electronics

MDPI

Article

Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection

Shamshair Ali [1], Saif Ur Rehman [1], Azhar Imran [2], Ghazif Adeem [1], Zafar Iqbal [3] and Ki-Il Kim [4,*]

[1] University Institute of Information Technology (UIIT), Pir Mehar Ali Shah Arid Agriculture University, Rawalpindi 46000, Pakistan
[2] Department of Creative Technologies, Air University, Islamabad 44000, Pakistan
[3] Department of Cyber Security, Air University, Islamabad 44000, Pakistan
[4] Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, Republic of Korea
[*] Correspondence: kikim@cnu.ac.kr

Abstract: Many intrusion detection and prevention systems (IDPS) have been introduced to identify suspicious activities. However, since attackers are exploiting new vulnerabilities in systems and are employing more sophisticated advanced cyber-attacks, these zero-day attacks remain hidden from IDPS in most cases. These features have incentivized many researchers to propose different artificial intelligence-based techniques to prevent, detect, and respond to such advanced attacks. This has also created a new requirement for a comprehensive comparison of the existing schemes in several aspects ; after a thorough study we found that there currently exists no detailed comparative analysis of artificial intelligence-based techniques published in the last five years. Therefore, there is a need for this kind of work to be published, as there are many comparative analyses in other fields of cyber security that are available for readers to review.In this paper, we provide a comprehensive review of the latest and most recent literature, which introduces well-known machine learning and deep learning algorithms and the challenges they face in detecting zero-day attacks. Following these qualitative analyses, we present the comparative evaluation results regarding the highest accuracy, precision, recall, and F1 score compared to different datasets.

Keywords: zero-day attacks; artificial intelligence; machine learning; deep learning; cyber security

check for updates

Citation: Ali, S.; Rehman, S.U.; Imran, A.; Adeem, G.; Iqbal, Z.; Kim, K.-I. Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection. *Electronics* **2022**, *11*, 3934. https://doi.org/10.3390/electronics11233934

Academic Editor: Martin Reisslein

Received: 20 October 2022
Accepted: 25 November 2022
Published: 28 November 2022

1. Introduction

The trend of adopting the Internet and online services has increased exponentially [1]. Technological advances have changed how people work, communicate, and socialize [2]. It has become an integral part of daily life, making it easy to access any information, enabling global communication, and also as a source of entertainment. The primary goal of the Internet is to transmit data from one end of the network (node) to another end (node) over the network. The Internet can be defined as an interconnected network of hundreds of thousands of networks, computers, and associated devices. The transformation, evolution, and invention of modern devices and gadgets such as IoT devices have significantly increased Internet usage worldwide. However, malware can cause disturbance in IoT devices [3].

With all these benefits, there is a scary aspect to the Internet: the privacy and security concerns that most people face on the Internet. As Internet usage and the number of devices are increasing daily, security issues are also dramatically increasing. This is why the Internet has become the playground of cyber criminals [4]. They penetration test the entire network to find the security loopholes and vulnerabilities that exist in a network. Cyberspace continuously intrudes upon, and malicious attacks are made against, any systems or services [5–8]. A system can be secure if it ensures the CIA triangle, which comprises three main components: confidentiality, integrity, and availability.

Privacy and confidentiality are substantially identical concepts. Measures for maintaining confidentiality are intended to guard against unauthorized access to sensitive data. To ensure confidentiality, organizations employ encryption and access control techniques so that data can be accessed and read only by relevant individuals. In contrast, integrity uses various techniques such as hash and cyclic redundancy check (CRC) to verify that nobody has tampered with data at rest or in transit. On the other hand, availability ensures that end systems and networks remain available to legitimate users. Organizations use various techniques to ensure the 24/7 availability of resources, namely fail-safe systems and distributed deployment.

Integrity is the upkeep of data throughout its life cycle in terms of consistency, accuracy, and dependability.

The security and integrity of a system are said to be compromised whenever an illegal activity, destructive program, or unauthorized entity enters a computer or network to harm [9]. There is no universally accepted definition for cyber security, but in the scope of this article, it can be stated that cyber security is a set of different tools, techniques, devices, and approaches that can be used to safeguard cyberspace [10,11].

Based on previous history, signature-based and rule-based algorithms have produced effective results in identifying traditional cyber attacks that indicate discriminate patterns [12–14], most likely known as signatures or fingerprints. Nowadays, cyber attacks are advanced and persistent, and are termed Advanced Persistent Threats (APTs). Such attacks can change memory [15], interact between components [6,16], communicate with command and control [17], activate threads and open files [5], and perform packet routing [17]. Therefore, to cope with such advanced threats, machine learning algorithms help to develop accurate and somewhat precise predictions of future events and circumstances and how to act intelligently in those circumstances [18,19]. In general, machine learning attempts to learn how to produce better and more advantageous circumstances in the future. Figure 1 briefly shows what a zero-day timeline looks like, e.g., the developer uploads software that can be accessed by the public, and hackers find the zero-day vulnerabilities and exploit and leverage them to take control of the system; the developer gets notified about the bugs, fixes it, and creates patches.

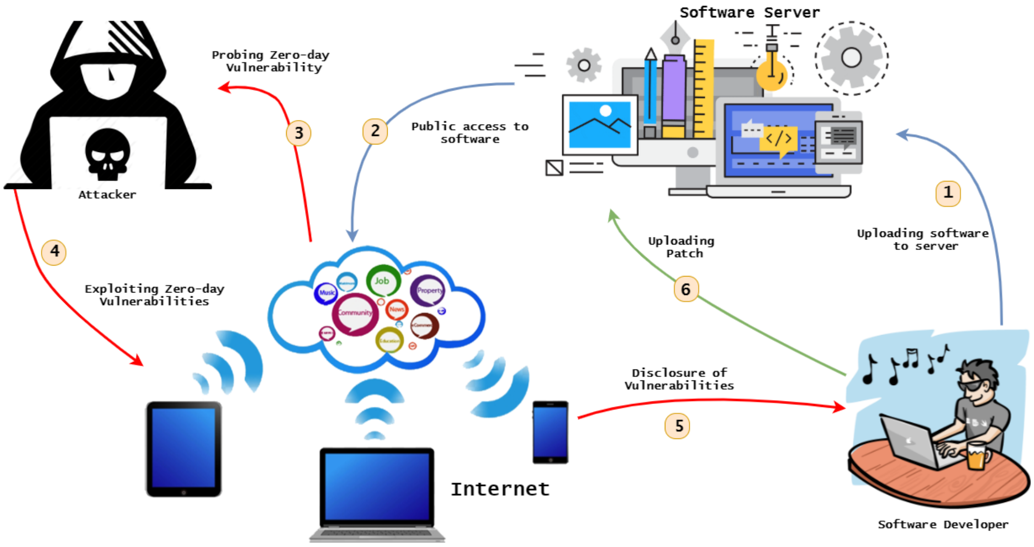
Software Server
Probing Zero-day Vulnerability
3
2
Public access to software
Attacker
1
Uploading Patch
Uploading software to server
6
4
Exploiting Zero-day Vulnerabilities
Job
Property
Community
News
Disclosure of Vulnerabilities
5
Internet
Software Developer

Figure 1. Zero-day real-life scenario.

According to Ref. [20], Deep Learning (DL) comes under the umbrella of machine learning (ML), comprising multiple hidden artificial neural network (ANN) layers. Its methodology includes applying high-level model abstractions and non-linear transformations in large databases. Deep learning (DL) algorithms allow computational models made up of multiple processing layers to learn data representation by using abstraction on multiple levels [21]. Deep Learning (DL) tries to find complex structures in large datasets using a back-propagation algorithm, which indicates when to change or adjust specific internal parameters used for computing every single layer representation from the previous layer. Techniques of DL can be utilized in multiple technical predicaments rather than one usual method in order to save resources and time [22].

A serious threat is posed by zero-day attacks on the security of the Internet, as zero-day vulnerabilities exploit computer systems. A zero-day attack can be described as "a traffic pattern of interest. In general, it has no matching patterns in malware or attack detection elements in the network", as stated by the authors of Ref. [23]. Zero-day attacks of unknown nature (previously not disclosed) are being taken advantage of by attackers, and they use them with other complex attacks to protect themselves from being detected by the intrusion detection techniques, thus making it harder to defend against these kinds of attacks. Zero-day attacks can come in many variations, such as worms (polymorphic), viruses, Trojans, network attacks, and other malware. Blended attacks are attacks that show effectiveness but are not detected, and the worms (polymorphic) are sometimes not detected. This comprises sophisticated mutations for evading target defenses, targeted exploitation to directly attack specific hosts, multiple active, passing, and scanning techniques for detecting vulnerabilities, dropping shells at compromised hosts for a future connection, and other post-exploitation techniques [24]. The schematic for a zero-day attack detection process is given in Figure 2.

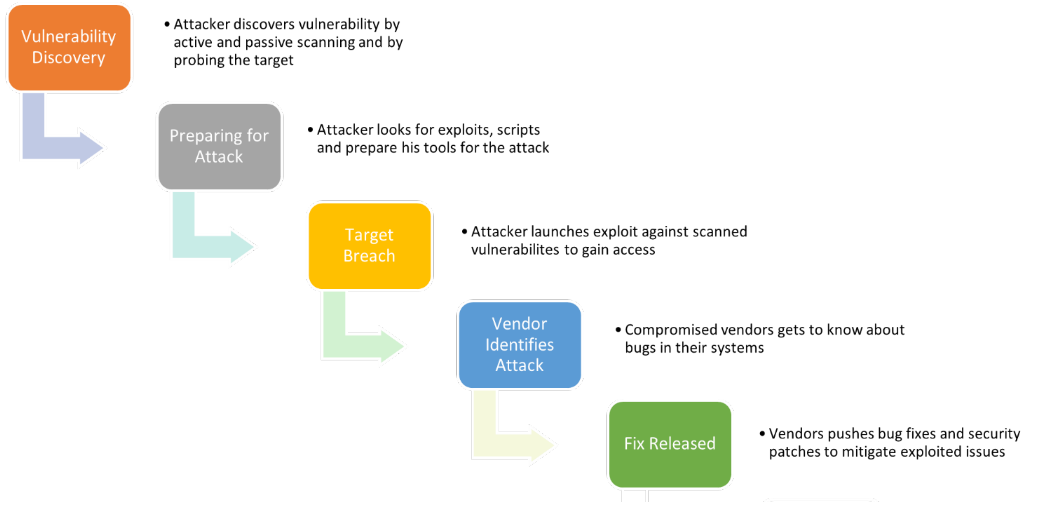
Vulnerability Discovery
• Attacker discovers vulnerability by active and passive scanning and by probing the target
Preparing for Attack
• Attacker looks for exploits, scripts and prepare his tools for the attack
Target Breach
• Attacker launches exploit against scanned vulnerabilites to gain access
Vendor Identifies Attack
• Compromised vendors gets to know about bugs in their systems
Fix Released
• Vendors pushes bug fixes and security patches to mitigate exploited issues

Figure 2. Zero-day attack detection process.

Researchers have also demonstrated that zero-day attacks are often used, but are not apparent, as 11 out of 18 attacks were identified as previously unknown [25]. Their investigation shows that a zero-day attack can be present in a compromised system for an extended period (10 months on average) before the security team can even detect them. The authors of Ref. [26] refer to a statistical study showing that more than 62% of attacks are detected after the system is compromised. Furthermore, zero-day attacks are becoming more and more unpredictable as their number is gradually increasing [27].

According to Ref. [28], anomaly-based techniques study the ordinary and usual network traffic and the entire network flow and try to find anomalies that are not according to the same or usual network traffic pattern. This behavior makes them a good candidate for detecting zero-day attacks. Network traffic and activity vary from user to user, which makes it harder for attackers to know which activities they can perform undetected. Signatures for misuse detectors can be defined upon the data on which anomaly-based techniques throw off alerts. This anomaly-based approach results in high false alarm rates as previously unseen traffic work can be classified as an anomaly.

According to Ref. [29], an attack graph represents previously disclosed knowledge about vulnerabilities, network connectivity, and dependability of vulnerabilities, that is, their dependencies. An attack graph can track vulnerabilities, dependencies, and attack paths without losing any information. The resultant graph has no redundant vertices and a polynomial size, which is calculated by multiplying the number of vulnerabilities and the number of connected host pairs.

The most common and effective way of detecting zero-day attacks is the Intrusion Detection System (IDS). IDS is good at tackling the exponential rise in detecting zero-day attacks, but it still shows a deficiency compared to the previously known attack detection [30,31]. Such attacks either take advantage of a new vulnerability in the system or exploit some previous vulnerability in a new way that cannot be detected in any of the IDS because it does not match with the existing known signatures. The growth of internet devices often exposes the systems to brand-new attacks that cause the growth of hacking activities. In such scenarios, the likelihood of a zero-day attacks becomes higher and higher. To design effective and efficient IDS, ML and DL techniques have been widely used for better performance of the systems [31,32].

This paper focuses specifically on zero-day attack detection techniques using different AI-based algorithms. It gives insights into previously used zero-day attack techniques as well as techniques used in recent years. A brief introduction of techniques with their performance is also given in the final sections. Not much up-to-date comparative research has been done in recent years from the perspective of zero-day attacks, which is why we chose to work on this area so as to show the latest and current techniques.

This paper is structured as follows: Section 2 contains a literature review along with a brief introduction of datasets, models, and algorithms. Section 3 provides a comparative analysis of state-of-the-art studies. In the end, Section 4 covers the limitations of existing studies. Moreover, the conclusion and future directions to mitigate zero-day attacks are provided in Section 5.

2. Literature Review

Zero-day attacks are subdivided into anomaly-based, graph-based, and AI-based attacks, as shown in Figure 3. The detail of these attacks is provided in the subsequent sections.

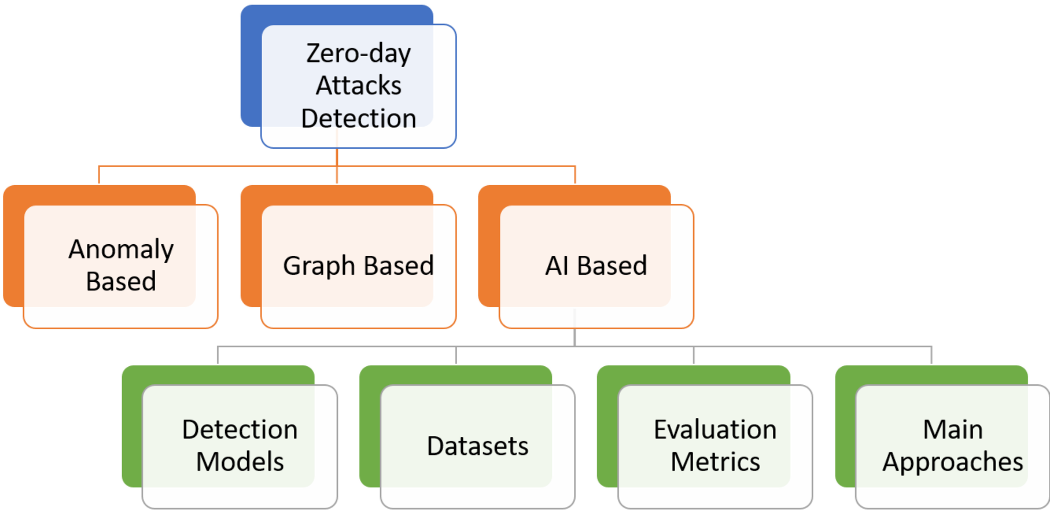

Figure 3. Zero-day attack detection approaches.

2.1. Anomaly-Based

The anomaly-based detection approach in Ref. [33] is represented using regression, i.e., logistics for in-variants that are chaotic, for example, entropy, dimensions of correlation, etc., which are non-linear and intrinsic features. The authors stated that the machine learning algorithms could use these properties, which are highly responsible for generating prominent attributes. The proposed scheme is not suitable, as it is most likely better for anomaly-based attacks that specifically hit buffer overflows, packets' contents, or other associated exploitable vulnerabilities.

Another zero-day detection technique was proposed by Duessel et al. in Ref. [34] to work with the new representation of data known as CN-Gram at the layer of the application level. Syntactic, as well as sequential attributes of payload fusions, were allowed in the combined feature space. The similarity of syntax level attributes alongside mapped byte messages utilizing data representation is calculated after training the algorithm that detects learning normality on a global level. Detection is determined by comparing the message of the model trained and doing a scoring assignment for the extent of the behavior that was found to be anomalous.

Moon et al. [35] proposed a detection system based on hosts for a human-centric computation that is secure. A total of 39 features were defined in 7 categories, i.e., file system, process, registry, thread, etc., to detect if a process running on a host system is malicious. Features taken from the host PC were stored in a separately created database. The decision tree was used to classify if a program was malicious, and a feature vector was used for this purpose, containing features from the database.

An Outlier Dirichlet Mixture (ODM) was proposed by Moustafa et al. in Ref. [36] as a detection system for fog. The authors of Ref. [37] proposed a zero-day polymorphic worm attack framework for detecting polymorphic worms that are zero-day in nature by their behavior, anomaly, and signature-based techniques. Three layers, namely, analysis, detection, and resource, were used in the proposed architecture. The detection engine used healthy and malicious traffic for detecting zero-day attacks. Khan et al. [38] proposed a multilevel anomaly detection model namely Supervisory Control and Data Acquisition Systems (SCADA). The core of the model depends upon the communication between the expected and consistent structure that is in place between devices in the whole setup. To construct the model, the dimensionality reduction technique was used for preprocessing the data; then, the Bloom filter was used to create the signature database. The integration of content-level detection with instance-based learning was done to make the model hybrid in detecting zero-day attacks.

2.2. Graph-Based

Detecting attacks through graphical models has shown improvements compared to behavioral-based or (anomaly-based) attack detection. Different concepts for implementing graphical models have been used, as stated by Refs. [39–41].

The authors of Ref. [42] proposed a detector for anomalies using the probability of network attack occurrences. A directed graph with nodes and edges showing their communication in the entire network can be visualized. First and foremost, a behavioral model for the stochastic attacker was introduced. Afterwards, the detector was used to compare the network probability of the attacker's behavior when he attacks the host under normal conditions and compromises.

For detecting variations in DDoS attacks, an architecture named DaMask was proposed by Wang et al. [43], which updates the model according to new observations based on Bayesian network inference. An attack graph-based zero-day attack detection using layered architecture was presented by Singh et al. in Ref. [44]. The proposed zero-day attack architecture consisted of a risk analyzer, physical, and path generator layers. The centralized server and database of this architecture were used for other layers. An algorithm named AttackRank was proposed to find the exploitation chances in the graph.

A content-based visualization framework for classifying diverse signatures of the worm using a Conjunction of Combinational Motifs (CCM) was devised by Bayoglu et al. in Ref. [41]. Vertices of the graph were taken and considered as the worms' invariant parts. The CCM automatically generated and detected signatures of unseen worms (polymorphic). For the discovery of an effective attack path using a compact graph, planning was suggested by Yichao et al. in Ref. [42]; the neighbor is involved in the three main steps of the proposed solution: the closure and formalism calculation, the construction of the graph, and the extraction of the attack path.

The basic building block of the approaches and techniques reviewed within this subsection is the graph, in which the edges and nodes are treated differently according to the different implementations. For example, in a few works, the nodes are taken and considered as hosts, and communication among the nodes is considered as edges. Alternatively, some researchers consider nodes as file structure instances and edges as ordinal relationships. The likelihood of a node being malicious or not is based upon some external evidence for some approaches. On the other hand, the likelihood of an attack occurrence is shown by the comparing network traffic and its flow under attack and normal conditions. Overall, the graph is used for signature extraction in all these approaches by applying specific criteria for detecting zero-day attacks.

2.3. AI-Based Solutions

A brief introduction of various AI models, different datasets, evaluation metrics, and various research work carried out in zero-day attack detection is provided in the following subsections.

2.3.1. Detection Models

This section briefly introduces various AI-based detection models used in zero-day attack detection.

Decision Trees

Decision Trees are a supervised non-parametric machine learning approach based on a recursive tree structure that may be used for regression and classification. The goal is to determine the values of target variables by learning basic decision rules abstracted from data attributes by creating a tree-like model. Unlike many other strategies, this approach does not need extensive data preprocessing. Furthermore, because the trees can be viewed, it is simple to comprehend and interpret. A root or intermediate node, a route, and a leaf node are the three components of a decision tree. An object/attribute is represented by a tree's root/intermediate node. Each divergence route in the tree indicates the parent node's potential values (object). The leaf node represents the predicted category/classified attribute. If-then rules are used to depict the resulting tree further. Entropy and information gain measurements are employed to identify the best potential intermediate node throughout the tree-building process [45].

Random Forests

Random forest is a simple supervised learning algorithm that, in most circumstances, produces excellent results even without hyper-parameter adjustments. It is one of the most extensively used models due to its simplicity and adaptability. It may be applied to both classification and regression. It produces a "forest" through a collection of decision trees that are frequently trained by utilizing the "bagging" method [46].

Multilayer Perceptron

The multilayer perceptron (MLP) is a technique for approximating any continuous function and dealing with non-linearly separable situations. MLP's typical uses are pattern classification, approximation, recognition, and prediction. MLP has three layers: one input layer, one output layer, and one or more hidden layers. The input layer receives the input signal. The output layer handles tasks such as categorization and prediction. Lastly, the hidden layers, located between the input and output layers, relate to the MLP algorithm's underlying computational engine [47].

Bloom Filter Model

Bloom Filter is a space-efficient probabilistic model used to check whether an object belongs to some specific set; it creates a database of signatures of anomalies and predicts the abnormal traffic based on the stored signatures [48]. It was invented by Bloom in 1970 [49] and has been widely used in various sub-domains of the IT sector, including by Google in their search engine, because it is much faster and lightweight [50]. A Bloom Filter can be implemented in both hardware and software.

Long Short-Term Memory

Long short-term memory (LSTM) [51] is a deep learning architecture based on artificial RNNs (Recurrent Neural Networks). Unlike traditional feed-forward neural networks, the latter has feedback connections. It can handle individual data points (such as photographs) and whole data sequences (such as videos). In regards to voice recognition, linked handwriting identification, unsegmented intrusion detection system (IDS), and anomaly detection can all take advantage of LSTM [52,53].

Hybrid Multilevel Anomaly Detection-IDS

HML-IDS is a hybrid multilevel anomaly detection approach that uses the Bloom Filters technique to check whether an item is part of some specific group. After that, it stores the signature from the network traffic and identifies malicious traffic (attacks). It is explicitly designed to detect previously unknown (zero-day) attacks on SCADA systems [42].

Kalman Filter

To solve the discrete-data linear filtering problem, R.E Kalman presented his famous statistical model in 1960, which is known as the 'Kalman Filter'; in his famous paper, he proposed a recursive solution to the problem mentioned above. It is a set of mathematical expressions that provides a robust computational (recursive) solution of the least-squares method. It has played a vital role in digital computing, mainly autonomous or assisted navigation. KF is very powerful in different aspects due to its ability to estimate present, past, and future states even when the distinctiveness of a modeled system is not known [54].

Zero-Shot Learning

Learning to perceive new ideas with just a description is known as zero-shot learning. A range of strategies has been offered to overcome the obstacles created by this topic. In this research, we propose a zero-shot learning strategy that requires only one line of code to implement and outperforms state-of-the-art algorithms on standard datasets. The method is built on a more general framework representing the interactions between characteristics, attributes, and classes as a two-layer network, with the top layer's weights determined by the environment rather than being learned. By recasting these approaches as domain adaptation methods, we also offer a learning constraint on their generalization error [55].

Support Vector Data Description

The Support Vector Classifier inspires Support Vector Data Description (SVDD), proposed to solve the multidimensional outlier detection problem. By avoiding the estimation of data density by probabilistic methods, it obtains a spherical boundary around the dataset described by a few training objects (support vectors) that can be made more versatile and flexible by applying various kernel functions similar to the support vector classifier. Using both actual and simulated data, they demonstrated the features of support vector data descriptions. They made this strategy robust to withstand outliers in the training dataset and help it tighten the description by using negative examples [56].

Autoencoder

Autoencoder is an unsupervised deep learning technique. It deconstructs, compresses, and removes noise from the data, which minimizes the size and dimensions of the input data. The reconstruction procedure can also restore the input's original form. The autoencoder assumes that the desired output values should be the same as the original input values. An autoencoder is made up of four primary components. An encoder is employed to learn how the data is compressed. Afterwards, it uses a bottleneck to hold the fully compressed data. Furthermore, the model learns how to reconstruct data by employing the decoder. Finally, after reconstruction, it measures the reconstruction loss and determines how much the output is near the desired output values [57].

Convolutional Neural Networks

Convolutional neural networks (CNNs) are multilayer neural networks that are an extension of feed-forward artificial neural networks (ANNs) [58]. Convolution is a linear mathematical action between matrices, which is how CNN got its name. The convolutional layer, non-linearity layer, pooling layer, and fully connected layer are some of the layers of CNN. The pooling and non-linearity layers do not have parameters, but the convolutional and fully connected layers have. In ML problems, CNN performs admirably. Awe-inspiring results were obtained where the applications deal with picture data, such as the world's most extensive image classification data collection (Image Net), computer vision, natural language processing (NLP) [59], drug discovery [60], and anomaly detection [61,62]. It retrieves higher-resolution features and transforms them into complex features as the resolution decreases. CNN architectures such as ZFNet [63], GoogLeNet [64], and ResNet [65] are widely employed.

Reinforcement Learning

Reinforcement learning (RL) is sometimes referred to as learning with criticism because there is always feedback to the algorithms against any incorrect prediction; it is a machine learning subdomain. However, the algorithm needs to be informed of how to solve it. Instead, the algorithm must consider and test various options until it discovers the true solution [66]. This phenomenon is based on a system of rewards and penalties. RL is the closest approach to simulating the human way of thinking by applying an unfamiliar and novel setting. The five components of RL's operation are the agent, environment, state, reward, and action. Through direct interaction with the environment, an agent creates its own learning experiences. As a result of this interaction, two modifications have happened. To begin, the environment is transformed into a new state. Secondly, the environment imposes a reward or penalty based on the activity. The reward function shows the agent how excellently or poorly an action was done in a given state. The agent gains knowledge from the reward and filters out the undesired behaviors. To address many complicated issues, deep learning approaches and RL are integrated. AlphaGo is an example of this strategy [67,68]. In cyber security, deep reinforcement learning is utilized for intrusion detection on hosts [69], fighting DDoS assaults [70], detecting phishing emails [71], and cyber-physical systems [72], to mention a few applications.

Deep Neural Network

A Deep Neural Network (DNN) is a feed-forward network comprised of three layers: input, hidden, and output layer; it contains multiple hidden layers between its input and output. Each layer of DNN consists of multiple nodes connected hierarchically to all the nodes in the subsequent layer. After feeding the features to the input layer, it does its processing in hidden layers and comes up with predicted values from the output layer. The model calculates the weighted sum and derives the predicted values from an activation function already defined in each hidden layer node, which receives a weighted sum of input nodes and converts them into valid results [73].

Transferred Deep-Convolutional Generative Adversarial Network (tDCGAN)

The authors proposed a new model called transferred deep-convolutional generative adversarial network (tDCGAN) to detect zero-day malware efficiently. It is based on Deep

AutoEncoder (DAE), and generative adversarial network (GAN) [36]. It learns different features of actual data and modified data generated by the model; then, by modifying those features, it generates some fake malware and learns to distinguish this fake malware from the real ones.

WAVED

In a recent study, the authors designed a model that constituted a weighted ensemble of distinct classifiers termed 'WAVED.' It is meant to detect anomalies in multi-sensor data streams by assigning unique voting weights to each classifier's prediction to detect abnormal behavior using the ideal weight vector of classifiers. To detect abnormalities in CAVs and benchmark the performance of the MSALSTM-CNN approach, it uses the average predicted probability of several classifiers [74].

2.3.2. Datasets

The ICS Gas Pipeline dataset was provided by the State University of Mississippi's in-house SCADA lab. For the performance analysis of the IDS, 35 cyber-attacks were present in the dataset that was used for training and testing the classifiers used by IDS. Both regular and cyber outbreak data were present in the dataset. The dataset contained almost 60,048 attacked packets, and 214,580 were regular network packets [38].

The Information Security and Object Technology (ISOT) dataset contains multiple botnets and various standard datasets, and it has a total of 1,675,424 instances of network traffic flow. For botnets, multiple honeypots such as Storm Botnet and Waledac Botnet were used for capturing malicious botnet traffic. Data from Traffic Lab Ericson was also gathered, which was non-malicious. This whole data and traffic flow was combined with another dataset created by Lawrence Berkeley National Lab (LBNL) [75].

The Information Security Centre of Excellence (ISCX) dataset was generated through SSH, HTTP, and SMTP traffic, and it contained a diverse intrusion scenario ranging in variety, complexity, and scope [74]. The ISCX-URL-2016 dataset contained multiple URLs in a dataset ranging from benign (35,300), spam (12,000), phishing (10,000), malware URLs (11,500+), and defacement URLs (45,540) as reported by the Canadian Institute for Cyber security.

The NSL KDD dataset was an improved version of the previously named KDD or KDD'99 dataset. It has some advantages over KDD, such as no redundant records being present in the training and testing sets. It contained data for Denial of Service (DoS), User to Root (U2R), Remote to Local Attack (R2L), and probing attacks. It was built on KDD, built by DARPA's 4 GB of compressed tcpdump data of 7 weeks of network traffic with about 5 million connection records [66]. The dataset can be found in JSON and CSV files [76].

Cloud Intrusion Detection Dataset (CIDD) considers network and host audit data. Data is correlated with the IP address of the user and audit times. CIDD has two main parts: the collection of Unix Solari's audits and Windows NT audit and their respective TCP data dump. CIDD has 35 days of Solaris audit for training and TCP data dump alongside labeled attacks for training any IDS. For testing, CIDD has 10 days of Solari's audits, and TCP data dumps [77].

The CICIDS2017 dataset was created by the Canadian Institute for Cyber security (CIC). It is a recording of 5 days of traffic capture for insider and outsider attacks and benign traffic. Data of a few attacks such as SSH and FTP brute-forcing, DoS/DDoS and Heartbleed, Brute-force, XSS, SQL injection and Infiltration, and BotNet was captured. The total data was almost 51 GB, including regular and attack activity [78]. The various datasets used in this study are graphically presented in Figure 4.

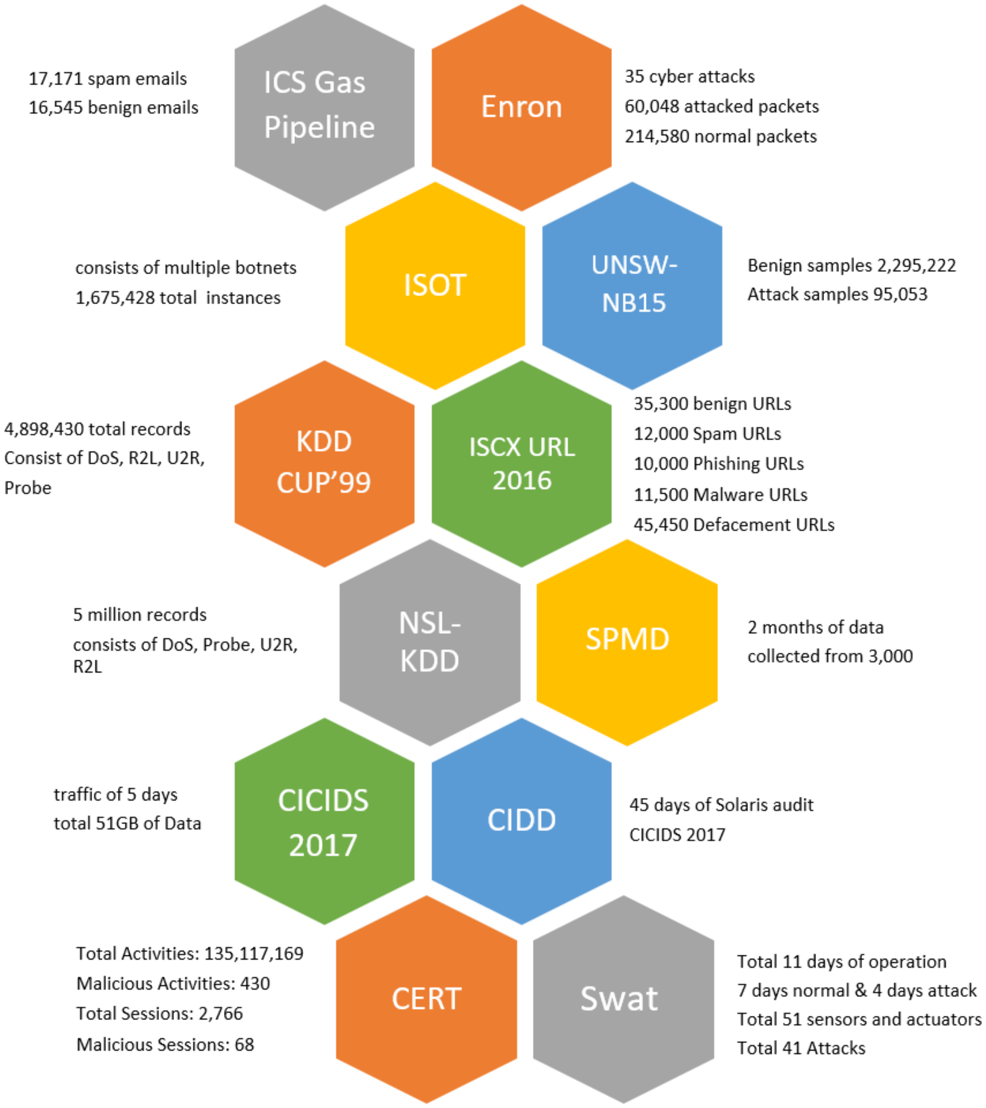

Figure 4. Datasets used in the comparative analysis.

The Safety Pilot Model Deployment dataset contained speed, GPS speed, and acceleration of vehicles. A researcher named Wyk added multiple simulations, such as instant, constant, gradual drift, and bias. These anomalies can cause malfunctioning vehicular services, i.e., speed and acceleration, due to cyberattacks and faults [74].

The Enron Spam dataset was collected by V. Metsis, I. Androutsopoulos, and G. Paliouras. It contained 17,171 spam emails and 16,545 ham (non-spam) emails, totaling 33,716 emails in the dataset. The emails contain a subject, message, date of arrival, and their categorization (either spam or ham). This dataset is publicly available [76].

UNSW—NB15 is basiSolari's network intrusion dataset, which has plenty of attacks such as DoS, worms, backdoors, and fuzzes. It has 49 features that were extracted by Bro-IDS and Argus tools and 12 other additional SQL algorithms. The benign samples count in the dataset is 2,218,761, while 321,283 were attack samples, making a total of 2,540,044 data samples of network traffic. NF-UNSW-NB15 dataset was generated by 43 net flow-based features from pcap files of the UNSW-NB15 dataset. The benign samples count in this dataset is 2,295,222, while the attack sample count is 95,053, making a total of 2,390,725 network traffic flow samples [77].

Kaggle: Microsoft Malware Classification Challenge was a competition conducted in which malware datasets consisting of malware or virus' samples such as Ramnit (1539 data, 145 test data), Lollipop (2459, 234), and Obfuscator. ACY (1221, 132) and Gatak (1011, 106) were used, and GAN generated other fake malware samples by the researchers [36].

Secure Water Treatment (SWaT) Testbed: To tackle the cyber security threats to Industrial Control Systems (ICS), the iTrust team designed the SWaT dataset to enable experimental research. They studied possible vectors for attacks against industrial facilities and developed field-proven recommendations for critical infrastructure protection. It contains 11 days of continuous operation: 7 under regular operation and 4 days with attack scenarios. Fifty-one sensors and actuators were used in the test to collect data from network traffic and recorded 41 attacks.

2.3.3. Evaluation Metrics

Before explaining the metrics, we will briefly introduce the confusion matrix. A confusion matrix is generally a 2 × 2 matrix layout used to visualize an algorithm's performance. The actual performance measures that must or should be met are written vertically, while the predicted ones by the algorithm are written horizontally. A 2 × 2 confusion matrix is shown in a tabular format in Table 1.

Table 1. A 2 × 2 confusion matrix.

		Actual	
		True	False
Predicted	True	True Positive	False Positive
	False	False Negative	True Negative

- True Positive (TP) is the total positive instances identified as positive.

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

- True Negative (TN) is the number of negative instances identified as negative.

$$TNR = \frac{TN}{TN + FP} \tag{2}$$

- False Positive (FP) is defined as the number of negative instances classified or predicted as positive.

$$FPR = \frac{FP}{FP + TN} \tag{3}$$

- False Negative (FN) is the number of positive instances classified or predicted as negative.

$$FNR = \frac{FN}{FN + TP} \tag{4}$$

- Accuracy is the ratio between the number of correct predictions and a total number of predictions.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

- *Precision*: is defined as the ratio between TPs combined with several TPs and FPs. It is the percentage of correctly identified positives out of all the results that were said to be positive either correctly or not.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

- *Recall*: the ratio between TPs combined to several TPs and FNs. It is the percentage of correctly identified positives out of all actual positives, either correctly or not.

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

- *F1-score*: It takes both false negatives and false positives into consideration, and is the harmonic mean of recall and precision. It performs well on imbalanced datasets.

$$F1\text{-}Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{8}$$

2.3.4. Main Approaches

This section touches upon some of the main approaches followed by researchers that employ Deep Neural Network (DNN) and Machine Learning (ML) techniques to develop their framework. Table 2 provides an overview of various state-of-the-art ML- and DL-based approaches for zero-day attack detection.

Table 2. Analysis of zero-day attack detection techniques.

Paper	Year	Methodology	Summary
[79]	2020	Deep Neural Network Approach	Optimization and dimensionality reduction were done using Grey Wolf Optimizer (GWO) and PCA. Dimensionality reduction technique resulted in 15% increased accuracy.
[80]	2020	Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM)	Abnormality detection through the applied voting scheme on automotive generated data using various classifiers for final decision.
[81]	2020	Port Analysis using Statistical Approach	Port uses detection based upon the profile used by the port. Host traffic collection in a distributed environment. High-volume attacks focused. Attacks that were low volume were not covered.
[82]	2020	Deep Neural Network	Auto-encoder with deep neural network architecture was presented. Performance was analyzed on NSL-KDD and CICIDS2017 datasets.
[83]	2020	Deep Neural Network	Manifold alignment for the unification of feature space. Soft labeling was employed. Zero-day detection was done using a high-volume attacks training phase. CIDD and NSL-KDD resulted in poor zero-day detection performance.
[74]	2020	Detection using Reinforcement Learning	CART algorithm utilized for model features selection. Network traffic (real-time) was used for evaluation.
[84]	2019	Deep Learning	Dynamic, static, and image-based analysis was conducted for malware detection. Researchers tested executable binaries of malware. The technique was Hosts oriented. For ZAs detection, work and study at the kernel level were also done.
[79]	2019	Snort Intrusion Detection System using Hybrid Approach	Assignment of exploitation likelihood was done using Ranking Algorithm based on frequency. High-volume attack-focused time stamp-based attack graphs were built. The main focus was on high-volume attacks. Low-volume attacks were ignored.
[39]	2019	K-Nearest Neighbor (KNN) and Bloom Filter based Hybrid approach	KNN and Bloom filter was used for capturing and analyzing network traffic. High false positives resulted from the Bloom filter. The anomaly-based approach was employed for high and low-volume zero-day attack detection.
[78]	2019	Stats Model	Disclosing relation of exploit and vulnerability was the main focus. Copula functions such as student-t and Gaussian were used.
[33]	2018	Generative Adversarial Network utilizing autoencoder	Detection of ZAs works by leveraging noise addition in existing malware. ZA detection of fixed-length malware was focused.
[40]	2018	Bayes Network	Nodes (hosts) of graph consisted of file instances, and edges were communication between nodes. Accurate evidence availability was a factor for performance. Host-based technique.
[35]	2017	Support Vector Machine (one-class)	Sequential features within protocol context were combined and focused upon. Only application layer attacks were considered.

The authors presented transfer learning [85] for detecting time series anomalies problems by changing the weights of the source instances to check and match against target instances. In actuality, nearest neighbor classification was employed for anomaly detection based on some of the labeled instances of the source. The authors of Ref. [86] focused on unifying homogeneous feature spaces. The domains' orthogonal transformation leveraging

Principal Component Analysis (PCA) was done. Afterwards, they opted for the K-nearest neighbors classification technique on that transformed space to detect zero-day attacks.

A novel approach known as "Transferred Deep-Convolutional Generative Adversarial Network" (tDCGAN) to detect actual malware from the fake malware that was generated by this approach itself was proposed [87]. This approach contains a detector that learns features of actual malware and generated malware by utilizing a deep autoencoder, by which GAN training is also stabilized, which is then used for detecting attacks of zero-day malware. For zero-day detection, the authors claimed that their model was the most robust.

The Cyber Resilience Recovery Model (CRRM) was proposed by Tran et al. [88], which handles attacks on closed networks. The NIST SP 800-61, an incident response framework for resilience and standard, is joined with the Susceptible Infected-Quarantined-Recovered (SIQR) model in Ref. [89] to capture zero-day attack and recovery. A deep learning identification technique utilizing an IoT environment's fog-based ecosystem has been proposed in Ref. [90]. As smart infrastructures and fog network has closeness, the nodes of the fog ecosystem are responsible for model training and performing the detection of the attack. The trained model results in models for detecting attacks and native associated learning parameters used by the fog network nodes for propagation and global update.

In Ref. [91], an approach of detection based upon Artificial Neural Network (ANN) was proposed by Saied et al. for unknown and known DDoS attacks depending upon particular attributes that can separate Distributed Denial of Service attacks from authentic traffic. This model was then trained with the help of Java Neural Network Simulator (JNNS) upon preprocessing data, and Snort AI was integrated within.

The authors employed the Gated Recurrent Unit (GRU) technique. Its primary purpose was to pick up new Distributed Denial of Service (DDoS) attacks. The authors claimed that the proposal shows better accuracy [92]. A recent study [93] implemented an approach to resolve issues of security in-vehicle communication that are possibly open to plenty of attacks. It is a hybrid technique based on GRU and CNN for detecting possible attacks.

A signature extraction technique based upon the concept of heavy hitters for high-volume zero-day attacks was proposed by Afek et al., who presented the Metwally's heavy hitter algorithm with a slight variation and modification. It generated all possible K-Grams of the sets from input data. The main concern behind detecting zero-day DDoS attacks is looking for the heavy hitters in real attack data. Then, depending upon the predefined threshold, the heavy hitters' output is checked and placed in different categories depending on their malicious extent. Finally, they can find and detect the attacks by going through the probable malicious heavy hitters [94].

Kim et al. proposed a malware detection system based on deep learning called Transferred Deep Convolutional Generative Adversarial Networks (tDCGAN). A decoder was used to learn the characteristics of malware, the deep auto-enc, and generate new malware samples. The newly generated data is then sent to the adversarial network generator. There were three main parts of the proposed system; compression and reconstruction of data, generation of fake malware data, and finally, malware detection [87].

The authors proposed a Deep Neural Network (DNN) approach for detecting cyber-attacks. It combines techniques such as PCA and the GWO algorithm, where the responsibility of PCA is to reduce the dimensions of the dataset [95]. Then GWO is used for optimizing the transformed dataset for redundancy reduction in the transformed dataset [95]. The main focus of this approach is the dimensionality reduction for making DNN-based IDS more responsive [80,96].

The authors proposed a framework for efficiently handling historical attack data volatility and multi-variate dependency among the attacks. Its main goal was to disclose trends regarding various vulnerabilities and their dependence and exploits on volatile historical data using copula. Two functions, Student-T and Gaussian, were used as the copula function. Using these functions gives us the attack risks joint property [78].

A multistage attention mechanism alongside CNN that is LSTM-based was proposed for anomaly detection [81]. Data abnormality generated through various sensors in automated vehicles is covered explicitly within the proposed method. An ensemble approach based on the voting technique for deciding anomalous data from different classifiers was also proposed.

The authors addressed the problem of detecting zero-day attacks because of lacking labeled data [74]. An approach, namely "Manifold Alignment," was used, which works by mapping the domain data of the target and source into the same latent space. It helps take care of various feature spaces and domain probability distribution. The generated space is also dealt with as a newly proposed technique that generates labels that are too soft to deal with the short number of labels utilized to construct the Deep Neural Network model for zero-day attack detection.

Another technique for intrusion detection and prevention system using classification and a one-time signature technique was proposed for the cloud [75] by the authors. The proposed technique OTS is different from OTP, which stands for one-time password, whereas OTS is used for accessing the data over the cloud. Hybrid classification consisting of normalized K-Means and Recurrent Neural Network (RNN) was used. An approach based on a deep autoencoder was proposed to detect zero-day attacks [82]. Its performance was demonstrated by using two popular and well-known datasets, NSL-KDD and CICIDS2017, and the performance was compared against the one-class SVM outlier detection.

3. Comparative Analysis

This section provides a comparative analysis of AI-based techniques for detecting zero-day attacks and their performance evaluation.

Machine learning and deep learning approaches are widely used to deal with cyber-attack detection and prevention. In the last few years, many researchers have applied ML and DL techniques to classify and detect zero-day attacks from systems, including malware, malicious URLs, and spam. Most of the zero-day malware is not detected by antiviruses, which is why they are problematic; in Ref. [36], the authors proposed a zero-day malware detection framework based on Deep Autoencoder (DAE). It generates fake malware and trains the model to distinguish between real and fake malware by comparing the fake data with the actual data. It learns different malware features from both real data and fake generated data using the proposed model (tDCGAN). It extracts the appropriate features from the data and stabilizes the training. The trained model used transfer learning to capture malware features with an average classification accuracy of 95.74%.

A hybrid model was used, which took patterns from communication that was consistent and anticipated from multiple devices of Industrial Control Systems (ICS). The ICS dataset was used, which was then improved (pre-processed) by using normalization, standardization, labeling (categorical) of the samples, and feature scaling. Then, features were extracted from the dataset, and the feature matrix was constructed using PCA, CCA, and ICA. The proposed technique, HML-IDS, was based on an instance-based k-NN learning algorithm. It was trained on the reduced features by reduction techniques, and then the results were compared with Bloom Filter (BF) and Random Forest (RF). The proposed technique showed better results as it achieved 0.97, 0.98, 0.92, and 0.95 in accuracy, precision, recall, and F1-score, respectively. The main question here is why deep learning techniques were not used. DL techniques improve accuracy, precision, recall, and F1 score [42].

A recent study [74] for detecting anomalies in connected automated vehicles (CAVs) proposed a multistage attention mechanism with an LSTM-based convolutional neural network model named MSALSTM-CNN for detecting different anomalies caused by faults, errors, or that were perhaps due to cyber-attacks; whatever the cause, these events can results in fatal accidents, and can therefore not be compromised in any way. MSALSTM-CNN converts data streams into multidimensional vectors and then processes them to detect anomalies. Another method the study introduced works on the principle of average predicted probability of multiple classifiers for anomaly detection, a weight-adjusted fine-tuned ensemble, or WAVED. They effectively improved the anomaly detection rate in both low- and high-magnitude cases of anomalous instances by gaining 2.54% in F1-score to detect single anomaly types in the dataset. Moreover, it showed promising performance with a gain of up to 3.24% in the F1-score in detecting mixed anomaly types in CAVs.

Regarding an autoencoder-based framework for the detection of zero-day attacks, the authors in Ref. [74] used the NSL-KDD and CICIDS2017 datasets to perform the tests; they compared their model with one class SVM on both the datasets and outperformed it by achieving a higher average accuracy of 92.96% for NSL-KDD and 95.19% for CICIDS2017.

Sameera et al. [88] proposed a zero-day attack detection model for IDS systems; they applied DNN to build their model. It was difficult for them to detect and identify zero-day attacks because there were no labels in the Intrusion detection systems, so they used the Manifold alignment method to map sources and targets using their mapping functions and assign soft labels, and then applied DNN to identify zero-day attacks from the test data. They used NSL-KDD and CIDD datasets for testing. By comparing their results with other ML Models, i.e., DT, RF, SVM, and KNN, they found that their proposed framework outperformed the other models and achieved an average accuracy of 91.83%.

The authors discussed and presented botnet detection using a reinforcement-based learning approach. The whole workflow is that the network traffic captured was filtered down and reduced by control filters, feature extraction from connections and hosts was done, and then reduced by the CART algorithm. Then, those features were passed into a multilayer neural network classifier while dividing features and data into training and testing and, in the end, classifying them into anomalies or not. The resultant detection accuracy is 98.3% with a lower false-positive rate of 0.012% [89].

Instead of targeting a system, hackers mostly try to target humans because they are easy targets. Fooling people by using their weaknesses is an ancient but helpful technique often used by attackers to gain access to a system; they somehow persuade the victim to unknowingly provide sensitive credentials to the attacker. Attackers send phishing links embedded in emails or text messages to the users. Once the person clicks the link inside the mail, their system gets affected by the malicious software sent by the attackers. A similar scenario is used by Ref. [51], where they tried to classify malicious emails from benign emails. A comparison of various AI-based techniques for zero-day attack detection is presented in Table 3.

Theyinvestigated the email header and body using a deep-learning approach and extracted features to test the abnormal score against it. They detected both zero-day malicious emails and regular spam emails, and achieved 78% accuracy for zero-day malicious email detection. It is difficult to obtain data samples for all attack classes in Network Intrusion Detection Systems (NIDS) to observe the traffic in production. Machine learning-based NIDSs face unknown attack traffic known as zero-day attacks that are not used in training because they did not exist during the time of training. The authors proposed a Zero-shot Learning ZSL technique to detect zero-day attacks in NIDS to evaluate the performance of the proposed model. It maps the features of unknown attacks from the network to differentiate its attributes from known attacks. They defined a new zero-day detection rate metric to measure the model's effectiveness [52].

Table 3. A comparison of different AI-based techniques for zero-day attack detection.

Attacks	Ref	Year	Datasets	Approach	Accuracy	Precision	Recall	F1 Score
IDS	[42]	2019	ICS	HML-IDS	97	98	92	95
	[42]	2019	ICS	Bloom Filter	89	97	67	78
	[42]	2019	ICS	RF	91	93	81	86
	[97]	2017	ICS	LSTM	92	94	78	85
	[97]	2017	ICS	SVDD	76	95	21	34
	[97]	2017	ICS	Bloom Filter	87	97	59	73
	[98]	2021	ICS	BLOSOM				
	[98]	2021	ICS	MLP	95	96	90	
	[99]	2018	ICS	CNN	97.85	98.8	83	
	[100]	2021	ICS					
Phishing	[89]	2020	ISOT, ISCX	Reinforcement learning	98.3		97.9	98.8
	[101]	2021	ISCX	Stacker	98.8	90.3	94.3	92.3
	[102]	2021	ISCX	Logic-Integrated Triplet Network	97.85		96.10	
Insider Threat Detection	[103]	2017	CERT	Unsupervised KNN	54	47.5	44.2	44.9
	[104]	2018	CERT	Hidden Markov Model	71.1	64.1	55.9	61.7
	[105]	2021	CERT	SVM	70	40	11	60
	[105]	2021	CERT	LSTM	75	20	59	30
	[105]	2021	CERT	DNN	86	36	73	48
	[105]	2021	CERT	MITD	92	54	54	55
	[105]	2021	CERT	HITD	97	77	92	84
	[78]	2020	NSL KDD	Autoencoder	92.96			
	[101]	2021	NSL KDD	Stacker	99.39	99.7	99	99.3
DoS/DDoS	[78]	2020	CICIDS 2017	Autoencoder	95.19			
	[52]	2021	UNSW-NB15	ZSL-RF	99.71		96.85	97
	[52]	2021	UNSW-NB15	ZSL-MLP	99.55		96.53	95
	[101]	2021	CICIDS 2017	Stacker	99.97	99.8	100	99.9
	[106]	2018	KDD CUP 99, NSL KDD	Autoencoder	86.96	88.65		
Anomaly-based	[105]	2021	CERT	AITD	90	49	50	49
	[73]	2021	SWaT	CNN	92	88	98	92
	[73]	2021	SWaT	DBN	80	72	72	83
	[73]	2021	SWaT	PCA+CNN	95	94	97	95
	[73]	2021	SWaT	PCA+DBN	91	88	95	91
	[73]	2021	SWaT	BLOSOM	96	96	98	96
	[73]	2020	SPMD	MSALSTM-CNN	96.56	99.06		97.37
	[73]	2020	SPMD	WAVED	94.87	98.87		95.44
	[97]	2019	SPMD	KF	97.4	94.5		91.7
	[97]	2019	SPMD	CNN	98.0	99.8		96.4
	[97]	2019	SPMD	CNN-KF	98.2	99.5		96.8
Spam-based	[51]	2020	Enron Spam Dataset	DL-based feature extraction	92.86	91.27	92.86	
	[101]	2021	UNSW-NB15	Stacker	97.19	98.2	94.6	96.4
Malware-based	[36]	2018	Malware Dataset	tDCGAN	95.74	94.4	91.5	92.4
	[98]	2021	Network Dataset	DT	-	100	98	99

Researchers developed a benign communication database from multiple devices in ICS by observing the communication patterns of a system for some time with the use of a Bloom Filter. They then created another SCADA gas pipeline dataset. Afterwards, they proposed a stacked LSTM for time-series anomaly detection by combining both datasets and the Bloom filter. The results were much better than the state-of-the-art techniques, as claimed by the researchers of the study. The framework outcomes were then compared with SVDD, BN, and BF. The proposed work's results were 0.94, 0.78, 0.92, and 0.85 for precision, recall, accuracy, and F1-score, respectively. Other deep learning approaches should have also been tried to maximize the performance. The proposed framework misclassified attack types such as MSCI and MPCI and considered their behavior normal instead of harmful [99].

The authors studied previous literature and noted problems with the previous approaches in the detection of zero-day attacks as well as irregularities in data, which results in a poor rate of attack detection. They proposed an approach with steps starting from Bloom Filter payload level detection, then used the Kohonen enhanced neural network by utilizing PCA and hyper-graph partitioning, and then finally used BLOSOM-based hybrid anomaly detection using datasets obtained from Singapore University (SWAT dataset) and Mississippi State University (gas-pipeline data from SCADA lab). The BLOSOM model imputes packet contents for checking the data's behavioral pattern in an unsupervised fashion. It helps identify whether the packet contents lie within the ANN training phase.

The proposed technique using both datasets showed improved results compared to BLF, RF, RNN, and CNN. The results were 0.97, 0.98, 0.92, and 0.95 in terms of accuracy, precision, recall, and F1-score, respectively [100].

The authors focused on anomaly detection in industrial control systems by considering packet latency and jitter. An algorithm based upon Grey Wolf optimizer neural network training for anomaly detection was proposed to be applied in industrial control systems. Multiple datasets, gas-pipeline, and swat from different sources were used. Grey Wolf Optimizer (GWO) is the principle used for imitating wolves' behavior in nature to hunt cooperatively. Leadership hierarchy is imitated by alpha, omega, alpha, delta, and beta wolves. Optimization is performed by three basic steps: prey searching, encircling, and attacking prey. The performance of a grey wolf algorithm using gas and swat datasets was compared against PSO, BBO, ACO, ES, and PBIL optimized algorithms with ANN. The accuracy of the Grey Wolf algorithm was 98% on gas-pipeline while it achieved 96% accuracy on swat. Regarding robustness and accuracy, GWO achieves higher. If time is of higher concern, then the ES algorithm takes less time [101].

The focus of the researchers was to combine supervised and unsupervised learning to detect known and unknown attacks. The primary proposed technique is a stacker, a two-layer meta-learning adaptation. Datasets such as ISCX, UNSW-NB15, ADFA-NetflowIDS, CIDDS, and NSL-KDD were taken from the public, containing real systems while creating variants allowing for anonymous attack analysis. The most valuable features were extracted from datasets using Information Gain, then the supervised algorithm picked and derived features from the meta-feature set. Then, for the entire dataset, unsupervised model-based features were generated. The dataset was also split into a 30:70 ratio. A 50:50 ratio was used for training and testing the supervised algorithm in the case of meta-level learners. The performance of the stacker was compared against different datasets, and the highest accuracy was shown for CICIDS2017 by 0.9997 alongside 0.998 (precision), 1.000 (recall), and 0.999 (F1-Score) [102].

For detecting zero-day attacks that are phishing URL-based, URLs that are constructed and destructed quickly after the attacks, deep learning, and logic programmed domain knowledge integration were suggested. Logic and neural classifiers were designed and approached joint learning on symbolic neurological integration. Datasets containing 222,541 URLs were used. The LSTM CNN-based triplet network comprising of shared weight based upon a neural feature extractor is a promising approach for learning the latent space of phishing by comparing URLs with standard-similarity functions. Two steps are used in the process of learning latent phishing space: normal and phishing URLs distance metrics and feature extraction of the learning function. The results were compared with previously proposed triplets while showing an accuracy and recall of 97.8% and 96.1% for the ISCX URL dataset [103].

Another study [104] used an ML-based insider threat detection model based on KNN; they used two approaches to detect insider threats: user-based and role base. In user-based, they calculate the abnormal score of the user's session with the previous sessions based on his activities in the system, and for role-based, they apply the same technique to sessions in different roles and compare the results with the previous sessions of the same role and then calculated the abnormal score.

Hidden Markov Model (HMM) was also used for insider threat detection by Ref. [105]; they used the CERT r.4.2, which contains 70 malicious users, and then trained the users in the first week, recorded their activities on different tasks on a weekly basis, found the similarities between the activities, and then classified them as malicious or normal.

In Ref. [106], the authors introduced a multilayer machine learning-based insider threat detection model that employed three techniques for insider threat detection, namely Misused Insider Threat Detection(MITD) using Random Forest Algorithm, Anomaly Insider Threat Detection (AITD) using K-Nearest Neighbor, and Hybrid Insider Threat Detection (HITD) by the combination of both MITD and AITD. The model is observed with 97% accuracy and 2.88% False Positive Rate (FPR) for unknown (Zero-day) threats using the

HITD, while MITD and AITD showed an accuracy of 92% and 90%, respectively. They also compared their model with existing ML-based insider threat detection approaches, including SVM, LSTM, and DNN. Although it showed better accuracy over the other algorithms, a still false negative is a bit high—9.56%—which is alarming because zero-day attacks (ZAs) are critical in any system because they can cause serious damage to the organizations.

Another autoencoder-based model is proposed by Kunang et al. [97]. They applied automatic feature extraction on IDS using the autoencoder approach. Their experiments showed a high accuracy of around 99.947%, which is quite impressive. However, unfortunately, they used very old datasets from 1998, and hence will be missing the signatures of new attacks.

With its many benefits, the increasing number of CAVs can give rise to new challenges in terms of privacy, security, and, more significantly, the safety of travelers [97]. Due to connectivity with the internet, their sensors can play a role in cyber-attacks on vehicles, which can result in fatal crashes or even car hijacking, etc. In the article, the authors presented a technique for anomaly detection from CAVs by combining CNN with a famous anomaly detection approach called 'Kalman filtering' to identify abnormal behaviors in CAVs. This hybrid approach outperformed Kalman filters and CNN in anomaly detection with an impressive accuracy of 98.2%.

In the coming years, objects will be connecting more and more, generating a massive amount of data that could grab the attention of potential attackers, posing different cyber threats to the privacy of individuals and organizations. The IP reputation system plays a vital role in identifying Malicious Internet Protocols (IPs). It profiles the behaviors of potential security threats to different cyber-physical systems, but previously available reputation systems were not suitable for everyone due to their high management costs; they also had limitations in terms of performance, i.e., false-positive rates, running time, and relying on minimal data sources for validating the reputation of IP addresses. Nighat et al. [98] introduced a Dynamic Malware Analysis, Machine Learning, Data Forensics, and Cyberthreat Intelligence-based unique hybrid technique to address the abovementioned concerns. IP reputation is anticipated in the pre-acceptance stage using the idea of big data forensics, and the associated zero-day assaults are classified using behavioral analysis, which uses the Decision Tree (DT) approach. The suggested method identifies extensive data forensic concerns and computes severity, risk score, confidence, and longevity simultaneously. The proposed system is assessed in two ways: first, they compare ML algorithms to determine the best F-measure, accuracy, and recall scores, and then they compare the complete reputation system to other reputation systems.

In today's Industrial Control Systems (ICS), the extensive connectivity between smart equipment opens up a large space for various cyber-attacks and harmful threats. These assaults have the potential to not only interrupt or fully impair system functionality but also to have major safety implications. As a result, one of the most critical challenges in ICS is cyber security. A recent study [100] proposed a way to design algorithms for detecting cyber-attacks on communication channels between smart devices in this study. The method is based on Convolutional Neural Networks and belongs to the family of semi-supervised data-driven approaches (CNN). The proposed technique automatically determines acceptable CNN architecture and thresholds for online intrusion detection based on a pre-set range of network hyperparameters and data received from a system operation without assaults. The suggested intrusion detection is host-based; in keeping with the features of ICS, they analyzed the structure of ICS and the practicality of implementing the attack detection algorithm on control system devices. Two case studies were used to validate the strategy experimentally. The comparison analysis of the proposed method with other approaches used a publicly accessible dataset derived from the Secure Water Treatment (SWaT) testbed.

Figure 5 shows the overall accuracy of different techniques (BF, RF, LSTM, SVDD, Stacker, etc.), which were trained and tested against different datasets covering various attack vectors from various sizes and multiple domains, including industrial-level systems.

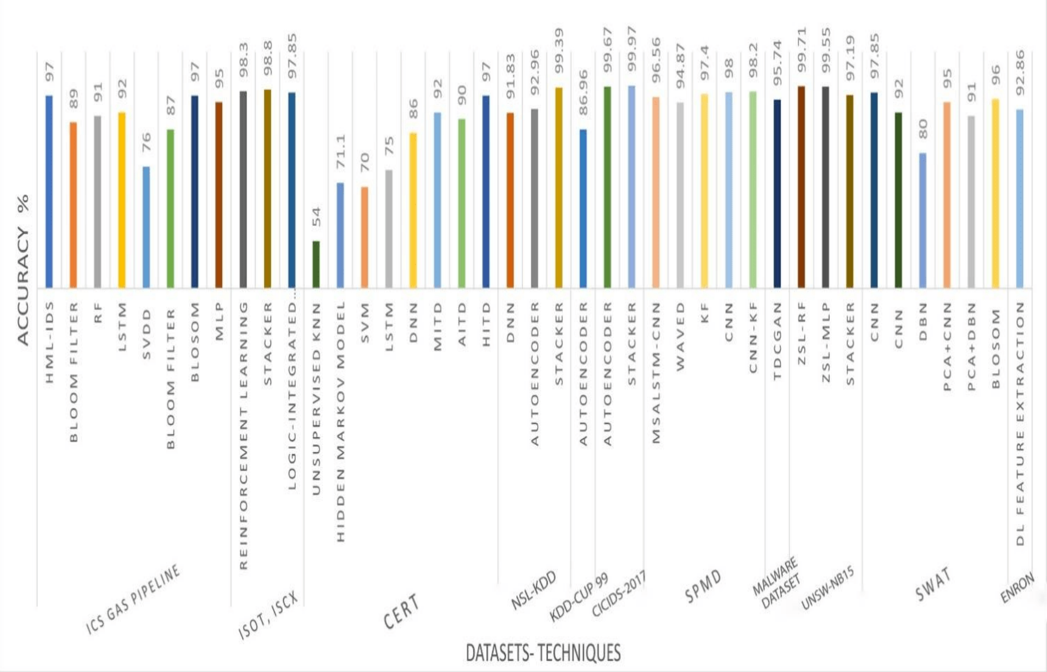

Figure 5. The accuracy of the compared techniques on different datasets.

The precision of previously discussed algorithms and techniques that are trained and tested against various datasets is shown in Figure 6.

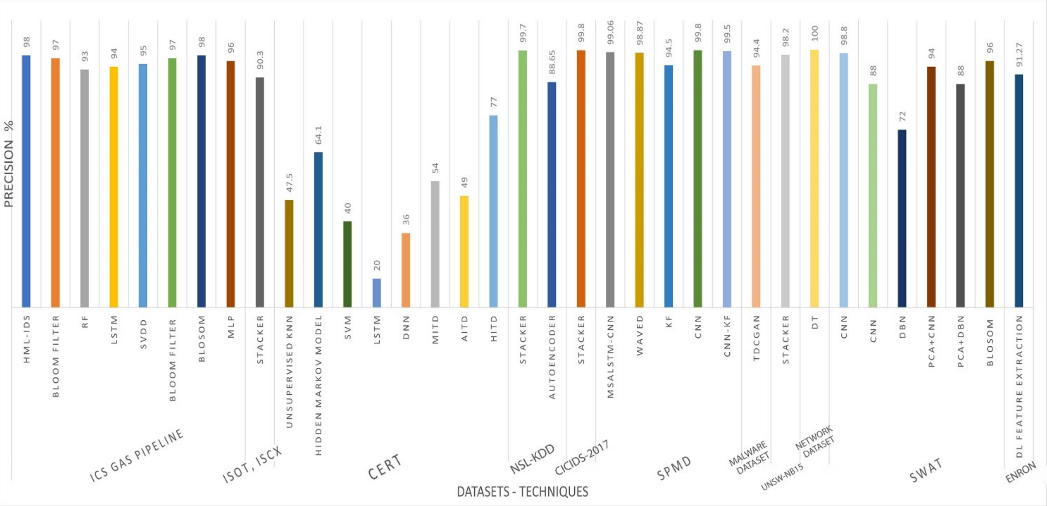

Figure 6. Precision of the compared techniques on different datasets.

Figure 7 presents the recall score of various algorithms and techniques alongside their datasets. The promising recall value of 100% is achieved using the STACKER approach evaluated on the CICIDS-2017 dataset.

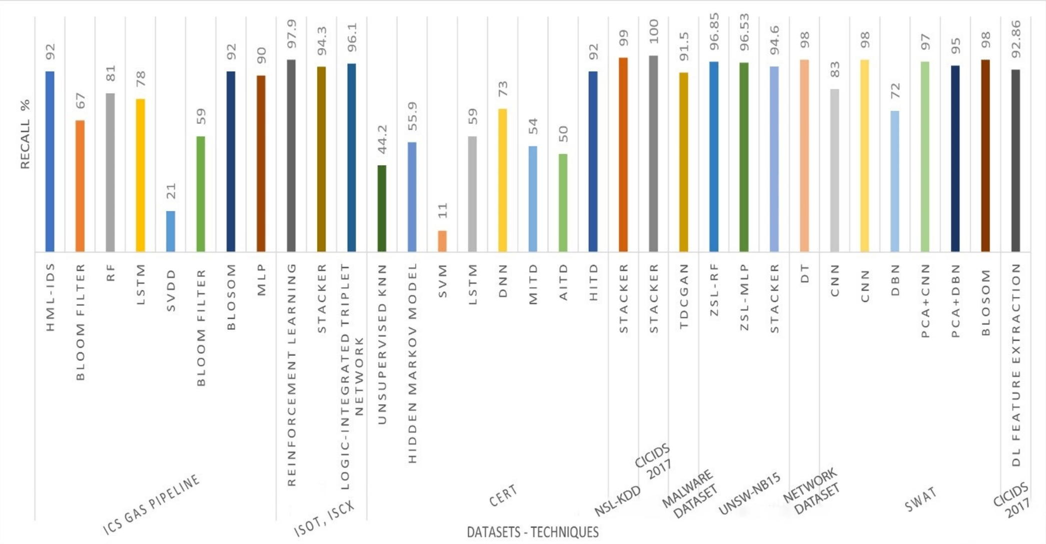

Figure 7. Recall of the compared techniques on different datasets.

The F1-score of the above-mentioned datasets and approaches is depicted in Figure 8. The highest F1-score of 99.9% has been achieved using STACKER on the CICIDS-2017 dataset.

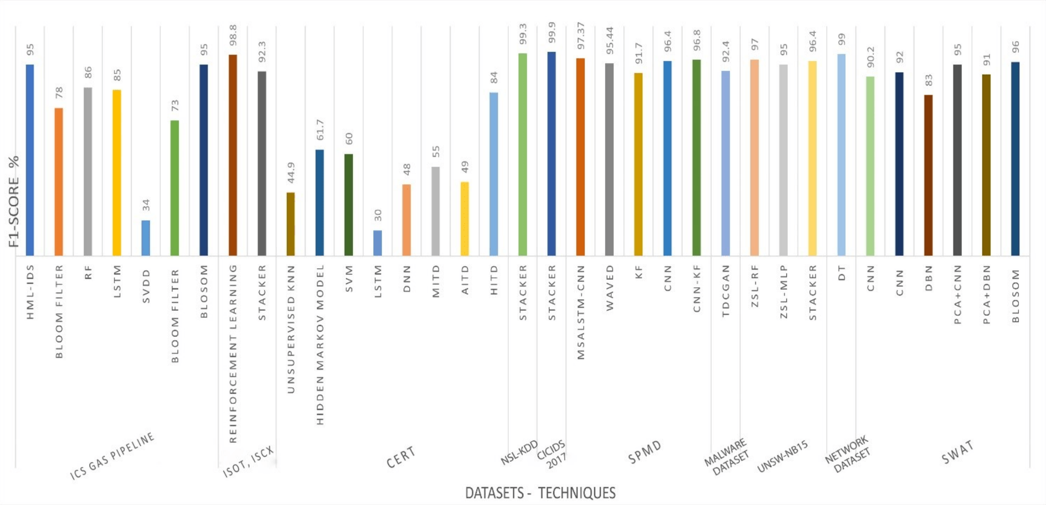

Figure 8. The F1-score of the compared techniques on different datasets.

4. Research Limitations

In the last decade, many researchers have introduced AI-based approaches to detect zero-day attacks based on their behaviors and other factors. However, they still face high false negative and false positive rates because most of them used very old datasets such as DARPA or KDD. These datasets are vast and famous within the field of cyber security; however, they will be missing the new attacks and new strategies that are used nowadays in attacks. Furthermore, most of the researchers trained and tested their models on high-volume traffic to capture the zero-day attacks, but in the case of zero-day attacks, they can be in the form of meager traffic; it is not always coming from DoS or DDoS. Since zero days are more critical than regular attacks, a very efficient model should handle the traffic in real-time. Although some of them achieved impressive results, their custom datasets may have been designed in a way that produced good results on their systems; when the same models were applied to other datasets, they came up with many different results [53]. Moreover, everyone uses different evaluation metrics to express their results, which are not even suitable for cyber security scenarios.

5. Conclusions and Future Work

This research has compared the latest articles regarding zero-day attack detection in multiple domains, such as IDS, DoS/DDoS, anomaly-based (IDS), and malware. We have reviewed almost 10 different approaches in the generics attacks detected by IDS. It can be seen that the stacker-based approach was the best in terms of accuracy as it reached up to 98.8% accuracy. In terms of precision, both HML-IDS and BLOSOM attained 98% precision. The reinforcement learning approach achieved higher recall and F1-Score, 97.9%, and 98.8, respectively. For Dos/DDoS attack, on the dataset of CICIDS2017, the stacker-based approach outperforms all the others by achieving 99.97% accuracy, 99.8% precision, 100% recall, and 99% in the F1-Score metric. Regarding anomaly-based attack detection, the CNN-KF-based approach on the dataset of SPMD achieved 98.2% accuracy, which is the highest among the other compared approaches. However, the highest precision was 99.8%, which CNN achieved on the same dataset of SPMD. The highest recall was achieved by the BLOSOM approach on the SWat dataset. MSALSTM-CNN approach outperformed others in terms of F1-score.

The approach of the tDCGAN, over the malware dataset, outperformed other compared approaches by achieving an accuracy of 95.74%. Other metrics such as precision, recall, and F1-score were maximum on the network dataset by a decision tree. In all aspects, zero-day attacks are critical to any system; whether in IDS systems or spam-based, they can cause massive damage to any organization. Rival companies find zero-day vulnerabilities and exploit them against their competitors by gaining access to their sensitive data by compromising their systems. Attackers normally try to remain inside the system without letting anyone know that they are about to steal confidential data instead of destroying the system. They tend to remain hidden as long as possible because, once the security team learns about the vulnerability, they fix the issue as quickly as possible. Due to there not being any previous records and signatures, zero-day attacks are hard to detect using firewalls or other security measures used by organizations to keep their systems secure from potential threats.

Due to the sensitivity of the zero-day attacks, relying on only accuracy or precision is not enough to measure the performance of the models; they should use false alarms to implement their models on existing systems. Secondly, instead of using outdated datasets, new benchmarks should be used to tackle modern attacks properly.

Author Contributions: Conceptualization, S.A. and A.I.; methodology, S.A. and S.U.R.; validation, G.A. and Z.I.; formal analysis, G.A.; resources, K.-I.K.; data curation, A.I. and Z.I.; writing—original draft preparation, S.A.; writing—review and editing, S.U.R.; visualization, K.-I.K.; supervision, S.U.R.; project administration, K.-I.K.; funding acquisition, K.-I.K. All authERROR: Failed to execute system command: ors have read and agreed to publish the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-01200, Training Key Talents in Industrial Convergence Security) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2022-00165225).

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

References

1. Yoon, H.; Jang, Y.; Kim, S.; Speasmaker, A.; Nam, I. Trends in internet use among older adults in the United States, 2011–2016. *J. Appl. Gerontol.* **2021**, *40*, 466–470. [CrossRef]
2. Alhashmi, A.A.; Darem, A.; Abawajy, J.H. Taxonomy of Cybersecurity Awareness Delivery Methods: A Countermeasure for Phishing Threats. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*. [CrossRef]
3. Al-Marghilani, A. Comprehensive Analysis of IoT Malware Evasion Techniques. *Eng. Technol. Appl. Sci. Res.* **2021**, *11*, 7495–7500. [CrossRef]
4. Bhattacharyya, D.K.; Kalita, J.K. *Network Anomaly Detection: A Machine Learning Perspective*; CRC Press: Boca Raton, FL, USA, 2013.

5. Zeng, Y.; Hu, X.; Shin, K.G. Detection of botnets using combined host-and network-level information. In Proceedings of the 2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), Chicago, IL, USA, 28 June 2010–1 July 2010; pp. 291–300.
6. Studnia, I.; Nicomette, V.; Alata, E.; Deswarte, Y.; Kaâniche, M.; Laarouchi, Y. Survey on security threats and protection mechanisms in embedded automotive networks. In Proceedings of the 2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W), Budapest, Hungary, 24–27 June 2013; pp. 1–12.
7. Meakins, J. A zero-sum game: The zero-day market in 2018. *J. Cyber Policy* **2019**, *4*, 60–71. [CrossRef]
8. Fang, B.; Lu, Q.; Pattabiraman, K.; Ripeanu, M.; Gurumurthi, S. ePVF: An enhanced program vulnerability factor methodology for cross-layer resilience analysis. In Proceedings of the 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Toulouse, France, 28 June–1 July 2016; pp. 168–179.
9. Ambalavanan, V. Cyber threats detection and mitigation using machine learning. In *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*; IGI Global: Hershey, PA, USA, 2020; pp. 132–149.
10. Nabi, S.; Rehman, S.U.; Fong, S.; Aziz, K. A model for implementing security at application level in service oriented architecture. *J. Emerg. Technol. Web Intell.* **2014**, *6*, 157–163. [CrossRef]
11. Craigen, D.; Diakun-Thibault, N.; Purse, R. Defining cybersecurity. *Technol. Innov. Manag. Rev.* **2014**, *4*, 13–21. [CrossRef]
12. He, S.; Zhu, J.; He, P.; Lyu, M.R. Experience report: System log analysis for anomaly detection. In Proceedings of the 2016 IEEE 27th international symposium on software reliability engineering (ISSRE), Ottawa, ON, Canada, 23–27 October 2016; pp. 207–218.
13. Al-Qatf, M.; Lasheng, Y.; Al-Habib, M.; Al-Sabahi, K. Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. *IEEE Access* **2018**, *6*, 52843–52856. [CrossRef]
14. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.K.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A taxonomy of network threats and the effect of current datasets on intrusion detection systems. *IEEE Access* **2020**, *8*, 104650–104675. [CrossRef]
15. Pan, K.; Rakhshani, E.; Palensky, P. False data injection attacks on hybrid AC/HVDC interconnected systems with virtual inertia—Vulnerability, impact and detection. *IEEE Access* **2020**, *8*, 141932–141945. [CrossRef]
16. Zoppi, T.; Ceccarelli, A.; Salani, L.; Bondavalli, A. On the educated selection of unsupervised algorithms via attacks and anomaly classes. *J. Inf. Secur. Appl.* **2020**, *52*, 102474. [CrossRef]
17. Hanselmann, M.; Strauss, T.; Dormann, K.; Ulmer, H. CANet: An unsupervised intrusion detection system for high dimensional CAN bus data. *IEEE Access* **2020**, *8*, 58194–58205. [CrossRef]
18. Latif, J.; Xiao, C.; Imran, A.; Tu, S. Medical imaging using machine learning and deep learning algorithms: A review. In Proceedings of the 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 30–31 January 2019; pp. 1–5.
19. Latif, J.; Xiao, C.; Tu, S.; Rehman, S.U.; Imran, A.; Bilal, A. Implementation and use of disease diagnosis systems for electronic medical records based on machine learning: A complete review. *IEEE Access* **2020**, *8*, 150489–150513. [CrossRef]
20. Vargas, R.; Mosavi, A.; Ruiz, R. Deep learning: A review *Advances in Intelligent Systems and Computing* **2017**, *5*, 1–10.
21. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
22. Biabani, S.A.A.; Tayyib, N.A. A Review on the Use of Machine Learning Against the Covid-19 Pandemic. *Eng. Technol. Appl. Sci. Res.* **2022**, *12*, 8039–8044. [CrossRef]
23. Chapman, C. *Network Performance and Security: Testing and Analyzing Using Open Source and Low-Cost Tools*; Syngress: Oxford, UK, 2016.
24. Singh, A.P. A study on zero day malware attack. *Int. J. Adv. Res. Comput. Commun. Eng.* **2017**, *6*, 391–392. [CrossRef]
25. Bilge, L.; Dumitraş, T. Before we knew it: An empirical study of zero-day attacks in the real world. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, Raleigh, NC, USA, 16–18 October 2012; pp. 833–844.
26. Nguyen, T.T.; Reddi, V.J. Deep reinforcement learning for cyber security. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**. [CrossRef] [PubMed]
27. Metrick, K.; Najafi, P.; Semrau, J. Zero-Day Exploitation Increasingly Demonstrates Access to Money, Rather than Skill—Intelligence for Vulnerability Management. Technical Report, Technical REPORT, FireEye Technical Report. Available online: https://www.fireeye.com/blog/threat-research/2020/04/zero-day-exploitation-demonstrates-access-to-money-not-skill.html (accessed on 1 September 2022).
28. 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]
29. Albanese, M.; Jajodia, S.; Singhal, A.; Wang, L. An efficient approach to assessing the risk of zero-day vulnerabilities. In Proceedings of the 2013 International Conference on Security and Cryptography (SECRYPT), Reykjavik, Iceland, 29–31 July 2013; pp. 1–12.
30. Kaloudi, N.; Li, J. The ai-based cyber threat landscape: A survey. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–34. [CrossRef]
31. Hindy, H.; Hodo, E.; Bayne, E.; Seeam, A.; Atkinson, R.; Bellekens, X. A taxonomy of malicious traffic for intrusion detection systems. In Proceedings of the 2018 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA), Scotland, UK, 11–12 June 2018; pp. 1–4.
32. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 1–22. [CrossRef]

33. Palmieri, F. Network anomaly detection based on logistic regression of nonlinear chaotic invariants. *J. Netw. Comput. Appl.* **2019**, *148*, 102460. [CrossRef]
34. Duessel, P.; Gehl, C.; Flegel, U.; Dietrich, S.; Meier, M. Detecting zero-day attacks using context-aware anomaly detection at the application-layer. *Int. J. Inf. Secur.* **2017**, *16*, 475–490. [CrossRef]
35. Moon, D.; Pan, S.B.; Kim, I. Host-based intrusion detection system for secure human-centric computing. *J. Supercomput.* **2016**, *72*, 2520–2536. [CrossRef]
36. Moustafa, N.; Choo, K.K.R.; Radwan, I.; Camtepe, S. Outlier dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1975–1987. [CrossRef]
37. Kaur, R.; Singh, M. A hybrid real-time zero-day attack detection and analysis system. *Int. J. Comput. Netw. Inf. Secur.* **2015**, *7*, 19–31. [CrossRef]
38. Khan, I.A.; Pi, D.; Khan, Z.U.; Hussain, Y.; Nawaz, A. HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. *IEEE Access* **2019**, *7*, 89507–89521. [CrossRef]
39. Sun, X.; Dai, J.; Liu, P.; Singhal, A.; Yen, J. Using Bayesian networks for probabilistic identification of zero-day attack paths. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2506–2521. [CrossRef]
40. Bayoğlu, B.; Soğukpınar, İ. Graph based signature classes for detecting polymorphic worms via content analysis. *Comput. Netw.* **2012**, *56*, 832–844. [CrossRef]
41. Yichao, Z.; Tianyang, Z.; Xiaoyue, G.; Qingxian, W. An improved attack path discovery algorithm through compact graph planning. *IEEE Access* **2019**, *7*, 59346–59356. [CrossRef]
42. Grana, J.; Wolpert, D.; Neil, J.; Xie, D.; Bhattacharya, T.; Bent, R. A likelihood ratio anomaly detector for identifying within-perimeter computer network attacks. *J. Netw. Comput. Appl.* **2016**, *66*, 166–179. [CrossRef]
43. Wang, B.; Zheng, Y.; Lou, W.; Hou, Y.T. DDoS attack protection in the era of cloud computing and software-defined networking. *Comput. Netw.* **2015**, *81*, 308–319. [CrossRef]
44. Singh, U.K.; Joshi, C.; Kanellopoulos, D. A framework for zero-day vulnerabilities detection and prioritization. *J. Inf. Secur. Appl.* **2019**, *46*, 164–172. [CrossRef]
45. Abirami, S.; Chitra, P. Energy-efficient edge based real-time healthcare support system. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 2020; Volume 117, pp. 339–368.
46. Ma, L.; Chamberlain, R.D.; Buhler, J.D.; Franklin, M.A. Bloom filter performance on graphics engines. In Proceedings of the 2011 International Conference on Parallel Processing, Taipei, Taiwan, 13–16 September 2011; pp. 522–531.
47. Bloom, B.H. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* **1970**, *13*, 422–426. [CrossRef]
48. Harrison, A.B. *Peer-to-Grid Computing: Spanning Diverse Service-Oriented Architectures*; Cardiff University: Cardiff, UK, 2008.
49. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
50. Jemal, I.; Haddar, M.A.; Cheikhrouhou, O.; Mahfoudhi, A. M-CNN: A new hybrid deep learning model for web security. In Proceedings of the 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), Antalya, Turkey, 2–5 November 2020; pp. 1–7.
51. Jemal, I.; Haddar, M.A.; Cheikhrouhou, O.; Mahfoudhi, A. Malicious http request detection using code-level convolutional neural network. In Proceedings of the International Conference on Risks and Security of Internet and Systems, Paris, France, 4–6 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 317–324.
52. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; ACM Inc.: New York, NY, USA, 1995.
53. Romera-Paredes, B.; Torr, P. An embarrassingly simple approach to zero-shot learning. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 2152–2161.
54. Tax, D.M.; Duin, R.P. Support vector data description. *Mach. Learn.* **2004**, *54*, 45–66. [CrossRef]
55. Kebede, T.M.; Djaneye-Boundjou, O.; Narayanan, B.N.; Ralescu, A.; Kapp, D. Classification of malware programs using autoencoders based deep learning architecture and its application to the microsoft malware classification challenge (big 2015) dataset. In Proceedings of the 2017 IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 27–30 June 2017; pp. 70–75.
56. Fukushima, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Netw.* **1988**, *1*, 119–130. [CrossRef]
57. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
58. Wallach, I.; Dzamba, M.; Heifets, A. AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv* **2015**, arxiv:1510.02855.
59. Ren, H.; Xu, B.; Wang, Y.; Yi, C.; Huang, C.; Kou, X.; Xing, T.; Yang, M.; Tong, J.; Zhang, Q. Time-series anomaly detection service at microsoft. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 3009–3017.
60. Vinayakumar, R.; Soman, K.; Poornachandran, P. Applying convolutional neural network for network intrusion detection. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Manipal, India, 13–16 September 2017; pp. 1222–1228.
61. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 8–11 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.

62. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
63. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
64. Marsland, S. *Machine Learning: An Algorithmic Perspective*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2011.
65. Granter, S.R.; Beck, A.H.; Papke Jr, D.J. AlphaGo, deep learning, and the future of the human microscopist. *Arch. Pathol. Lab. Med.* **2017**, *141*, 619–621. [CrossRef]
66. Chen, J.X. The evolution of computing: AlphaGo. *Comput. Sci. Eng.* **2016**, *18*, 4–7. [CrossRef]
67. Xu, X.; Xie, T. A reinforcement learning approach for host-based intrusion detection using sequences of system calls. In *Proceedings of the International Conference on Intelligent Computing*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 995–1003.
68. Xu, X.; Sun, Y.; Huang, Z. Defending DDoS attacks using hidden Markov models and cooperative reinforcement learning. In *Proceedings of the Pacific-Asia Workshop on Intelligence and Security Informatics*; Springer: Berlin/Heidelberg, Germany, 2007, pp. 196–207.
69. Smadi, S.; Aslam, N.; Zhang, L. Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decis. Support Syst.* **2018**, *107*, 88–102. [CrossRef]
70. Feng, M.; Xu, H. Deep reinforecement learning based optimal defense for cyber-physical system in presence of unknown cyber-attack. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–8.
71. Baek, J.; Choi, Y. Deep neural network for predicting ore production by truck-haulage systems in open-pit mines. *Appl. Sci.* **2020**, *10*, 1657. [CrossRef]
72. Feng, C.; Li, T.; Chana, D. Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks. In Proceedings of the 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Denver, CO, USA, 26–29 June 2017; pp. 261–272.
73. Jagtap, S.S.; Sriram, S.V.S.; Subramaniyaswamy, V. A hypergraph based Kohonen map for detecting intrusions over cyber–physical systems traffic. *Future Gener. Comput. Syst.* **2021**, *119*, 84–109. [CrossRef]
74. Alauthman, M.; Aslam, N.; Al-Kasassbeh, M.; Khan, S.; Al-Qerem, A.; Choo, K.K.R. An efficient reinforcement learning-based Botnet detection approach. *J. Netw. Comput. Appl.* **2020**, *150*, 102479. [CrossRef]
75. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
76. Sarhan, M.; Layeghy, S.; Gallagher, M.; Portmann, M. From Zero-Shot Machine Learning to Zero-Day Attack Detection. *arXiv* **2021**, arxiv:2109.14868.
77. Shaukat, K.; Luo, S.; Varadharajan, V.; Hameed, I.A.; Xu, M. A survey on machine learning techniques for cyber security in the last decade. *IEEE Access* **2020**, *8*, 222310–222354. [CrossRef]
78. Sterman, J. *Business Dynamics*; Irwin/McGraw-Hill: Irvine, CA, USA, 2010.
79. RM, S.P.; Maddikunta, P.K.R.; Parimala, M.; Koppu, S.; Gadekallu, T.R.; Chowdhary, C.L.; Alazab, M. An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. *Comput. Commun.* **2020**, *160*, 139–149.
80. Javed, A.R.; Usman, M.; Rehman, S.U.; Khan, M.U.; Haghighi, M.S. Anomaly detection in automated vehicles using multistage attention-based convolutional neural network. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4291–4300. [CrossRef]
81. Blaise, A.; Bouet, M.; Conan, V.; Secci, S. Detection of zero-day attacks: An unsupervised port-based approach. *Comput. Netw.* **2020**, *180*, 107391. [CrossRef]
82. Hindy, H.; Atkinson, R.; Tachtatzis, C.; Colin, J.N.; Bayne, E.; Bellekens, X. Utilising deep learning techniques for effective zero-day attack detection. *Electronics* **2020**, *9*, 1684. [CrossRef]
83. Sameera, N.; Shashi, M. Deep transductive transfer learning framework for zero-day attack detection. *ICT Express* **2020**, *6*, 361–367. [CrossRef]
84. Vinayakumar, R.; Alazab, M.; Soman, K.; Poornachandran, P.; Venkatraman, S. Robust intelligent malware detection using deep learning. *IEEE Access* **2019**, *7*, 46717–46738. [CrossRef]
85. Vercruyssen, V.; Meert, W.; Davis, J. Transfer learning for time series anomaly detection. In Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning@ ECMLPKDD 2017, CEUR Workshop Proceedings, Skopje, Macedonia, 18 September 2017; Volume 1924, pp. 27–37.
86. Sameera, N.; Shashi, M. Transfer learning based prototype for zero-day attack detection. *Int. J. Eng. Adv. Technol. (IJEAT)* **2019**, *8*, 1326–1329.
87. Kim, J.Y.; Bu, S.J.; Cho, S.B. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. *Inf. Sci.* **2018**, *460*, 83–102. [CrossRef]
88. Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener. Comput. Syst.* **2018**, *82*, 761–768. [CrossRef]
89. Saied, A.; Overill, R.E.; Radzik, T. Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing* **2016**, *172*, 385–393. [CrossRef]

90. Ur Rehman, S.; Khaliq, M.; Imtiaz, S.I.; Rasool, A.; Shafiq, M.; Javed, A.R.; Jalil, Z.; Bashir, A.K. Diddos: An approach for detection and identification of distributed denial of service (ddos) cyberattacks using gated recurrent units (gru). *Future Gener. Comput. Syst.* **2021**, *118*, 453–466. [CrossRef]
91. Javed, A.R.; Ur Rehman, S.; Khan, M.U.; Alazab, M.; Reddy, T. CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1456–1466. [CrossRef]
92. Afek, Y.; Bremler-Barr, A.; Feibish, S.L. Zero-day signature extraction for high-volume attacks. *IEEE/ACM Trans. Netw.* **2019**, *27*, 691–706. [CrossRef]
93. More, P.; Mishra, P. Enhanced-PCA based dimensionality reduction and feature selection for real-time network threat detection. *Eng. Technol. Appl. Sci. Res.* **2020**, *10*, 6270–6275. [CrossRef]
94. Balamurugan, V.; Saravanan, R. Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation. *Clust. Comput.* **2019**, *22*, 13027–13039. [CrossRef]
95. Saba Jameel, S.U.R. An optimal feature selection method using a modified wrapper-based ant colony optimisation. *Natl. Sci. Found Sri Lanka* **2018**, *46*, 143–151. [CrossRef]
96. Yavanoglu, O.; Aydos, M. A review on cyber security datasets for machine learning algorithms. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 2186–2193.
97. Van Wyk, F.; Wang, Y.; Khojandi, A.; Masoud, N. Real-time sensor anomaly detection and identification in automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1264–1276. [CrossRef]
98. Usman, N.; Usman, S.; Khan, F.; Jan, M.A.; Sajid, A.; Alazab, M.; Watters, P. Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics. *Future Gener. Comput. Syst.* **2021**, *118*, 124–141. [CrossRef]
99. Mansouri, A.; Majidi, B.; Shamisa, A. Metaheuristic neural networks for anomaly recognition in industrial sensor networks with packet latency and jitter for smart infrastructures. *Int. J. Comput. Appl.* **2021**, *43*, 257–266. [CrossRef]
100. Nedeljkovic, D.; Jakovljevic, Z. CNN based method for the development of cyber-attacks detection algorithms in industrial control systems. *Comput. Secur.* **2022**, *114*, 102585. [CrossRef]
101. Zoppi, T.; Ceccarelli, A. Prepare for trouble and make it double! Supervised–Unsupervised stacking for anomaly-based intrusion detection. *J. Netw. Comput. Appl.* **2021**, *189*, 103106. [CrossRef]
102. Bu, S.J.; Cho, S.B. Integrating deep learning with first-order logic programmed constraints for zero-day phishing attack detection. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 2685–2689.
103. Böse, B.; Avasarala, B.; Tirthapura, S.; Chung, Y.Y.; Steiner, D. Detecting insider threats using radish: A system for real-time anomaly detection in heterogeneous data streams. *IEEE Syst. J.* **2017**, *11*, 471–482. [CrossRef]
104. Lo, O.; Buchanan, W.J.; Griffiths, P.; Macfarlane, R. Distance measurement methods for improved insider threat detection. *Secur. Commun. Netw.* **2018**, *2018*, 5906368. [CrossRef]
105. Al-Mhiqani, M.N.; Ahmad, R.; Abidin, Z.Z.; Abdulkareem, K.H.; Mohammed, M.A.; Gupta, D.; Shankar, K. A new intelligent multilayer framework for insider threat detection. *Comput. Electr. Eng.* **2022**, *97*, 107597. [CrossRef]
106. Kunang, Y.N.; Nurmaini, S.; Stiawan, D.; Zarkasi, A. Automatic features extraction using autoencoder in intrusion detection system. In Proceedings of the 2018 International Conference on Electrical Engineering and Computer Science (ICECOS), Pangkal, Indonesia, 2–4 October 2018; pp. 219–224.