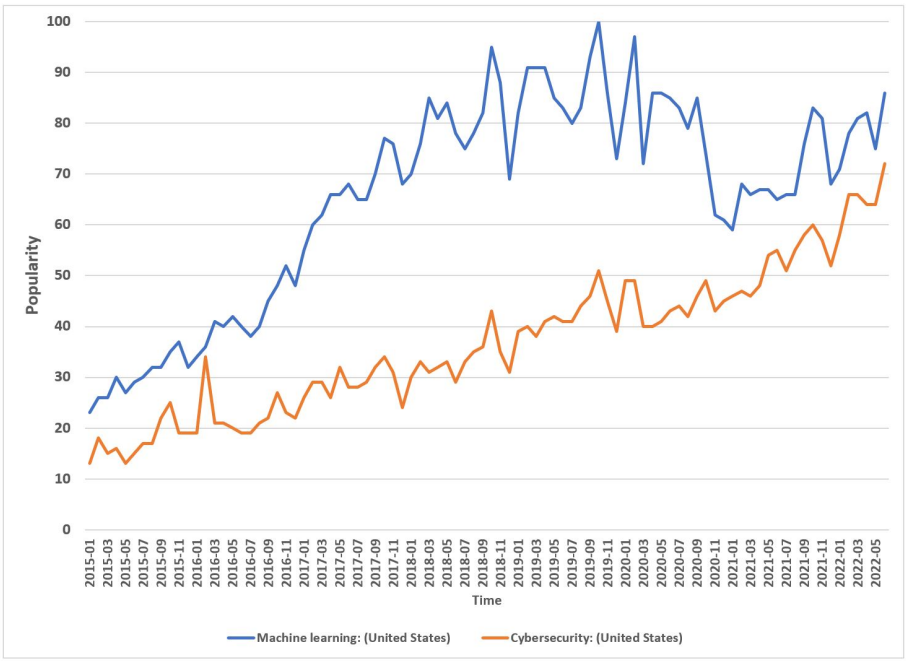

Figure 1. Google Trend for machine learning vs. data science vs. cybersecurity from 2015 to present.

Preventing cybersecurity attacks beyond a set of fundamental functional needs and knowledge about risks, threats or vulnerabilities requires analyzing cybersecurity data and building the right tools to process them successfully. Several machine learning techniques, which include but are not limited to feature reduction, regression analysis, unsupervised learning, finding associations or neural network-focused deep learning techniques, can be used to effectively extract the insights or patterns of security incidents. This is briefly discussed in the "Machine learning techniques in cybersecurity" section. These learning techniques can detect anomalies or malicious conduct and data-driven patterns of related security issues and make intelligent judgments to avert cyber-assaults.

Machine learning is a partial but significant departure from traditional well-known security solutions, including user authentication and access control, firewalls and cryptography systems, which may or may not be effective in meeting today's cyber business needs [16–18]. The critical difficulty is that domain experts and security analysts fix these manually in situations where ad hoc data management is required [19]. However, as a growing number of cybersecurity incidents in various formats are emerging over time, traditional solutions have proven ineffective in managing these cyber-hazards. As a result, a slew of new, complex attacks emerges and spreads rapidly over the network. Thus, several academics apply diverse data analytic and knowledge extraction models to create cybersecurity models, which are covered in the section "Machine learning techniques in cybersecurity", based on the efficient identification of security insights and the most recent security trends that may be more relevant. According to research, addressing the cyber problem necessitates the development of more flexible and efficient security systems that can adapt to attacks and update security policies to eradicate them on a timely basis intelligently. To do this, a huge amount of relevant cybersecurity data collected from different sources, such as network and system sources, must be analyzed. Moreover, these techniques should be implemented in a way that increases automation, with minimal to no human intervention.

The discussions of this study are listed below.

- To comprehend the applicability of data-driven decision making, a report on the existing idea of cybersecurity protection plans and associated approaches is presented first. To do this, several machine learning techniques used in cybersecurity have been discussed and numerous cybersecurity datasets, emphasizing their importance and applicability in this domain, have been presented.
- In addition, an examination of several related research challenges and future objectives in the field of cybersecurity machine learning approaches has been presented.
- Finally, the most common issues in applying machine learning algorithms on cybersecurity datasets have been explored within the scope of improvements to build a robust system.

The rest of this research is structured as follows. Section 2 describes the motivation for our research and provides an overview of cybersecurity technologies. Next, it defines and briefly covers numerous types of cybersecurity data, including cyber incident data. A brief discussion of different categories of machine learning techniques and their relationships with various cybersecurity tasks is presented in Section 3. The section also summarizes a number of the most effective machine learning algorithms for cybersecurity models in this domain. Section 4 briefly covers and emphasizes different research concerns and future directions in cybersecurity. Finally, an emphasis on some crucial elements of our findings has been presented in Section 5.

2. Background

Information and Communication Technology (ICT) infrastructure has significantly evolved over the last decade, and it is now widespread and thoroughly integrated into our modern civilization. As a result, today's security policymakers are urging ICT systems and applications to be protected from cyber-attacks [20]. Protecting an ICT infrastructure from various cyber-threats or attacks is referred to as cybersecurity [9]. Different aspects of

cybersecurity are associated with it, such as measures to protect ICT, the raw data and information it contains, as well as their processing and transmission. Other factors include the association of virtual and physical elements of the systems; the level of protection provided by these measures; and, finally, the associated field of professional endeavor [21]. According to [22], cybersecurity consists of different tools, guidelines and practices and is employed to protect software programs, computer networks and data from attack, unauthorized access or damage [22]. Research in [12] indicated that cybersecurity uses different processes and technologies that are useful to protect networks, programs, computers and data against assaults, unlawful access and destruction. In a nutshell, cybersecurity is concerned with the identification of various cyber-attacks and the implementation of appropriate defense tactics to protect the properties indicated below [22–24].

- Confidentiality is a property that prevents information from being shared with unauthorized entities, people or systems.
- Integrity is a property that protects data from being tampered with or destroyed without permission.
- Availability is used to ensure that authorized entities have timely and reliable access to information assets and systems.

2.1. Cyber-Attacks and Security Risks

There are three major security factors that are typically considered as risks: (1) attacks—who is attacking, vulnerabilities in the system; (2) the flaws or security pockets that they are attacking, and the impacts; (3) the consequences of the attack. These are all elements to consider [9]. A security breach occurs when information assets and systems' confidentiality, integrity or availability are endangered. Different forms of cybersecurity incidents might put an organization's or an individual's systems and networks at threat [9]; they can be grouped as follows.

Malware is malicious software that is designed to cause damage to a personal system, client, server or computer network [24]. Malware breaches a network by creating a vulnerable situation, such as a user clicking a dangerous link or email attachment and, hence, installing a risky software program. In most cases, the presence of such malicious software is not acknowledged by the authorized user(s) of the system. A system can become infected by malware in various ways. Examples include, but are not limited to, a victim being tricked into installing malware by opening a fake version of a legitimate file; a victim tricked in downloading malware by visiting malware-propagating websites; or a victim connecting to a malware-infected machine or device.

Malware victims can be any device containing computational logic. The victims can be end users and/or servers, their connecting network devices, process control systems, e.g., Supervisory Control and Data Acquisition systems. As with its victim types, malware can be of several types: bot executable, Trojan horses, spyware, viruses, ransomware and worms. Both the number and technologies of malware are growing fast. The most cost-effective solution is protecting the perimeter of the system by installing appropriate controls. Examples are intrusion detection/prevention systems (firewall, anti-virus software). With perimeter defense, an access control mechanism can control the access to a particular internal resource of the system. Despite these measures, there can be people violating their access rights. In such a situation, an organization policy on accountability can be implemented to punish a misdemeanor. Unfortunately, this combined effort of perimeter defense techniques with access control mechanisms and accountability may fail. Table 1 lists the recent defenses against malware attacks [24]. Typically, malware affects the network as follows:

- It blocks network key components.
- It installs additional harmful software for spying with malware itself.
- It gains access to personal data and transmits information.
- It disrupts certain components and makes the system inoperable for users.

Table 1. Defenses to protect data against malware.

Defense Technology	Categories of Defense Technologies Used Against Malware	Description of Defense Categories
Cryptography is a way to change data in such a way that only the intended receiver has the information to extract information from the changed form (encrypted data). It is the most used method to secure data.	Identity-based cryptography [25]	This is a public key generated using identification-based information, e.g., email address. The generation is processed by a trusted certifying authority. This is an active research area, to overcome the inconveniences of this cryptography against malware attacks.
	Quantum cryptography [26]	In this cryptography, for the two parties, sender and receiver, the transmission generates cryptographic keys to encrypt data, following the laws of quantum mechanics. Hence, this encryption is not hackable.
Perimeter defense/defense in depth is securing an organization's network from outside intrusion	Firewall is the prevalent perimeter defense technology that controls network traffic (input data and outgoing data). It decides whether the data will go through or not based on a set of preset rules [27]. Despite the sophistication of firewalls, they can fail when a compromised but previously trusted system sends any request, and the attacking machine uses a trusted system's identity.	1. Network-layer firewall or packet filtering works at the network layer controlling data flow but has the drawback of having static rules that are not able to block undesirable data. Hence, it cannot block malware payload. 2. Application-layer firewall controls the flow of input, output and system calls by an application. This firewall makes the tempering of internal components by malware difficult. 3. Proxy servers work as a mediator between outside connections and internal components of a system and hence can hinder the tampering of these components by malware.
	Network forensics [28] is the process of eavesdropping on the internet, Ethernet or TCP/IP to learn the attack pattern. There are numerous network forensics tools.	1. eMailTrackerPro investigates the header of an email to look for an IP address, to find the sender. 2. Web browser traffic forensic tool, SmartWhoIs, can provide all available information about an IP address. 3. WebHistorian analyzes a website's URL. 4. Index.datanalyzer analyzes the browsing history, cache and cookies. 5. In the wireless LAN interface and network interface, packet intercepts can be caught using AirPcap and WinPcap, respectively. 6. Honeypots are mock resources that trap the attacker and gather information.
	Access control [29] differentiates between users and controls resource access of the user based on the user's preset rights. It provides authentication, authorization and accountability.	1. Two broad divisions of access control, used in malware defense, are capability-based access control and the access control list-based approach. 2. Three access control models are Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-Based Access Control (RBAC).

Ransomware blocks access to the victim's data and threatens the client with its destruction unless a ransom is paid. The Trojan horse is the most dangerous malware, which appears as useful and routine software and is mostly designed to steal financial information. A drive-day attack is a common method for distributing malware. These data require any action of a user to be activated. The users simply visit a benign-looking website and their personal system is infected silently and becomes an IFrame that redirects the victim's browser into a site controlled by the attacker.

Phishing is the practice of sending fraudulent communications or social engineering, which is mostly spread through emails. The goal is to steal the victim's data, such as credit card numbers and login credentials. As part of a larger operation as an advanced and persistent threat, this assault is frequently used to achieve a foothold in government or business networks. Spear phishing is targeted to particular individuals or organizations, governments or military intelligence to acquire trade secrets, financial gains or information. Whale phishing is mostly aimed at high-profile employees such as a CFO or CEO to gain vital access to a company's sensitive data.

Man-in-the-middle (MITM), also known as eavesdropping, occurs when the intruders successfully include themselves inside a two-party transaction or communication. The most common entries for MITM attackers are:

- Unsecured public WiFi, where intruders insert themselves between a visitor's device and the network.
- If an attacker's malware successfully breaches the victim's system, they can install software to gain the victim's secure information.

Denial-of-service (DDoS) is a type of attack that involves shutting down a network or service with a high volume of traffic to deplete resources and bandwidth, resulting in the system being unable to fulfill legitimate requests. DDoS attacks are frequently designed to target high-profile businesses' web servers, such as trading platforms, media, finance and government.

SQL injection (SQLI) aims to employ malicious code to manipulate back-end database access information that was not intended for display. Intruders could carry out an SQL injection simply by submitting malicious code into a vulnerable website search box.

A zero-day exploit attack refers to the threat of an unknown security vulnerability for which a fix has not yet been provided or about which the program developers are uninformed. To detect this threat, the developers require constant awareness.

DNS tunneling uses the DNS protocol to communicate non-DNS traffic over port 53 by sending HTTP and other protocol traffic over DNS. Since using DNS tunneling is a common and legitimate process, its use for malicious reasons is very often overlooked. Attackers can disguise outbound traffic as DNS, concealing data that are shared through an internet connection.

2.2. Defense Strategies

Defense strategies are required to protect data or information, information systems and networks from cyber-attacks or intrusions. They are in charge of preventing data breaches and security incidents, as well as monitoring and responding to threats, defined as any illegal activity that damages a network or personal system [30]. In this section, a popular perimeter defense strategy, the intrusion detection system, is presented. A detailed discussion on defense strategies can be observed in Figure 2.

An intrusion detection system (IDS) is described as "a software, device or application that monitors a systems or computer network for malicious activity or policy violations" [31]. User authentication, access control, anti-virus, firewalls, cryptography systems and data encryption are all well-known security solutions that may not be successful in today's cyber sector [16–18]. An IDS analyzes security data from numerous essential locations in a network or system to remedy the issues [32,33]. Furthermore, an IDS may detect both internal and external threats. Intrusion detection systems are classified into several groups based on their intended use. There are two major domains of IDS. One focuses on the intrusion detection techniques, and another focuses on the deployment or data source to which the IDS will be applicable. The deployment opportunities can be grouped into multiple research areas [34]. Two of the possible classifications could be the host-based intrusion detection system (HIDS), which monitors and analyzes data, files and secure information on a single system, and also the network intrusion detection system (NIDS), which monitors and analyzes network connections for suspicious activity. These two IDSs are able to scale based on the file system and network size. On the other hand,

misuse detection or signature-based IDS and anomaly-based IDS are most well-known intrusion detection systems used in theory [30]. Misuse detection is very effective against known attack types, which implies that it requires specific domain knowledge of intrusive incidents [35]. One of the most popular examples of misuse detection is SNORT.

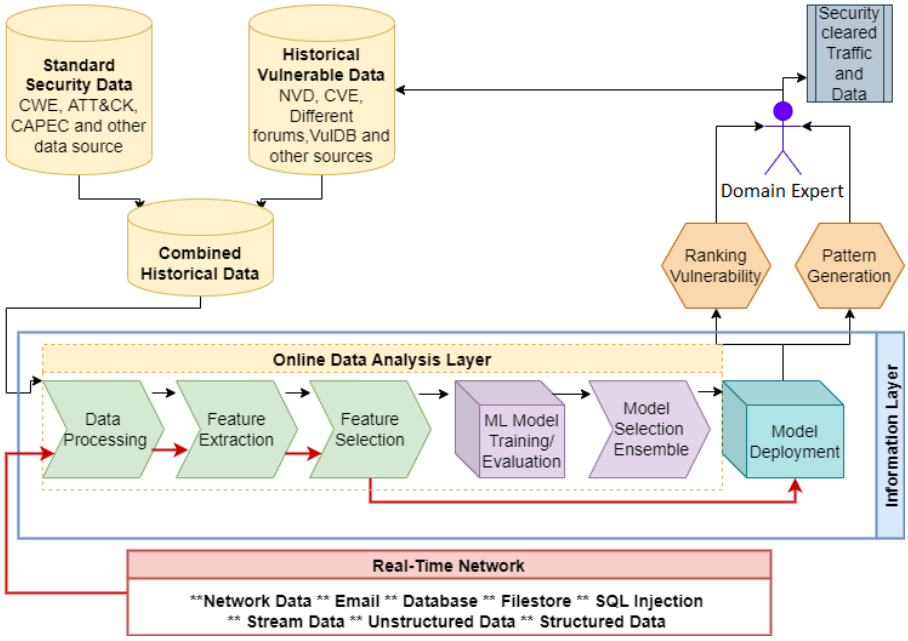

Figure 2. Flow chart of defense strategies in cybersecurity.

Signature-based detection works by signatures or fingerprint detection of network traffic [24]. This detection does not work properly for sophisticated and advanced malware that continuously evolves its patterns. This signature can be a pre-defined string, pattern or rule that correlates to a attack that has already occurred. A known pattern is defined as the detection of corresponding similar threats according to a signature-based intrusion detection system. An example of a signature-based IDS can be sequences used by mostly different types of malware, or known patterns or a byte sequence in a network traffic. Anti-virus software is used to detect these attacks, by identifying the patterns or sequences as a signature while performing a similar operation. As a result, a signature-based IDS is sometimes referred to as a knowledge-based or misuse detection system [36]. This technique can quickly process a large amount of network traffic, but it is firmly limited to rule-based or supervised detection. As a result, a signature-based system's most challenging difficulty is detecting new or unknown attacks using past knowledge.

Anomaly-based detection works by learning the pattern of normal network traffic and then flags the network traffic as anomalous if it is outside of this pattern [24]. The concept of anomaly-based detection is proposed to address the problems with signature-based IDSs that have been mentioned previously. The user activity and network traffic are first investigated in an anomaly-based intrusion detection system to discover dynamic trends, automatically create a data-driven model, profile normal behavior and detect anomalies during any departure [36]. As a result, an anomaly-based IDS is a dynamic approach that employs both supervised and unsupervised detection techniques. The capacity to detect zero-day assaults and wholly unknown threats is a significant advantage of anomaly-based IDS [37]. However, the identified anomaly or suspicious behavior sometimes leads to false alarms. Occasionally, it may identify several factors, such as policy changes or offering a new service, as an intrusion.

A hybrid detection approach [38,39] considers the anomaly-based and the signature-based techniques discussed above and can be used to identify intrusions. In a hybrid system, the signature-based detection system is used to identify known types of intrusions and an anomaly detection system is used for unknown attacks [40]. In addition to these

methods, stateful protocol analysis, which is similar to the anomaly-based method but employs established standard profiles based on agreed definitions of benign activity, can be used to detect intrusions that identify protocol state deviations [36]. A self-aware automatic response system would be the most effective of these options since it eliminates the requirement for a human link between the detection and reaction systems. There is a recent concept called Advanced Anomaly-Based Detection, which works by observing the network traffic for a certain duration [24]. Reinforcement learning (RL) is one of the advancements of Artificial Intelligence that can extend the logical reasoning of intrusion scenarios and prevent inexperienced attacks. The lack of cybersecurity attack data makes this technique extremely valuable to prevent the system against future attacks. Based on the agent or attack type, RL can be grouped into model-based and model-free approaches [41]. The application of machine learning techniques is extended in each of the branches of IDS. In the early period, it was only applicable for anomalous network data [42]. Later on, machine learning techniques were proven to be highly effective for the deployment of other IDS techniques on both the host and network domain [42]. With this observation, an adaptive and evolving model was built to deal with evolving malware signatures. Figure 3 summarizes the types of IDS based on detection and deployment.

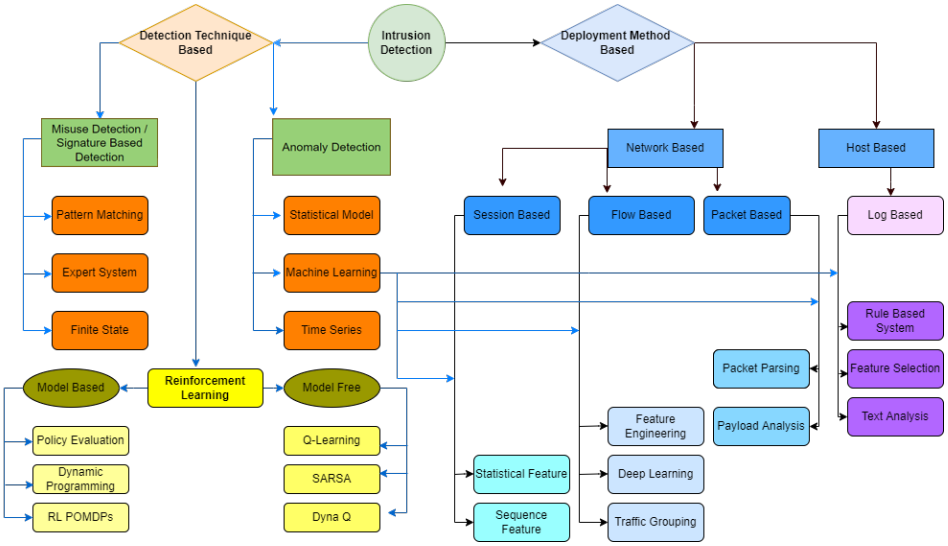

Figure 3. Types of intrusion detection systems.

2.3. Cybersecurity Framework

Cybersecurity forms an integral component of effective risk management in an organization. In order to handle cybersecurity risks, the role of NIST was modified to support the development of cybersecurity risk frameworks through the Cybersecurity Enhancement Act of 2014 [43]. This framework comprises three components, namely the framework core, implementation tiers and profiles. The framework core contains the basic guidance and standards that are imperative for an organization to manage risks posed by cyber threats. Implementation tiers proposed by NIST revolve around deciding on the scale of the proposed approach to mitigate threats. In other words, it allows an organization to understand the security standards required to guarantee protection. Finally, these frameworks support the creation of profiles that relate the cybersecurity activities to their respective outcomes. Profiles allow an organization to adapt their current approach to better suit demands.

The NIST framework is primarily divided into five functions [44]. These are identify, protect, detect, respond and recover. Identify revolves around the organization's capability to understand and effectively manage risks posed to assets such as data and physical devices by cyber threats. Protect is responsible for ensuring security mechanisms for the safe transmission of vital data and resources. Detect ensures that the organization is ready to implement techniques that can effectively recognize cyber threats. Respond ensures

that the organization is able to implement techniques that offer them the capability to respond to a threat. Finally, recover refers to activities that allow the organization to safely recover from a cybersecurity-related incident. Machine learning finds application in all these functions, especially in protect and detect. Protection categories such as access control can be implemented using machine learning. For example, NISTIR 8360 [45] utilizes a straightforward classification algorithm for verifying access control. Detect is perhaps the most widely explored area of machine learning. Almost all fields, such as anomaly detection and continuous monitoring, can benefit from a machine learning-based approached trained using a large amount of data. Discussion on machine learning techniques is given in the next section.

2.4. Cybersecurity Data

The availability of cybersecurity data drives machine learning in cybersecurity [46]. Machine learning techniques in cybersecurity are built on datasets, which are collections of records that contain information in the form of numerous qualities or features and related facts. As a result, it is vital to understand the nature of cybersecurity data, which encompasses a wide range of cyber events and critical components. The reasoning is that raw security data acquired from similar cyber sources can be used to investigate distinct patterns of security incidents. They can also be used to detect malicious behavior in order to develop a data-driven security model. This model will assist researchers to accomplish their objectives. In the field of cybersecurity, many datasets are available for various purposes, such as network intrusion analysis, malware analysis, phishing detection, fraud, anomalies or spam analysis. Numerous types of datasets have been outlined, including their varied aspects and incidents that are accessible on the internet, in Table 2, and we emphasize their use in diverse cyber applications based on machine learning techniques, analyzing and processing these networks effectively.

Table 2. Summary of cybersecurity databases.

Dataset	Description
IMPACT [47]	Mostly known as the Protected Repository for the Defense of Infrastructures Against Cyber Threats (PREDICT), a community that produces security-relevant network operation data and research. Repository provides regularly updated network operations data of cyber defense technology development.
SNAP [48]	Not specific to security, but there are several relevant graph datasets.
KYOTO [49]	Traffic data from Kyoto University's Honeypots.
KDD'99 Cup [50]	Contains 41 features that could be used to evaluate ML models. Threats are categorized into four major target labels, such as remote-to-local (R2L), denial of service (DoS), probing and user-to-remote (U2R).
NSL-KDD [51]	Updated variant of KDD'99 Cup dataset. Records that are redundant have been removed. It also addresses issues associated with class imbalance.
DARPA [52]	LLDOS 1.0 and LLDOS 2.0.2 attack scenario data from the Authenticated Intrusion Detection System (IDS). MIT Lincoln Laboratory collects data traffic and threats from the DARPA dataset in order to evaluate network intrusion detection systems (NIDS).
UNSW-NB15 [53]	It has 49 independent features spread over nine different threat types, including DoS, which were gathered from the University of New South Wales (UNSW) cybersecurity Lab in 2015. UNSW-NB15 can be used for evaluation of ML-based anomaly detection systems in cyber applications.
ADFA IDS [54]	This is an intrusion dataset with different versions, named ADFA-LD and ADFA-WD, that is issued by the Australian Defense Academy (ADFA). This dataset is designed to evaluate host-based IDS.

Table 2. *Cont.*

Dataset	Description
MAWI [55]	A cybersecurity dataset regulated by Japanese network research institutions and academic institutions that is commonly used to detect and assess DDoS threats using machine learning techniques.
CERT [56]	The purpose of creating user activity logs was to validate insider-threat detection algorithms in this dataset. Based on machine learning, it can be used to track and evaluate user behavior.
Bot-IoT [57]	This is a dataset that includes authentic and simulated Internet of Things (IoT) network traffic, as well as various assaults for network forensic analytics in the IoT space. Bot-IoT is primarily used in forensics to assess reliability using multiple statistics and machine learning techniques.
DGA [58]	The Alexa Top Sites dataset reliably hosts domain names that are benign. Malicious domain names are collected from OSINT and DGArchive. These datasets find perfect application in DGA botnet detection or domain classification using automated ML models.
CTU-13 [59]	This is a labeled malware dataset including background traffic, botnet and normal user activities, which was captured at CTU University, Czech Republic. CTU-13 is used for data-driven malware analysis using machine learning techniques and to evaluate the standard malware detection system.
CAIDA [60]	The CAIDA'07 and CAIDA'08 datasets contain DDoS attack traffic and normal standard traffic history. They are primarily used to assess machine learning-based DDoS attack detection models and to spot internet DOS activities.
CIC-DDoS2019 [61]	The Canadian Institute for Cybersecurity has compiled a database of historical DDoS assaults. CIC-DDoS is an excellent network traffic behavioral analytics tool for detecting DDoS attacks using machine learning approaches.
ISCX'12 [62]	This dataset contains 19 features and 19.11% of the network traffic belongs to DDoS attacks. ISCX'12 was documented at the Canadian Institute for Cybersecurity and is well known for its use in the evaluation of the effectiveness of machine learning-based network intrusion detection modeling.
Malware [63]	This is a collection of malicious files from several malware-based datasets such as the Genome Project, VirusTotal, Virus Share, Comodo, Contagio, Microsoft and DREBIN. These datasets are commonly used for data-driven malware analysis and evaluation of existing malware detection systems utilizing machine learning techniques.
EnronSpam [64]	Email-based datasets are difficult to collect because of privacy concerns. This dataset is a collection of emails with spam and ham classification.
DREBIN [65]	Researchers have created these datasets from the Drebin project, which is publicly available, in order to encourage and improve research on Android malware. There are 5560 programs in this collection, spanning 179 different malware categories. The samples were collected between August 2010 and October 2012, and the MobileSandbox initiative made them freely available to cybersecurity practitioners.
CDX 2009 Network USMA [66]	This dataset highlights the correlation found between IP addresses associated with the PCAP files to hosts that are found on the internal USMA network. Not all network modifications are reflected in this dataset.

3. Machine Learning Techniques in Cybersecurity

Machine learning (ML) is typically described as a branch of "Artificial Intelligence" that is closely related to data mining, computational statistics and analytics and data science, particularly focusing on allowing systems to learn from historical data [67,68]. As a result, machine learning models are often made up of a set of rules, procedures or complex functions and equations. These features can be used to uncover intriguing data patterns, recognize sequences or anticipate behavior [69]. As a result, ML could be useful in the field of cybersecurity. Figure 4 depicts a summarized view of the most frequently used machine learning techniques for cybersecurity. The taxonomy is primarily divided into three sections, namely deep learning models, shallow models and reinforcement learning.

Machine learning algorithms—in this case, shallow models—are further classified into and supervised learning and unsupervised learning. In supervised learning, the models usually do not have a dependent variable and mostly rely on the internal patterns available in the dataset to group the data into different categories. This can be achieved using different algorithms, such as K-means, Sequential Pattern Mining, DB scan [70] and the a

priori algorithm [71]. In supervised learning, the models usually have class labels to verify the predictions. Naïve Bayes, for example, uses probabilistic distribution to identify to which category a class label belongs. Decision trees create a tree-like structure based on a training set. For prediction, once the tree is built, any unknown record can be sorted based on the tree structure. Random forest [72] uses a similar approach, but instead of building one decision tree, it builds multiple decision trees and then uses a voting scheme to classify a record. Because of the collective nature of the decision-making process, random forest usually has higher classification accuracy. Support vector machine (SVM) [73] works by creating a linear decision boundary from the dataset. This can be compared to a binary classification. SVMs are also capable of transforming the data using a kernel trick. This allows SVMs to classify nonlinear datasets as well.

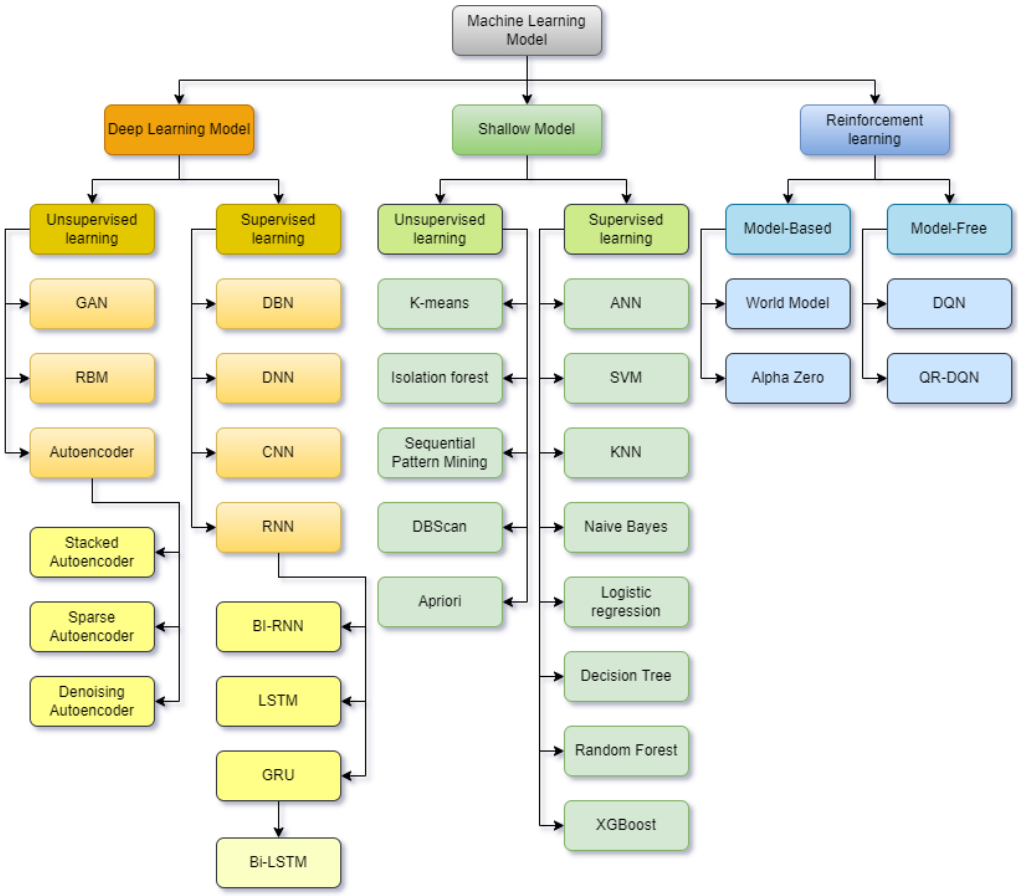

Figure 4. Taxonomy of machine learning algorithms.

Deep learning models can also classify or cluster algorithms. However, their approach is quite different from machine learning models. Unlike their counterparts, they do not have a fixed algorithm for prediction. Hence, they are also known as black box models because they analyze the data, identify patterns and use the patterns for production. Deep learning models use artificial neural networks, which are built using several perceptrons. These perceptrons are connected in a randomized fashion during the onset of model training. By looking at the data and training over a given time, the perceptrons gain values, also known as weights, that are better suited for classifying the dataset at hand. There are different varieties of deep learning models. Convolutional neural networks find application in classifying image data. They have also been used to classify cybersecurity datasets by transforming the data into a format that resembles an image. Recurrent neural networks (RNN) find application in classifying data that have a temporal aspect. Several improved versions of RNN include LSTM or long short-term memory, as well as Bi-LSTM. Unsupervised learning with deep learning includes autoencoders as well as generative

adversarial networks. Autoencoders mostly use feature reduction, where information is transformed into a compressed form before further processing. They are well adapted to compressing information in a meaningful way, thereby increasing the prediction accuracy.

Reinforcement learning explores a different approach of training a model to differentiate between long-term and short-term goals. Agents interact with their environment and, based on their actions, they are rewarded or penalized. The reward is variable, thereby teaching the model to become better. A popular example is DQN [74] or Deep Q Networks. In DQN, the mapping function between states and actions is accomplished using deep learning, thereby reducing the requirement for a large table of Q-learning (TQL). A variant of DQN is QR-DQN [75], which uses quantile regression to model the possible distributions, instead of returning the mean distribution. This can be compared to the difference between decision trees and random forest, summarized above.

In this section, different methods that can be used to solve machine learning techniques and how they are related to cybersecurity are explored. The traditional machine learning models are often known as shallow models for intrusion detection systems (IDS). Some of these techniques have been researched very extensively, and their approach is well known. They concentrate on tasks other than intrusion detection, and include tasks such as labeling, efficiently detecting attacks, as well as the optimal management of available and processed data.

3.1. Stages of a Cyber-Attack

Organizations can assess the cybersecurity risk to them and can identify certain security threats. They can then implement security controls or measures against these threats. They can utilize the National Institute of Standards and Technology (NIST) Special Publications, although they may not be a US federal agency or related contractor [76]. NIST Special Publications provide step-by-step guidance for applying a risk management framework to federal information systems. In this guidance, a set of security issues are identified and common controls or measures against these security issues/threats are listed. In a recent study, machine learning tools were suggested as efficient controls or measures [77]. Such measures can be applicable to all five phases of a cyber-attack.

There are five phases of a cyber-attack. They are reconnaissance, scan, attack (denial-of-service attacks, gain access using application and operating system attacks, network attacks), maintain access (using Trojans, backdoors, rootkits, etc.) and cover tracks and hiding. An interruption at any phase can either interrupt or halt the entire process of attack. Machine learning algorithms can be used in all of these phases to help fight against cyber-attacks by disrupting the attacker's workflow.

During the reconnaissance or preparation phase of the attack, an adversary uses techniques such as a social engineering attack (phishing, malicious call, etc.). Machine learning algorithms can look for email signatures and detect malicious or phishing email signatures and block them. There are cases when an attacker calls the target organization and impersonates a third party to obtain valuable information (known as "voice phishing" or "vishing"). Call source analysis using machine learning algorithms can flag and block such calls. Another example use of machine learning is scanning any external devices connected to the organization's property, e.g., a USB device. Such a scan prevents malicious software from propagating through such devices. Another example is when the adversary wishes to guess the access password to obtain unauthorized access (violating confidentiality) [78]. Rule-based machine learning algorithms can detect the most common passwords that are used by the organization's employees and can recommend a list of unrecommended passwords. This will hinder the reconnaissance step. Such machine learning algorithms can be placed in strategic locations, e.g., key machines and networks.

During the scan phase, sometimes called "Weaponization", the cyber-attacker or adversary exploits the vulnerabilities of the target system. The attacker uses automated tools, such as Metasploit, AutoSploit and Fuzzers [78]. Machine learning algorithms can be used to automatically scan and find the vulnerabilities by an ethical hacker before the

adversary can. For example, a machine learning-based penetration test can be implemented, specifically by integrating the algorithms into the penetration testing tools, e.g., Metasploit. Such algorithms, upon being used by a pen tester, can find novel weaknesses.

Machine learning algorithms are a strong measure against the attacks (phase 3 of cyber-attack). Machine learning algorithms that can be used to provide cybersecurity are linear regression, polynomial regression, logistic regression, naïve Bayes classifier, support vector machine, decision tree, nearest neighbor, clustering, dimensionality reduction, linear discriminant analysis and boosting [4]. The applications of these algorithms as a measure against cybersecurity problems are spam detection (includes phishing), malware detection, denial-of-service attacks (including DDoS) and network anomaly detection. Other forms of attacks are associated with biometric recognition, authentication, identity theft detection and social media analytics. Information leakage detection, APT or advanced persistent threat detection, hidden channel detection and software vulnerability detection are also some modern threats that need addressing.

During phase four of a cyber-attack, malware is used to maintain access by the attacker, e.g., Trojans, backdoors or rootkits. Machine learning algorithms can detect such malware traffic packets when the malware contacts the attacker and vice versa. For example, for malware detection, support vector machines (SVM) are an efficient option [79]. SVM was implemented using Weka to detect Android OS malware (260 samples), using static features analysis. Here, a black box method was used by analyzing the malware's behavior rather than executing the malware. In the first step, a Python code was used to extract Android application packages' (APK) features, one package at a time. Both malicious (201 samples) and benign (59 samples) APKs were selected. In the second step, an SVM classifier (Weka and LibSVM classifier) was trained by these features, to identify malware from these APKs. In the testing phase, the used APKs were downloaded from the repositories: Android, Malware Dataset, Kaggle, Drebin, Android Malshare and APKPure. The receiver operating characteristic or ROC curve was used to present the result. An enhancement of this application used the dynamic features of malware, as malware keeps changing its features. An SVM model can be trained to perform binary classification from a set of features of network packets. The trained classifier can detect a DDoS attack by identifying normal vs. abnormal network traffic, especially for IoT devices. Examples of the features used to train machine learning algorithms include the destination IP address, sequence number and minimum, maximum and average packets for each destination IP address, received signal strength indication, network allocation vector, value injection rate and inter-arrival time between consecutive frames, etc. The traffic information was collected by placing sensors at significant points of the network, e.g., at the gateway level, for a traffic session of 15–20 min. This classifier can be used as an extra security layer in IDS.

Another example is the application of various clustering techniques (K-means, DBSCAN and Hierarchical) [79]. Clustering is useful for malware detection, phishing attack detection, spam filtering and detecting the larger family of software bugs known as side-channel attack detection. In [79], both malware and goodware Android APKs were installed in an Android emulator. Then, their resource usage statistics (CPU and RAM) were recorded for all three clustering techniques. For all three clustering algorithms, a total of 217 data instances were used, with 145 for training and 72 for testing. The conclusion was that CPU-RAM usage statistics are not an efficient feature for clustering malware and goodware. The nearest neighborhood (NN) search is used in access control. For example, an NN is used to identify actual vs. forged biometrics (e.g., fingerprints) through classification based on their patterns [79]. The CSD200 model was used as the fingerprint scanner to take 100 samples (10 people, total 100 fingers). MATLAB was used to convert these images into an array or matrix. Such a machine learning algorithm can automatically make decisions as to whether a biometric is forged or original. In [79], decision trees were used, e.g., Iterative dichotomizer 3 (ID 3) and its successor, C4.5, to identify malware efficiently. The dataset used was from the Cardiff University Research Portal. In another research work [80], anomalous services running into the computer systems, both offline and online, were iden-

tified using a neural network-based model (NARX-RNN), AI-based multi-perspective SVM, principal component analysis (PCA) and hierarchical process tree-based reinforcement learning techniques.

During phase five or the covering tracks phase, the attacker wishes to confirm that their identification is not being tracked. They employ different techniques, including corrupting machine learning tools' training data to misidentify their data. The machine learning algorithms themselves can be robust but their training data may not be. Deceptive training data make the algorithm inefficient. This process of forging training data is called adversarial machine learning (AML). The severity is serious for cybersecurity applications. The countermeasures against such polluted data include game theory (non-cooperative game/Nash equilibrium, Zero-Sum Versus Non-Zero Sum Game, simultaneous move vs. sequential game or Bayesian Game) [81]. An example of AML is network traffic classification. Performing deep packet inspection is hard when the traffic payload is encrypted [82]. For such traffic, it is possible that the machine learning classifier (e.g., network scanning detector) is deceived by an adversary, to tag malware or botnet communications or NMap network scanning traffic as benign. It is possible that the adversary can mimic the features of benign traffic and can infer the classification output. What happens if the adversary's traffic is classified as malicious? The adversary does not obtain any feedback but their traffic will probably be blocked. This will give them an indication that their traffic has been classified as malicious and prompt the adversary to change the traffic signature. Improved machine learning techniques exist that can work as a measure against adversarial attacks [83]. For example, an activation clustering method was introduced that identifies the hidden layer of a deep neural network where an adversarial trigger lies. Using empirical learning algorithms, poisonous data points can be identified when poisoning attacks happen against an SVM.

3.2. Supervised Learning

Supervised learning relies on useful information in historical labeled data. Supervised learning is performed when targets are predefined to reach from a certain set of inputs, i.e., task-driven approach. Classification and regression methods are the most popular supervised learning techniques [84]. These methods are well known for classifying or predicting the target variable for a particular security threat. For example, classification techniques can be used in cybersecurity to indicate a denial-of-service (DoS) attack (yes, no) or identify distinct labels of network risks, such as scanning and spoofing. Naive Bayes [85], support vector machines (SVM) [86], decision tree [87,88], K-nearest neighbors [89], adaptive boosting [90] and logistic regression [91] are some of the most well-known classification techniques in shallow models.

Naive Bayes finds a good amount of use in cybersecurity. The authors in [92] used the naive Bayes classifier from the Weka package and KDD'99 data for training and testing. Data were grouped into the four attack types (probe and scan, DoS, U2R and R2L) and their classifier achieved 96%, 99%, 90% and 90% testing accuracy, respectively. The cumulative false positive rate was 3%. The authors in [93] developed a framework using a simple form of Bayesian network using the KDD'99 data and used categories to depict different attack scenarios. Solving an anomaly detection problem, the reported results were 97%, 96%, 9%, 12% and 88% accuracy for normal, DoS, R2L, U2R and probe or scan categories, respectively. The false positive rate was not reported but can be inferred to be less than 3%. Naive Bayes was also used as one of the methods in [94] to solve a DoS problem, which attempted to resolve the botnet traffic in filtered Internet Relay Chat (IRC), therefore determining the botnet's existence and origin. The study conducted used TCP-level data that were collected from 18 different locations on the Dartmouth University campus' wireless network. This data collection occurred over a span of four months. A filter layer was used to extract IRC data from the network data. Labeling was a challenge so the study utilized simulated data for the experiments. The performance of the Bayesian network showed 93% precision with a false positive rate of 1.39%. C4.5 decision trees were also used for comparison

and achieved 97% precision, but the false positive rates were higher, at 1.47% and 8.05%, respectively.

In [95], the authors used an SVM classifier to detect DDoS attacks in a software-defined network. Experiments were conducted on the DARPA dataset, comparing the SVM classifier with other standard classification techniques. Although the classifier had higher accuracy,the SVM took more time, which is an obvious flaw. In [96], the authors used a least-squares SVM to decrease the training time on large datasets. Using three different feature extraction algorithms, they reduced the number of features from 41 to 19. The data were resampled to have around 7000 instances for each of the five classes of the KDD'99 dataset. Overall, the classification was reported at 99% for DoS, probe or scan, R2L and normal classes and 93% for U2R. Research in [97] utilized a robust SVM, which is a variation of SVM where the discriminating hyperplane is averaged to be smoother and the regularization parameter is automatically determined. Preprocessing training and testing of data was done on the Basic Security Module from the DARPA 1998 dataset. It showed 75% accuracy with no false positives and 100% accuracy with 3% false positives.

In [98], the authors utilized decision trees to generate detection rules against denial-of-service and command injection attacks on robotic vehicles. Results showed that different attacks had different impacts on robotic behavior. A decision tree-based intrusion detection system that may change after intrusion by analyzing the behavior data through a decision tree was implemented in [99]. The model was used to prevent advanced persistent threat (APT) attacks, which use social engineering to launch various forms of intrusion attacks. The detection accuracy in their experiments was 84.7%, which is very high for this experiment. Decision trees were also used in [100], where the authors replaced the misuse detection engine of SNORT with decision trees. SNORT is a known tool that follows the signature-based approach. The authors utilized clustering of rules and then derived a decision tree using a version of an ID3 algorithm. This rule clustering reduced the number of comparisons to determine which rules were triggered by given input data, and the decision tree located the most discriminating features of the data, thereby achieving parallel feature evaluation. This method achieved superior performance to that of SNORT when applied to the 1999 DARPA intrusion detection dataset. The results varied drastically depending on traffic type. The fastest were up 105%, with an average of 40.4% and minimum of 5% above the normal detection speed of SNORT. The number of rules was also increased from 150 to 1581, which resulted in a pronounced speed-up versus SNORT.

Ensemble learning techniques such as random forest (RF) have also been explored in cybersecurity research. Random forest uses multiple decision trees to draw a conclusion and can be more accurate than a single decision tree. In [101], the authors employed a random forest algorithm on the KDD dataset for misuse, anomaly and hybrid-network-based intrusion detection. Patterns were created by the random forest and matched with the network for misuse detection. To detect anomalies, the random forest detected novel intrusions via outliers. Using the patterns built by the model, new outliers were also discovered. The study implemented a full system solution including anomaly detection. Data were grouped into majority attacks and minority attacks. This hybrid system achieved superior performance for misuse detection, where the error rate on the original dataset was 1.92%, and it was 0.05% for the balanced dataset.

Regression algorithms are useful for forecasting a continuous target variable or numeric values, such as total phishing attacks over a period of time or network packet properties. In addition to detecting the core causes of cybercrime, regression analysis can be utilized for various risk analysis forms [102]. Linear regression [67], support vector regression [86] and random forest regressor are some of the popular regression techniques. The main distinction between classification and regression is that, in regression, the output variable is numerical or continuous, but in classification, the projected output is categorical or discrete. Ensemble learning is an extension of supervised learning that mixes different shallow models, e.g., XGBoost and random forest learning [72], to complete a specific security task. A summary of supervised approaches is shown in Table 3.

Table 3. Shallow machine learning algorithms Used in cybersecurity.

Algorithm	Objective	Dataset	Accuracy	Reference
Naive Bayes	Can be used to analyze continuous and discrete values. Features are evaluated in a mutually exclusive fashion, making it relatively fast, thereby finding applicability in real-time decision making.	KWeka package, KDD 1999	90–99%	[103]
		KDD 1999	97%	[93]
		TCP data collected from the Dartmouth University campus' wireless network	93%	[94]
Support Vector Machines	Effective in high-dimensional spaces. Relatively memory-efficient. Numerical and categorical features.	DARPA	95.11%	[95]
		KDD-99	93–99%	[96]
		DARPA 1998	75–100%	[97]
Decision Tree	Requires little data preparation. Can be used to analyze continuous and discrete data. Can be generalized using dynamic tree-cut parameters.	TCP data collected from the Dartmouth University campus' wireless network	97%	[94]
		3000 behavior events collection	84.7%	[99]
		KDD dataset	94.7%	[101]
Sequential Pattern Mining	Frequent sequential patterns for a frequency support measure.	DARPA 1999 and 2000	93%	[104]
DBSCAN	Identify outliers, separate clusters of high density from clusters of low density.	KDD-99	98%	[105]
ADMIT	Not reliant on a lot of labeled data. Uses a dynamic clustering technique Modified form of K-means clustering.	Data collected from UNIX users from Purdue University	80%	[106]
A priori algorithm	The resulting rules are intuitive. Does not require labeled data as it is fully unsupervised.	Nine different-sized custom databases	70–100%	[107]
Radial Basis Function (RBF)	Real-time network anomaly detection.	KYOTO	95.6%	[108]
Random forest	Multi-class classification of network traffic threats.	KDD'99 Cup	99.0%	[109]
Extra-tree classifier (ETC)	Multi-class classification of DoS, probe, R2L and U2R.	KDD'99 Cup	99.51%	[110,111]
Radial Basis Function (RBF)	Comparative classification between lazy, eager learning and deep learning.	DARPA	97.41%	[108,112]
Random forest	Comparative classification between lazy, eager learning and deep learning.	UNSWNB15	95.43%	[113,114]
Random forest	Android malware detection.	DREBIN	94.33%	[115–117]
XGBoost	Classification of spam and ham from emails.	ENRON Spam	98.67%	[118–120]

3.3. Unsupervised Learning

The main goal of unsupervised learning, also known as data-driven learning, is to uncover patterns, structures or relevant information in unlabeled data [121]. Risks such as malware can be disguised in a variety of ways in cybersecurity, including changing their behavior dynamically to escape detection. Clustering techniques, which are a form of unsupervised learning, can aid in the discovery of hidden patterns and insights in datasets, as well as the detection of indicators of sophisticated attacks. In addition, clustering algorithms can effectively spot anomalies and policy violations, recognizing and eliminating noisy occurrences in data, for example. K-means [122] and K-medoids [123] are clustering

algorithms used for partitioning, while single linkage [124] and complete linkage [125] are some of the most widely utilized hierarchical clustering methods in various application sectors. Furthermore, well-known dimensionality reduction techniques such as linear discriminant analysis (LDA), principal component analysis (PCA) and Pearson correlation, as well as positive matrix factorization, can be used to handle such problems [67]. Another example is association rule mining, in which machine learning-based policy rules can learn to avert cyber incidents. The rules and logic of an expert system are normally manually documented and implemented by a knowledge engineer working with a domain expert [30,121,126]. In contrast, association rule learning identifies the rules or associations among a set of security aspects or variables in a dataset [127]. Different statistical analyses are performed to quantify the strength of associations [102]. In the domain of machine learning and data mining, various association rule mining methods have been presented, such as tree-based [128], logic-based [129], frequent pattern-based [130–132], etc. Moreover, a priori [130], a priori-TID and a priori-Hybrid [130], AIS [127], FP-Tree [128], RARM [133] and Eclat [134] have been used extensively for association rule learning. These algorithms are able to resolve such difficulties by creating a set of cybersecurity policy rules.

The authors in [104] applied sequential pattern mining on the DARPA 1999 and 2000 data to reduce the redundancy of alerts and minimize false positives. Using the a priori algorithm, they discovered attack patterns and created a pattern visualization for the users with a support threshold of 40%. They were able to detect 93% of the attacks in twenty seconds. The study reported a real-time scenario with 84% attack detection, with the main variation coming from the support threshold. In [105], DBSCAN was used as a clustering method to group normal versus anomalous network packets in the KDD'99 dataset. The dataset was preprocessed and features were selected using correlation analysis. In this study, the performance was shown at 98% for attack or no-attack detection. The authors in [106] took a different data mining approach to create an intrusion detection system called ADMIT. This system does not rely on labeled data; rather, it builds user profiles incrementally using a dynamic clustering technique. It uses sequences, which are a list of tokens collected from users' data, to build a profile. These sequences are classified as normal or anomalous based on using a modified form of K-means clustering. They set a value of 20 for the maximum sequence length, which resulted in 80% performance accuracy with a 15% false positive rate. A unique algorithm based on the signature a priori algorithm was used to find new signatures of attacks from existing attack signatures, proposed in [107]. Here, the authors compared their algorithm processing time to that of the a priori and found that their algorithm was much faster. A summary of unsupervised approaches is also shown in Table 3.

3.4. Artificial Neural Networks (ANN)

ANN is a segment of machine learning, in the area of Artificial Intelligence, that is a computationally complex model inspired by the biological neural networks in the human brain [67]. The basic idea behind ANN was first introduced in the 1940s, with ANNs becoming a popular idea and technology in the late 1970s and continuing into the 1990s. Although SVMs were prominent in the 1990s, overshadowing the ANNs, they have received popularity recently and are steadily increasing in use. ANNs consist of multiple neurons that work together in layers to extract information from the dataset. The primary difference is the performance of ANN versus shallow machine learning as the amount of security data grows. The number of studies of ANN-based intrusion detection systems has increased rapidly from 2015 to the present. In [135], the authors utilized ANNs to detect misuse. Using data generated by a RealSecure network monitor, ten thousand events were collected, of which 3000 were simulated attacks by programs. Preprocessing of the data was performed and ten percent of the data were selected randomly for testing, with the rest used for training the ANN. The error rates for training and testing were 0.058 and 0.070, respectively. Each packet was categorized as normal or attack. A combination of keyword selection and ANN was proposed in [136] to improve IDS. Keyword selection was

performed on Telnet sessions and statistics were derived from the number of times that the keywords occurred (from a predefined list). These statistics were given as input to the ANN and the output identified the probability of an attack. The authors in [137] compared fuzzy logic and artificial neural networks to develop comprehensive intrusion detection systems and tested them using the KDD'99 dataset. They presented a detailed discussion on the preprocessing, training and validation of the proposed approach. The "class" characteristic in this dataset, which is made up of around 5 million data instances with 42 properties, determines whether a particular instance is a typical connection instance or one of the four types of attacks that need to be recognized. Five different machine learning approaches were compared, among which the FC-ANN-based approach [138] and the hierarchical SOM-based approach [139] were the best detectors.

Deep learning models, which are a form of ANN, learn feature representations directly from the original data, such as photos and texts, without the need for extensive feature engineering. As a result, deep learning algorithms are more effective and need less data processing. For large datasets, deep learning methods have a significant advantage over classical machine learning models. Some of the widely used deep learning techniques in cybersecurity include deep belief networks (DBNs), convolutional neural networks (CNNs) and recurrent neural networks (RNNs) as supervised learning models. Several variants of autoencoders, restricted Boltzmann machines (RBMs) and generative adversarial networks (GANs) have been used with success for unsupervised learning. The authors in [140] used DBNs to detect malware with an accuracy of 96.1%. The DBNs used unsupervised learning to discover layers of features and then used a feed-forward neural network to optimize discrimination. DBNs can learn from unlabeled data, so, in the experiments, DBNs provided a better classification result than SVM, KNN and decision tree. DBNs were also used in [141] to create an IDS. The proposed four-layer model was used on the KDD'99 Cup dataset, where the authors reported accuracy, precision and false acceptance rate (FAR) of 93.49%, 92.33% and 0.76%, respectively. In [142], a DBN-based ad hoc network intrusion detection model was developed with an experiment on the Network Simulator (NS2) platform. The experiment showed that this method can be added to the ad hoc network intrusion detection technology. Accuracy and FAR were reported as 97.6% and 0.9%, respectively. A deep learning approach called DeepFlow was proposed in [143] to directly detect malware in Android applications. The architecture consisted of three components for feature extraction, feature coarse granularity and classification. Two modules were used to assess malware sources from the Google Play Store. Experiments showed that DeepFlow outperformed SVM, ML-based algorithms and multi-layer perceptron (MLP). A summary of neural network approaches is shown in Table 4.

Table 4. Deep machine learning algorithms used in cybersecurity.

Algorithm	Objective	Dataset	Accuracy	Reference
ANN	Abilities to learn, classify and process information; faster self-organization.	RealSecure network monitor	96.5%	[135]
DeepFlow	Custom-developed to distinguish malware. It uses the static taint analysis tool FlowDroid. Identifies sensitive data flows in Android apps.	Features extracted from 11,000 benign and malicious apps from Google Play Store	95.05%	[143]
DBNs	Discovers layers of features and uses feed-forward neural network to optimize discrimination.	Custom dataset	96%	[140]
		KDD'99 Cup	93.49%	[141]
		Network feature sample	97.60%	[142]

Table 4. *Cont.*

Algorithm	Objective	Dataset	Accuracy	Reference
Deep Belief Network (DBN)	Real-time network anomaly detection.	KYOTO	98%	[144]
Gated Recurrent Unit (GRU)	Multi-class classification of network traffic threats.	KDD'99 Cup	98.64%	[145,146]
CNN-LSTM	Multi-class classification of DoS, probe, R2L and U2R.	KDD'99 Cup	99.70%	[147–150]
Deep Feed Forward (DFF)	Comparative classification between lazy, eager learning and deep learning.	DARPA	99.63%	[151,152]
Temporal convolutional networks (TCN)	Comparative classification between lazy, eager learning and deep learning.	UNSWNB15	99.6%	[153–155]
CNN	Android malware detection.	DREBIN	99.29%	[156–158]
Bi-LSTM	Classification of spam and ham from emails.	ENRON Spam	98.84%	[103,159]

4. Future Improvements and Challenges for ML-Based Cybersecurity

Various research concerns and obstacles in machine learning in cybersecurity must be addressed while extracting insights from relevant data to make data-driven intelligent cybersecurity decisions. The issues in the following section range from data collection to decision making.

4.1. Cybersecurity Dataset Availability

In cybersecurity, source datasets are critical, as they are in machine learning. The majority of publicly available datasets are outdated and may not be sufficient in identifying the undocumented behavioral patterns of various cyber-attacks. Even though current data can be translated into knowledge after a series of processing steps, there is still a lack of understanding of the nature of recent attacks and their recurrence patterns. As a result, additional processing or machine learning approaches may result in a low accuracy rate when it comes to making final decisions. One of the fundamental obstacles in using machine learning techniques in cybersecurity is establishing a large number of recent cybersecurity datasets for particular issues such as attack prediction or intrusion detection.

4.2. Cybersecurity Dataset Standard

The cybersecurity datasets could be unbalanced, noisy, incomplete, irrelevant or contain inconsistent examples of a particular security violation. The quality of the learning process and the performance of machine learning-based models may be harmed by such issues in a dataset [160,161]. To build a data-driven solution for cybersecurity, such problems in data need to be addressed before the application of machine learning techniques. It is imperative to understand the problems in cybersecurity data and effectively address these issues using existing or novel algorithms to perform tasks such as malware and intrusion detection, among others. Some methods to solve these issues are associated with feature engineering [154], where model features are analyzed to remove correlated features. This technique reduces data dimensionality, thereby reducing complexity. Handling data imbalance is imperative, which can be done by utilizing hybrid models, as reported in [137], or generating synthetic data [162,163]. Other issues pertaining to data leakage should also be addressed.

4.3. Hybrid Learning

Signature-based intrusion detection methods are the most common and well-established techniques in the cybersecurity domain [36,164]. However, these algorithms may overlook undiscovered assaults or incidents due to missing features, substantial feature reduction or poor profiling. Anomaly-based or hybrid techniques, including anomaly-based and signature-based detection techniques, can be utilized to overcome these shortcomings. To

extract the target insight for a particular problem domain, such as intrusion detection, malware analysis, phishing detection and so on, a hybrid learning technique combining multiple machine learning techniques is useful. A combination of deep learning, statistical analysis and machine-learning methods can also be used to make an intelligent decision for corresponding cybersecurity solutions.

4.4. Feature Engineering in Cybersecurity

Due to the vast volume of network traffic data and large number of minor traffic aspects, the effectiveness and performance of a machine learning-based security model have frequently been challenged. Several techniques, such as principal component analysis, have been used to deal with the high dimensionality of data [165,166], including singular value decomposition (SVD) [167,168] and Linear Discriminant Analysis (LDA), for example. Contextual links between suspicious actions and low-level information in datasets may be useful. Such contextual data might be processed through an ontology or taxonomy for further investigation. As a result, another research challenge for machine learning approaches in cybersecurity is how to effectively choose the ideal features or extract the significant characteristics while considering machine-readable features as well as contextual features for efficient cybersecurity solutions.

4.5. Data Leakage

Data leakage (or leaking) happens when the training dataset contains relevant data, but similar data are unavailable or show wide variation as the models are utilized to make predictions [169]. Typically, this results in highly optimistic predictions during the model-building process, followed by the unwelcome shock of disappointing outcomes once the prediction model is put into use and tested on new data. Research in [170] identifies the problem as "leaks from the future", calling it "one of the top 10 data mining mistakes", and recommends using exploratory data analysis (EDA) to identify and eliminate leakage sources. EDA can be helpful to increase the dataset's effectiveness, thereby making the machine learning models more accurate at predicting unknown data. In recent work [171], finding and using leakage has been discussed as one of the crucial elements for winning data mining competitions, and the authors showed it to be one of the critical elements for failing a data mining application. Another study [172] describes the inclusion of giveaway qualities that forecast the target in data mining competitions since they are introduced later in the data collection process. Research in [173] provides a review of some common classifiers that are used to classify documents and datasets that have been formulated for binary prediction. To prevent leaking, researchers implemented a two-stage approach that involved marking each observation with a legitimacy tag during data collection and then observing what they referred to as a learn–predict separation. The proposed approach was significantly useful as they witnessed a maximum of 91.2% in naïve Bayes, 87.5% using k-NN and 94.2% with centroid based on various categories. In many cases, where the machine learning scientist has no control over the data collection procedure, EDA is a valuable technique for identifying leaks [174], and it could be promising for future work.

4.6. Homomorphic Encryption

Homomorphic Encryption (HE) is considered as one of the greatest advancements in cryptography [175]. HE provides access to a non-trustworthy third-party to process data without any clue, thereby allowing access to confidential data. The user end or the unauthorized remote server obtains access to the encrypted data and not the secret key for decryption. Hence, the host can be assured that the data are not leaked outside of the domain. HE has a long list of applications, including cloud computing, financial transactions, quantum computing threat shields, etc. HE can be applied in two ways, partial and fully. Fully Homomorphic Encryption (FHE) makes the machine learning training process easier without data leakage. Deep learning and shallow machine learning algorithms heavily rely on domain data, which are often difficult to share publicly [176]. FHE

has facilitated a new process to delegate these kinds of sensitive data sharing without sharing the actual meaningful data. One of the major drawbacks of FHE is its limitation to the use of integers [177]. Thus, researchers are trying to find matrix-based schemes for FHE. Recent research has proven effective by using the lowest degree of polynomial approximation functions such as Chebyshev with a continuous function such as sigmoid. This has created a new encryption over FHE to use in homogeneous networking [178]. Federated learning has accelerated multi-party joined learning processes by applying FHE in case of image data with sample expansions [179]. Medical and health-related data are always highly confidential since they contain Protected Health Information (PHI). One of the major breakthroughs happened when healthcare data became accessible through FHE. Researchers proved multiple effective ways to use medical images or other data from Internet of Medical Things (IoMT) using machine learning-based HE. In 2019, HE combined with chaotic mapping successfully secured data transfer, but the computational privacy was vulnerable [180]. In 2021, HE was combined with secret sharing and computation was performed on the edge computation layer. Moreover, the mathematical operation was conducted in a distributed manner, but no data leakage happened [181]. A recurrent neural network, CryptoRNN, has been introduced recently, which is mostly focused on the privacy preservation of blockchain technology [182]. Cloud-based FHE integration is the most advanced and commonly used because of the versatility of domain data access and vast computational power. The Machine Learning as a Service Platform (MLaaS) provides a wide variety of machine learning algorithms to enable FHE to protect confidential data [183]. In the early exploration of HE in Wireless Sensor Networks (WSNs), researchers experimented with the performance of HE in a network simulation NS-2 tool, where, for each experimental agent, the environment remained similar. For data aggregation operation, FHE consumed less time than DAA, which decrypts hop-by-hop, and achieved the time complexity of O(n) [184]. HE increases the global data stream and machine learning's practical applications by scaling, along with enhancing the overall cybersecurity [185].

4.7. Quantum Computing

While in their infancy, quantum computers were once established as having the potential to break the security offered by asymmetric encryption techniques [186]. Asymmetric key encryption relies on public and private keys. These keys are generated by factoring two extremely large prime numbers. Factoring small prime numbers is possible but keys that are very large can take thousands of years to decrypt, making our data secure. Shor's algorithm [187] provides an alternative solution to factor these large prime numbers, but, again, it is slow. Quantum computing, with its principal of superposition, can rapidly derive the factors at a fraction of the time that a binary computing system would take. This makes algorithms such as RSA and DES, elliptic curve algorithms such as ECDSA and digital signature algorithms no longer secure. The authors in [188] mentioned that to break a 56-bit DES, Grover's algorithm [189] utilizing quantum computing would only require 185 searches for key identification. Symmetric key algorithms such as AES are still resistant to quantum computing. Researchers are exploring both quantum and mathematical techniques to circumvent these limitations. An example is the BB84 protocol [190], which is a type of quantum key distribution. Mathematical approaches such as lattice-based cryptography [191] are also being explored. While quantum computing can be detrimental to asymmetric encryption, it can also speed up machine learning if used as sub-routines [192]. This can significantly reduce the prediction time if used by algorithms such as SVM, which can require a lot of time, implementing kernel transformations to derive a hyperplane. They can also be used in deep learning if configured properly. However, there are some challenges since quantum neural networks have linear dynamics [193].

5. Conclusions

In this study, which was prompted by the growing importance of cybersecurity and machine learning technology, we looked at how machine learning techniques are utilized to make data-driven intelligent decision making in cybersecurity systems and services successful. A discussion about how it affects security data, both in terms of gaining insight into security occurrences and evaluating the data, has also been presented. The focus was on machine learning advancements and difficulties in the cybersecurity area. Therefore, a discussion about the security dataset and services that go with it has been presented. Our contribution also looked at how machine learning techniques might affect the cybersecurity area and the security concerns that still exist. Thus far, traditional security options receive much attention, while machine learning algorithm-based security systems receive less attention. The study uses an IDS classification to present the many machine learning techniques employed in this discipline, with data sources as the major theme. This is followed by an explanation of IDSs' application to various data sources using this classification. Because IDSs are designed to identify attacks, it is critical to choose the appropriate data source based on the attack's characteristics. Logs include rich semantic information that can be utilized to detect SQL injection, U2R and R2L attacks, as well as for further analysis using machine learning techniques. Packets contain communication data that can be used to detect U2L and R2L assaults. Various key issues in security analysis to show the interest of future research ideas in the domain of machine learning with cybersecurity have been discussed. As attacks evolve, so will the machine learning techniques, making this a very dynamic field. To mitigate the damage caused by cyber-attacks, constant support is necessary from not only machine learning experts but from researchers and institutions, responsible for providing the latest datasets for training, thereby making this a collective approach.

Future work will focus on a feasibility analysis of machine learning and its deployment to monitor real-time traffic. This is extremely challenging due to the diverse nature of internet packets. Another aspect that complicates the approach is that all packets are encrypted. Hence, a focus of our future research will be Homomorphic Encryption. A brief discussion on this new technique has been presented in the review, but it requires a detailed analysis. Threats revolving around quantum computing and its impact on public key encryption will also be explored.

Author Contributions: Conceptualization, M.A., K.E.N., R.G. and M.M.C.; methodology, M.A. and K.E.N.; software, M.A.; validation, M.A., K.E.N. and R.G.; data curation, M.A., N.R., J.F.C. and K.E.N.; writing—original draft preparation, M.A., R.G., N.R., J.F.C. and M.M.C.; writing—review and editing, R.G., N.R. and M.M.C.; visualization, R.G., M.A., N.R. and M.M.C.; supervision, K.E.N.; funding acquisition, K.E.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Department of Computer Science at North Dakota State University.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
Bi-LSTM	Bidirectional Long Short-Term Memory
GRU	Gated Recurrent Units
RF	Random Forest
NB	Naive Bayes
DoS	Denial of Service
DDoS	Distributed Denial of Service
SVM	Support Vector Machines
ICT	Information and Communication Technology
MITM	Man-in-the-Middle attack
IDS	Intrusion Detection System
FAR	False Acceptance Rate
RBF	Radial Basis Function

References

1. Li, S.; Da Xu, L.; Zhao, S. The internet of things: A survey. *Inf. Syst. Front.* **2015**, *17*, 243–259. [CrossRef]
2. Sun, N.; Zhang, J.; Rimba, P.; Gao, S.; Zhang, L.Y.; Xiang, Y. Data-driven cybersecurity incident prediction: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1744–1772. [CrossRef]
3. McIntosh, T.; Jang-Jaccard, J.; Watters, P.; Susnjak, T. The inadequacy of entropy-based ransomware detection. In Proceedings of the International Conference on Neural Information Processing, Sydney, Australia, 12–15 December 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 181–189.
4. Alazab, M.; Venkatraman, S.; Watters, P.; Alazab, M. Zero-day malware detection based on supervised learning algorithms of API call signatures. In Proceedings of the Ninth Australasian Data Mining Conference (AusDM'11), Ballarat, Australia, 1–2 December 2011.
5. Shaw, A. Data breach: From notification to prevention using PCI DSS. *Colum. JL Soc. Probs.* **2009**, *43*, 517.
6. Gupta, B.B.; Tewari, A.; Jain, A.K.; Agrawal, D.P. Fighting against phishing attacks: State of the art and future challenges. *Neural Comput. Appl.* **2017**, *28*, 3629–3654. [CrossRef]
7. Geer, D.; Jardine, E.; Leverett, E. On market concentration and cybersecurity risk. *J. Cyber Policy* **2020**, *5*, 9–29. [CrossRef]
8. Buecker, A.; Borrett, M.; Lorenz, C.; Powers, C. *Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security*; International Technical Support Organization: Riyadh, Saudi Arabia, 2010.
9. Fischer, E.A. *Cybersecurity Issues and Challenges: In Brief*; Library of Congress: Washington, DC, USA, 2014.
10. Chernenko, E.; Demidov, O.; Lukyanov, F. *Increasing International Cooperation in Cybersecurity and Adapting Cyber Norms*; Council on Foreign Relations: New York, NY, USA, 2018.
11. Papastergiou, S.; Mouratidis, H.; Kalogeraki, E.M. Cyber security incident handling, warning and response system for the european critical information infrastructures (cybersane). In Proceedings of the International Conference on Engineering Applications of Neural Networks, Crete, Greece, 24–26 May 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 476–487.
12. O'Connell, M.E. Cyber security without cyber war. *J. Confl. Secur. Law* **2012**, *17*, 187–209. [CrossRef]
13. Tolle, K.M.; Tansley, D.S.W.; Hey, A.J. The fourth paradigm: Data-intensive scientific discovery [point of view]. *Proc. IEEE* **2011**, *99*, 1334–1337. [CrossRef]
14. Benioff, M. Data, data everywhere: A special report on managing information (pp. 21–55). *The Economist*, 27 February 2010.
15. Cost of Cyber Attacks vs. Cost of Cybersecurity in 2021 | Sumo Logic. Available online: https://www.sumologic.com/blog/cost-of-cyber-attacks-vs-cost-of-cyber-security-in-2021/ (accessed on 10 May 2022).
16. Anwar, S.; Mohamad Zain, J.; Zolkipli, M.F.; Inayat, Z.; Khan, S.; Anthony, B.; Chang, V. From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. *Algorithms* **2017**, *10*, 39. [CrossRef]
17. Mohammadi, S.; Mirvaziri, H.; Ghazizadeh-Ahsaee, M.; Karimipour, H. Cyber intrusion detection by combined feature selection algorithm. *J. Inf. Secur. Appl.* **2019**, *44*, 80–88. [CrossRef]
18. Tapiador, J.E.; Orfila, A.; Ribagorda, A.; Ramos, B. Key-recovery attacks on KIDS, a keyed anomaly detection system. *IEEE Trans. Dependable Secur. Comput.* **2013**, *12*, 312–325. [CrossRef]
19. Saxe, J.; Sanders, H. *Malware Data Science: Attack Detection and Attribution*; No Starch Press: San Francisco, CA, USA, 2018.
20. Rainie, L.; Anderson, J.; Connolly, J. *Cyber Attacks Likely to Increase*; Pew Research Center: Washington, DC, USA, 2014.
21. Fischer, E.A. *Creating a National Framework for Cybersecurity: An Analysis of Issues and Options*; Library of Congress Washington DC Congressional Research Service: Washington, DC, USA, 2005.
22. Craigen, D.; Diakun-Thibault, N.; Purse, R. *Technology Innovation Management Review Defining Cybersecurity*; Technology Innovation Management Review: Ottawa, ON, Canada, 2014.

23. Goodman, S.E.; Lin, H.S. *Toward a Safer and More Secure Cyberspace*; National Academies of Sciences, Engineering, and Medicine: Washington, DC, USA, 2007; pp. 1–328. [CrossRef]
24. Jang-Jaccard, J.; Nepal, S. A survey of emerging threats in cybersecurity. *J. Comput. Syst. Sci.* **2014**, *80*, 973–993. [CrossRef]
25. Joye, M.; Neven, G. *Identity-Based Cryptography*; IOS Press: Amsterdam, The Netherlands, 2009; Volume 2,
26. Gisin, N.; Ribordy, G.; Tittel, W.; Zbinden, H. Quantum cryptography. *Rev. Mod. Phys.* **2002**, *74*, 145. [CrossRef]
27. Zou, C.C.; Towsley, D.; Gong, W. *A Firewall Network System for Worm Defense in Enterprise Networks*; Technical Report TR-04-CSE-01; University of Massachusetts: Amherst, MA, USA, 2004.
28. Corey, V.; Peterman, C.; Shearin, S.; Greenberg, M.S.; Van Bokkelen, J. Network forensics analysis. *IEEE Internet Comput.* **2002**, *6*, 60–66. [CrossRef]
29. Hu, V.C.; Ferraiolo, D.; Kuhn, D.R. *Assessment of Access Control Systems*; US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2006.
30. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 1–22. [CrossRef]
31. Brahmi, I.; Brahmi, H.; Yahia, S.B. A multi-agents intrusion detection system using ontology and clustering techniques. In Proceedings of the IFIP International Conference on Computer Science and Its Applications, Saida, Algeria, 20–21 May 2015; Springer: Berlin/Heidelberg, Germany, 2015, pp. 381–393.
32. Johnson, L. *Computer Incident Response and Forensics Team Management: Conducting a Successful Incident Response*; Newnes: Oxford, UK, 2013.
33. Qu, X.; Yang, L.; Guo, K.; Ma, L.; Sun, M.; Ke, M.; Li, M. A survey on the development of self-organizing maps for unsupervised intrusion detection. *Mob. Netw. Appl.* **2019**, *26*, 808–829. [CrossRef]
34. Radivilova, T.; Kirichenko, L.; Alghawli, A.S.; Ilkov, A.; Tawalbeh, M.; Zinchenko, P. The complex method of intrusion detection based on anomaly detection and misuse detection. In Proceedings of the 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 14–18 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 133–137.
35. Mosqueira-Rey, E.; Alonso-Betanzos, A.; Río, B.B.d.; Pineiro, J.L. A misuse detection agent for intrusion detection in a multi-agent architecture. In Proceedings of the KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, Wroclaw, Poland, 31 May–1 June 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 466–475.
36. Liao, H.J.; Lin, C.H.R.; Lin, Y.C.; Tung, K.Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [CrossRef]
37. Alazab, A.; Hobbs, M.; Abawajy, J.; Alazab, M. Using feature selection for intrusion detection system. In Proceedings of the 2012 International Symposium on Communications and Information Technologies (ISCIT), Sydney, Australia, 9–12 September 2012; IEEE: Piscataway, NJ, USA, 2012, pp. 296–301.
38. Viegas, E.; Santin, A.O.; Franca, A.; Jasinski, R.; Pedroni, V.A.; Oliveira, L.S. Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems. *IEEE Trans. Comput.* **2016**, *66*, 163–177. [CrossRef]
39. Xin, Y.; Kong, L.; Liu, Z.; Chen, Y.; Li, Y.; Zhu, H.; Gao, M.; Hou, H.; Wang, C. Machine learning and deep learning methods for cybersecurity. *IEEE Access* **2018**, *6*, 35365–35381. [CrossRef]
40. Dutt, I.; Borah, S.; Maitra, I.K.; Bhowmik, K.; Maity, A.; Das, S. Real-time hybrid intrusion detection system using machine learning techniques. In *Advances in Communication, Devices and Networking*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 885–894.
41. Ghanem, M.C.; Chen, T.M. Reinforcement learning for efficient network penetration testing. *Information* **2019**, *11*, 6. [CrossRef]
42. Alghamdi, M.I. Survey on Applications of Deep Learning and Machine Learning Techniques for Cyber Security. *Int. J. Interact. Mob. Technol.* **2020**, *14*, 210–224. [CrossRef]
43. Text—S.1353—113th Congress (2013–2014): Cybersecurity Enhancement Act of 2014 | Congress.gov | Library of Congress. Available online: https://www.congress.gov/bill/113th-congress/senate-bill/1353/text (accessed on 10 May 2022).
44. Cybersecurity, C.I. Framework for Improving Critical Infrastructure Cybersecurity. 2018. p. 4162018. Available online: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP (accessed on 10 May 2022).
45. Hu, V. *Machine Learning for Access Control Policy Verification*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2021.
46. Rizk, A.; Elragal, A. Data science: Developing theoretical contributions in information systems via text analytics. *J. Big Data* **2020**, *7*, 1–26. [CrossRef]
47. IMPACT. Available online: https://www.impactcybertrust.org/ (accessed on 10 May 2022).
48. Stanford Large Network Dataset Collection. Available online: https://snap.stanford.edu/data/index.html (accessed on 10 May 2022).
49. Traffic Data from Kyoto University's Honeypots. Available online: http://www.takakura.com/Kyoto_data/ (accessed on 10 May 2022).
50. KDD Cup 1999 Data. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 10 May 2022).
51. NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 10 May 2022).
52. 1998 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory. Available online: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset (accessed on 10 May 2022).

53. The UNSW-NB15 Dataset | UNSW Research. Available online: https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 10 May 2022).
54. ADFA IDS Datasets | UNSW Research. Available online: https://research.unsw.edu.au/projects/adfa-ids-datasets (accessed on 10 May 2022).
55. MAWI Working Group Traffic Archive. Available online: https://mawi.wide.ad.jp/mawi/ (accessed on 10 May 2022).
56. Insider Threat Test Dataset. Available online: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099 (accessed on 10 May 2022).
57. The Bot-IoT Dataset | UNSW Research. Available online: https://research.unsw.edu.au/projects/bot-iot-dataset (accessed on 10 May 2022).
58. Cucchiarelli, A.; Morbidoni, C.; Spalazzi, L.; Baldi, M. Algorithmically generated malicious domain names detection based on n-grams features. *Expert Syst. Appl.* **2021**, *170*, 114551. [CrossRef]
59. García, S.; Grill, M.; Stiborek, J.; Zunino, A. An empirical comparison of botnet detection methods. *Comput. Secur.* **2014**, *45*, 100–123. [CrossRef]
60. CAIDA Data—Completed Datasets—CAIDA. Available online: https://www.caida.org/catalog/datasets/completed-datasets/ (accessed on 10 May 2022).
61. Sharafaldin, I.; Lashkari, A.; Hakak, S.; Ghorbani, A.A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; [CrossRef]
62. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A. Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
63. Yang, L.; Ciptadi, A.; Laziuk, I.; Ahmadzadeh, A.; Wang, G. BODMAS: An open dataset for learning based temporal analysis of PE malware. In Proceedings of the 2021 IEEE Security and Privacy Workshops (SPW), Virtual, 27 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 78–84.
64. Keila, P.S.; Skillicorn, D.B. Structure in the Enron Email Dataset. *Comput. Math. Organ. Theory* **2005**, *11*, 183–199. [CrossRef]
65. Arp, D.; Spreitzenbarth, M.; Hübner, M.; Gascon, H.; Rieck, K. Drebin: Effective and Explainable Detection of Android Malware in Your Pocket. In Proceedings of the NDSS'14, San Diego, CA, USA, 23–26 February 2014.
66. Sangster, B.; O'connor, T.J.; Cook, T.; Fanelli, R.; Dean, E.; Adams, W.J.; Morrell, C.; Conti, G. *Toward Instrumenting Network Warfare Competitions to Generate Labeled Datasets*; United States Military Academy: New York, NY, USA, 2009.
67. Han, J.; Kamber, M.; Pei, J. Data mining concepts and techniques third edition. *Morgan Kaufmann Ser. Data Manag. Syst.* **2011**, *5*, 83–124.
68. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. Practical machine learning tools and techniques. *Morgan Kaufmann* **2005**, *2*, 578.
69. Dua, S.; Du, X. *Data Mining and Machine Learning in Cybersecurity*; CRC Press: Boca Raton, FL, USA, 2016.
70. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In In Proceedings of the KDD-94, Oregon, Portland, 2–4 August 1996; Volume 96, pp. 226–231.
71. Inokuchi, A.; Washio, T.; Motoda, H. An apriori-based algorithm for mining frequent substructures from graph data. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Lyon, France, 13–16 September 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 13–23.
72. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
73. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
74. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
75. Dabney, W.; Rowland, M.; Bellemare, M.; Munos, R. Distributional reinforcement learning with quantile regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
76. Force, J.T. Risk management framework for information systems and organizations. *NIST Spec. Publ.* **2018**, *800*, 37.
77. Breier, J.; Baldwin, A.; Balinsky, H.; Liu, Y. Risk Management Framework for Machine Learning Security. *arXiv* **2020**, arXiv:2012.04884.
78. Buchanan, B.; Bansemer, J.; Cary, D.; Lucas, J.; Musser, M. *Automating Cyber Attacks: Hype and Reality*; Center for Security and Emerging Technology: Washington, DC, USA, 2020. [CrossRef]
79. Thomas, T.; Vijayaraghavan, A.P.; Emmanuel, S. *Machine Learning Approaches in Cyber Security Analytics*; Springer: Berlin/Heidelberg, Germany, 2020.
80. Sakthivel, R.K.; Nagasubramanian, G.; Al-Turjman, F.; Sankayya, M. Core-level cybersecurity assurance using cloud-based adaptive machine learning techniques for manufacturing industry. *Trans. Emerg. Telecommun. Technol.* **2020**, *33*, e3947. [CrossRef]
81. Dasgupta, P.; Collins, J. A survey of game theoretic approaches for adversarial machine learning in cybersecurity tasks. *AI Mag.* **2019**, *40*, 31–43. [CrossRef]
82. De Lucia, M.J.; Cotton, C. Adversarial machine learning for cyber security. *J. Inf. Syst. Appl. Res.* **2019**, *12*, 26.
83. Xi, B. Adversarial machine learning for cybersecurity and computer vision: Current developments and challenges. *Wiley Interdiscip. Rev. Comput. Stat.* **2020**, *12*, e1511. [CrossRef]
84. Sarker, I.H.; Kayes, A.; Watters, P. Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *J. Big Data* **2019**, *6*, 1–28. [CrossRef]

85. John, G.H.; Langley, P. Estimating continuous distributions in Bayesian classifiers. *arXiv* **2013**, arXiv:1302.4964.
86. Keerthi, S.S.; Shevade, S.K.; Bhattacharyya, C.; Murthy, K.R.K. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Comput.* **2001**, *13*, 637–649. [CrossRef]
87. Salzberg, S.L. C4. 5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc. *Mach. Learn.* **1994**, *16*, 235–240. [CrossRef]
88. Sarker, I.H.; Colman, A.; Han, J.; Khan, A.I.; Abushark, Y.B.; Salah, K. Behavdt: A behavioral decision tree learning to build user-centric context-aware predictive model. *Mob. Netw. Appl.* **2020**, *25*, 1151–1161. [CrossRef]
89. Aha, D.W.; Kibler, D.; Albert, M.K. Instance-based learning algorithms. *Mach. Learn.* **1991**, *6*, 37–66. [CrossRef]
90. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. *ICML* **1996**, *96*, 148–156.
91. Le Cessie, S.; Van Houwelingen, J.C. Ridge estimators in logistic regression. *J. R. Stat. Soc. Ser. Appl. Stat.* **1992**, *41*, 191–201. [CrossRef]
92. Panda, M.; Patra, M.R. Network intrusion detection using naive bayes. *Int. J. Comput. Sci. Netw. Secur.* **2007**, *7*, 258–263.
93. Amor, N.B.; Benferhat, S.; Elouedi, Z. Naive bayes vs decision trees in intrusion detection systems. In Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 14–17 March 2004; pp. 420–424.
94. Carl, L. Using machine learning technliques to identify botnet traffic. In Proceedings of the 2006 31st IEEE Conference on Local Computer Networks, Tampa, FL, USA, 14–16 November 2006; IEEE: Piscataway, NJ, USA, 2006.
95. Kokila, R.; Selvi, S.T.; Govindarajan, K. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In Proceedings of the 2014 Sixth International Conference on Advanced Computing (ICoAC), Chennai, India, 17–19 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 205–210.
96. Amiri, F.; Yousefi, M.R.; Lucas, C.; Shakery, A.; Yazdani, N. Mutual information-based feature selection for intrusion detection systems. *J. Netw. Comput. Appl.* **2011**, *34*, 1184–1199. [CrossRef]
97. Hu, W.; Liao, Y.; Vemuri, V.R. Robust Support Vector Machines for Anomaly Detection in Computer Security. In Proceedings of the ICMLA, Los Angeles, CA, USA, 23–24 June 2003; pp. 168–174.
98. Vuong, T.P.; Loukas, G.; Gan, D.; Bezemskij, A. Decision tree-based detection of denial of service and command injection attacks on robotic vehicles. In Proceedings of the 2015 IEEE International Workshop on Information Forensics and Security (WIFS), Rome, Italy, 16–19 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6.
99. Moon, D.; Im, H.; Kim, I.; Park, J.H. DTB-IDS: An intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *J. Supercomput.* **2017**, *73*, 2881–2895. [CrossRef]
100. Kruegel, C.; Toth, T. Using decision trees to improve signature-based intrusion detection. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, 8–10 September 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 173–191.
101. Zhang, J.; Zulkernine, M.; Haque, A. Random-forests-based network intrusion detection systems. *IEEE Trans. Syst. Man Cybern. Part Appl. Rev.* **2008**, *38*, 649–659. [CrossRef]
102. Watters, P.A.; McCombie, S.; Layton, R.; Pieprzyk, J. Characterising and predicting cyber attacks using the Cyber Attacker Model Profile (CAMP). *J. Money Laund. Control* **2012**, *15*, 430–441. [CrossRef]
103. Kaddoura, S.; Alfandi, O.; Dahmani, N. A spam email detection mechanism for english language text emails using deep learning approach. In Proceedings of the 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Virtual, 10–13 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 193–198.
104. Li, Z.; Zhang, A.; Lei, J.; Wang, L. Real-time correlation of network security alerts. In Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE'07), Hong Kong, China, 24–26 October 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 73–80.
105. Blowers, M.; Williams, J. Machine learning applied to cyber operations. In *Network Science and Cybersecurity*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 155–175.
106. Sequeira, K.; Zaki, M. Admit: Anomaly-based data mining for intrusions. In Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 386–395.
107. Zhengbing, H.; Zhitang, L.; Junqi, W. A novel network intrusion detection system (nids) based on signatures search of data mining. In Proceedings of the First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008), Adelaide, Australia, 23–24 January 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 10–16.
108. Zaman, M.; Lung, C.H. Evaluation of machine learning techniques for network intrusion detection. In Proceedings of the NOMS 2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–5.
109. Ravipati, R.D.; Abualkibash, M. Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets—A review paper. *Int. J. Comput. Sci. Inf. Technol.* **2019**, *11*, 65–80. [CrossRef]
110. Abrar, I.; Ayub, Z.; Masoodi, F.; Bamhdi, A.M. A machine learning approach for intrusion detection system on NSL-KDD dataset. In Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 10–12 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 919–924.
111. Gao, X.; Shan, C.; Hu, C.; Niu, Z.; Liu, Z. An adaptive ensemble machine learning model for intrusion detection. *IEEE Access* **2019**, *7*, 82512–82521. [CrossRef]

112. Rupa Devi, T.; Badugu, S. A review on network intrusion detection system using machine learning. In *Advances in Decision Sciences, Image Processing, Security and Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 598–607.
113. Kocher, G.; Kumar, G. Performance analysis of machine learning classifiers for intrusion detection using unsw-nb15 dataset. *Comput. Sci. Inf. Technol.* **2020**, 31–40.
114. Kasongo, S.M.; Sun, Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *J. Big Data* **2020**, *7*, 1–20. [CrossRef]
115. Rana, M.S.; Gudla, C.; Sung, A.H. Evaluating machine learning models for Android malware detection: A comparison study. In Proceedings of the 2018 VII International Conference on Network, Communication and Computing, Taipei, Taiwan, 14–16 December 2018; pp. 17–21.
116. Li, C.; Mills, K.; Niu, D.; Zhu, R.; Zhang, H.; Kinawi, H. Android malware detection based on factorization machine. *IEEE Access* **2019**, *7*, 184008–184019. [CrossRef]
117. Raghuraman, C.; Suresh, S.; Shivshankar, S.; Chapaneri, R. Static and dynamic malware analysis using machine learning. In Proceedings of the First International Conference on Sustainable Technologies for Computational Intelligence, Jaipur, India, 29–30 March 2019; Springer: Berlin/Heidelberg, Germany, 2020; pp. 793–806.
118. Singh, M. *User-Centered Spam Detection Using Linear and Non-Linear Machine Learning Models*; University of Victoria : Victoria, BC, Canada, 2019.
119. Islam, M.K.; Al Amin, M.; Islam, M.R.; Mahbub, M.N.I.; Showrov, M.I.H.; Kaushal, C. Spam-Detection with Comparative Analysis and Spamming Words Extractions. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 4–5 June 2020; IEEE: Piscataway, NJ, USA, 2021; pp. 1–9.
120. Şahin, D.Ö.; Demirci, S. Spam Filtering with KNN: Investigation of the Effect of k Value on Classification Performance. In Proceedings of the 2020 28th Signal Processing and Communications Applications Conference (SIU), Gaziantep, Turkey, 5–7 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–4.
121. Sarker, I.H. Context-aware rule learning from smartphone data: Survey, challenges and future directions. *J. Big Data* **2019**, *6*, 1–25. [CrossRef]
122. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 21 June–18 July 1965; Volume 1, pp. 281–297.
123. Ricci, F.; Rokach, L.; Shapira, B. Introduction to recommender systems handbook. In *Recommender Systems Handbook*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–35.
124. Sneath, P.H. The application of computers to taxonomy. *Microbiology* **1957**, *17*, 201–226. [CrossRef] [PubMed]
125. Sorensen, T.A. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *Biol. Skar.* **1948**, *5*, 1–34.
126. Kim, G.; Lee, S.; Kim, S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Syst. Appl.* **2014**, *41*, 1690–1700. [CrossRef]
127. Agrawal, R.; Imieliński, T.; Swami, A. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, 26–28 May 1993; pp. 207–216.
128. Han, J.; Pei, J.; Yin, Y. Mining frequent patterns without candidate generation. *ACM Sigmod Rec.* **2000**, *29*, 1–12. [CrossRef]
129. Flach, P.A.; Lachiche, N. Confirmation-guided discovery of first-order rules with Tertius. *Mach. Learn.* **2001**, *42*, 61–95. [CrossRef]
130. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules. In Proceedings of the 20th International Conference Very Large Data Bases, VLDB, Santiago, Chile, 12–15 September 1994; Volume 1215, pp. 487–499.
131. Houtsma, M.; Swami, A. Set-oriented mining for association rules in relational databases. In Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, 6–10 March 1995; IEEE: Piscataway, NJ, USA, 1995; pp. 25–33.
132. Liu, B.; Hsu, W.; Ma, Y. Integrating classification and association rule mining. *Knowl. Discov. Data Min. Inf.* **1998**, *98*, 80–86.
133. Das, A.; Ng, W.K.; Woon, Y.K. Rapid association rule mining. In Proceedings of the Tenth International Conference on Information and Knowledge Management, Atlanta, GA, USA, 5–10 October 2001; pp. 474–481.
134. Zaki, M.J. Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.* **2000**, *12*, 372–390. [CrossRef]
135. Cannady, J. Artificial neural networks for misuse detection. In Proceedings of the 1998 National Information Systems Security Conference (NISSC'98), Arlington, VA, USA, 5–8 October 1998; pp. 443–456.
136. Lippmann, R.P.; Cunningham, R.K. Improving intrusion detection performance using keyword selection and neural networks. *Comput. Netw.* **2000**, *34*, 597–603. [CrossRef]
137. Li, J.; Qu, Y.; Chao, F.; Shum, H.P.; Ho, E.S.; Yang, L. Machine learning algorithms for network intrusion detection. In *AI in Cybersecurity*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 151–179.
138. Wang, G.; Hao, J.; Ma, J.; Huang, L. A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Syst. Appl.* **2010**, *37*, 6225–6232. [CrossRef]
139. Kayacik, H.G.; Zincir-Heywood, A.N.; Heywood, M.I. A hierarchical SOM-based intrusion detection system. *Eng. Appl. Artif. Intell.* **2007**, *20*, 439–451. [CrossRef]
140. Ding, Y.; Chen, S.; Xu, J. Application of deep belief networks for opcode based malware detection. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 3901–3908.

141. Gao, N.; Gao, L.; Gao, Q.; Wang, H. An intrusion detection model based on deep belief networks. In Proceedings of the 2014 Second International Conference on Advanced Cloud and Big Data, Huangshan, China, 20–22 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 247–252.
142. Tan, Q.S.; Huang, W.; Li, Q. An intrusion detection method based on DBN in ad hoc networks. In Proceedings of the International Conference on Wireless Communication and Sensor Network (WCSN 2015), Changsha, China, 12–13 December 2015; World Scientific: Singapore, 2016; pp. 477–485.
143. Zhu, D.; Jin, H.; Yang, Y.; Wu, D.; Chen, W. DeepFlow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data. In Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 3–6 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 438–443.
144. Alrawashdeh, K.; Goldsmith, S. Optimizing Deep Learning Based Intrusion Detection Systems Defense Against White-Box and Backdoor Adversarial Attacks Through a Genetic Algorithm. In Proceedings of the 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 13–15 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
145. Choudhary, S.; Kesswani, N. Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. *Procedia Comput. Sci.* **2020**, *167*, 1561–1573. [CrossRef]
146. Sai, N.R.; Kumar, G.S.C.; Safali, M.A.; Chandana, B.S. Detection System for the Network Data Security with a profound Deep learning approach. In Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 8–10 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1026–1031.
147. Ahsan, M.; Nygard, K.E. Convolutional Neural Networks with LSTM for Intrusion Detection. *CATA* **2020**, *69*, 69–79.
148. Gurung, S.; Ghose, M.K.; Subedi, A. Deep learning approach on network intrusion detection system using NSL-KDD dataset. *Int. J. Comput. Netw. Inf. Secur.* **2019**, *11*, 8–14. [CrossRef]
149. Ding, Y.; Zhai, Y. Intrusion detection system for NSL-KDD dataset using convolutional neural networks. In Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, Shenzhen, China, 12–14 December 2018; pp. 81–85.
150. Su, T.; Sun, H.; Zhu, J.; Wang, S.; Li, Y. BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. *IEEE Access* **2020**, *8*, 29575–29585. [CrossRef]
151. Jameel, A.S.M.M.; Mohamed, A.P.; Zhang, X.; El Gamal, A. Deep learning for frame error prediction using a DARPA spectrum collaboration challenge (SC2) dataset. *IEEE Netw. Lett.* **2021**, *3*, 133–137. [CrossRef]
152. Nilă, C.; Patriciu, V.; Bica, I. Machine Learning Datasets for Cyber Security Applications. *Secur. Future* **2019**, *3*, 109–112.
153. Zhiqiang, L.; Mohi-Ud-Din, G.; Bing, L.; Jianchao, L.; Ye, Z.; Zhijun, L. Modeling network intrusion detection system using feed-forward neural network using unsw-nb15 dataset. In Proceedings of the 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 12–14 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 299–303.
154. Ahsan, M.; Gomes, R.; Chowdhury, M.; Nygard, K.E. Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector. *J. Cybersecur. Priv.* **2021**, *1*, 199–218. [CrossRef]
155. Al, S.; Dener, M. STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment. *Comput. Secur.* **2021**, *110*, 102435. [CrossRef]
156. Millar, S.; McLaughlin, N.; del Rincon, J.M.; Miller, P. Multi-view deep learning for zero-day Android malware detection. *J. Inf. Secur. Appl.* **2021**, *58*, 102718. [CrossRef]
157. Naway, A.; Li, Y. A review on the use of deep learning in android malware detection. *arXiv* **2018**, arXiv:1812.10360.
158. Pei, X.; Yu, L.; Tian, S. AMalNet: A deep learning framework based on graph convolutional networks for malware detection. *Comput. Secur.* **2020**, *93*, 101792. [CrossRef]
159. Gao, J.; Lanchantin, J.; Soffa, M.L.; Qi, Y. Black-box generation of adversarial text sequences to evade deep learning classifiers. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 50–56.
160. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
161. Sarker, I.H.; Colman, A.; Han, J. Recencyminer: Mining recency-based personalized behavior from contextual smartphone data. *J. Big Data* **2019**, *6*, 1–21. [CrossRef]
162. Massaoudi, M.; Refaat, S.S.; Abu-Rub, H. Intrusion Detection Method Based on SMOTE Transformation for Smart Grid Cybersecurity. In Proceedings of the 2022 3rd International Conference on Smart Grid and Renewable Energy (SGRE), Doha, Qatar, 20–22 March 2022; IEEE: Piscataway, NJ, USA, 2022, pp. 1–6.
163. Ahsan, M.; Gomes, R.; Denton, A. Smote implementation on phishing data to enhance cybersecurity. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018; IEEE: Piscataway, NJ, USA, 2018, pp. 0531–0536.
164. Tsai, C.W.; Lai, C.F.; Chao, H.C.; Vasilakos, A.V. Big data analytics: A survey. *J. Big Data* **2015**, *2*, 1–32.
165. Sarker, I.H.; Abushark, Y.B.; Khan, A.I. Contextpca: Predicting context-aware smartphone apps usage based on machine learning techniques. *Symmetry* **2020**, *12*, 499. [CrossRef]
166. Qiao, L.B.; Zhang, B.F.; Lai, Z.Q.; Su, J.S. Mining of attack models in ids alerts from network backbone by a two-stage clustering method. In Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & Phd Forum, Shanghai, China, 21–25 May 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1263–1269.

167. Wall, M.E.; Rechtsteiner, A.; Rocha, L.M. Singular value decomposition and principal component analysis. In *A Practical Approach to Microarray Data Analysis*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 91–109.
168. Zhao, S.; Leftwich, K.; Owens, M.; Magrone, F.; Schonemann, J.; Anderson, B.; Medhi, D. I-can-mama: Integrated campus network monitoring and management. In Proceedings of the 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow, Poland, 5–9 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–7.
169. Kaufman, S.; Rosset, S.; Perlich, C.; Stitelman, O. Leakage in data mining: Formulation, detection, and avoidance. *ACM Trans. Knowl. Discov. Data TKDD* **2012**, *6*, 1–21. [CrossRef]
170. Nisbet, R.; Elder, J.; Miner, G.D. *Handbook of Statistical Analysis and Data Mining Applications*; Academic Press: Cambridge, MA, USA, 2009.
171. Rosset, S.; Perlich, C.; Świrszcz, G.; Melville, P.; Liu, Y. Medical data mining: Insights from winning two competitions. *Data Min. Knowl. Discov.* **2010**, *20*, 439–468. [CrossRef]
172. Kohavi, R.; Brodley, C.E.; Frasca, B.; Mason, L.; Zheng, Z. KDD-Cup 2000 organizers' report: Peeling the onion. *ACM Sigkdd Explor. Newsl.* **2000**, *2*, 86–93. [CrossRef]
173. Gupta, I.; Mittal, S.; Tiwari, A.; Agarwal, P.; Singh, A.K. TIDF-DLPM: Term and Inverse Document Frequency based Data Leakage Prevention Model. *arXiv* **2022**, arXiv:2203.05367.
174. Stuart, M. Understanding robust and exploratory data analysis. *J. R. Stat. Soc. Ser. D***1984**, *33*, 320–321. [CrossRef]
175. Pulido-Gaytan, L.B.; Tchernykh, A.; Cortés-Mendoza, J.M.; Babenko, M.; Radchenko, G. A Survey on Privacy-Preserving Machine Learning with Fully Homomorphic Encryption. In Proceedings of the Latin American High Performance Computing Conference, Cuenca, Ecuador, 2–4 September 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 115–129.
176. Kjamilji, A.; Savaş, E.; Levi, A. Efficient secure building blocks with application to privacy preserving machine learning algorithms. *IEEE Access* **2021**, *9*, 8324–8353. [CrossRef]
177. Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 1333–1345.
178. Takabi, H.; Hesamifard, E.; Ghasemi, M. Privacy preserving multi-party machine learning with homomorphic encryption. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016.
179. Fang, H.; Qian, Q. Privacy preserving machine learning with homomorphic encryption and federated learning. *Future Internet* **2021**, *13*, 94. [CrossRef]
180. Yang, Y.; Xiao, X.; Cai, X.; Zhang, W. A secure and high visual-quality framework for medical images by contrast-enhancement reversible data hiding and homomorphic encryption. *IEEE Access* **2019**, *7*, 96900–96911. [CrossRef]
181. Salim, M.M.; Kim, I.; Doniyor, U.; Lee, C.; Park, J.H. Homomorphic Encryption Based Privacy-Preservation for IoMT. *Appl. Sci.* **2021**, *11*, 8757. [CrossRef]
182. Bakshi, M.; Last, M. Cryptornn-privacy-preserving recurrent neural networks using homomorphic encryption. In *International Symposium on Cyber Security Cryptography and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 245–253.
183. Guan, Z.; Bian, L.; Shang, T.; Liu, J. When machine learning meets security issues: A survey. In Proceedings of the 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR), Shenyang, China, 24–27 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 158–165.
184. Li, X.; Chen, D.; Li, C.; Wang, L. Secure data aggregation with fully homomorphic encryption in large-scale wireless sensor networks. *Sensors* **2015**, *15*, 15952–15973. [CrossRef] [PubMed]
185. Latif, S.; Dola, F.F.; Afsar, M.; Esha, I.J.; Nandi, D. Investigation of Machine Learning Algorithms for Network Intrusion Detection. *Int. J. Inf. Eng. Electron. Bus.* **2022**, *14*, 1–22.
186. Mavroeidis, V.; Vishi, K.; Zych, M.D.; Jøsang, A. The impact of quantum computing on present cryptography. *arXiv* **2018**, arXiv:1804.00200.
187. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; IEEE: Piscataway, NJ, USA, 1994, pp. 124–134.
188. Bone, S.; Castro, M. *A Brief History of Quantum Computing*; Imperial College in London: London, UK, 1997. Available online: http://www.doc.ic.ac.uk/~{}nd/surprise_97/journal/vol4/spb3 (accessed on 10 May 2022).
189. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.
190. Cerf, N.J.; Levy, M.; Van Assche, G. Quantum distribution of Gaussian keys using squeezed states. *Phys. Rev. A* **2001**, *63*, 052311. [CrossRef]
191. Ding, J.; Yang, B.Y. Multivariate public key cryptography. In *Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 193–241.
192. Hassija, V.; Chamola, V.; Goyal, A.; Kanhere, S.S.; Guizani, N. Forthcoming applications of quantum computing: Peeking into the future. *IET Quantum Commun.* **2020**, *1*, 35–41. [CrossRef]
193. Schuld, M.; Sinayskiy, I.; Petruccione, F. The quest for a quantum neural network. *Quantum Inf. Process.* **2014**, *13*, 2567–2586. [CrossRef]