smart cities

Article

Intrusion Detection in Critical Infrastructures: A Literature Review

Fountas Panagiotis [1], Kouskouras Taxiarxchis [1], Kranas Georgios [1], Leandros Maglaras [2,*] and Mohamed Amine Ferrag [3]

[1] Department of Computer Science and Biomedical Informatics, University of Thessaly, Papasiopoulou 2-4, 35131 Lamia, Greece; pfountas@uth.gr (F.P.); tkouskouras@uth.gr (K.T.); gekranas@uth.gr (K.G.)
[2] School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK
[3] Department of Computer Science, Guelma University, Guelma 24000, Algeria; ferrag.mohamedamine@univ-guelma.dz
* Correspondence: leandros.maglaras@dmu.ac.uk

Abstract: Over the years, the digitization of all aspects of life in modern societies is considered an acquired advantage. However, like the terrestrial world, the digital world is not perfect and many dangers and threats are present. In the present work, we conduct a systematic review on the methods of network detection and cyber attacks that can take place in a critical infrastructure. As is shown, the implementation of a system that learns from the system behavior (machine learning), on multiple levels and spots any diversity, is one of the most effective solutions.

Keywords: critical infrastructures; intrusion detection systems; digitization

Citation: Panagiotis, F.; Taxiarxchis, K.; Georgios, K.; Maglaras, L.; Ferrag, M.A. Intrusion Detection in Critical Infrastructures: A Literature Review. *Smart Cities* **2021**, *4*, 1146–1157. https://doi.org/10.3390/smartcities4030061

Academic Editors: Antonio Moreno-Munoz and Emilio José Palacios-García

Received: 30 July 2021
Accepted: 25 August 2021
Published: 28 August 2021

1. Introduction

Nowadays, the tremendous development of technology has resulted in the permeation of it in all sectors of industries and infrastructures. This phenomenon is inevitable because modern societies have increased demands of resources to function properly [1]. Due to the extensive use of technology, production can be increased and society's demands handled [2]. These infrastructures are called critical infrastructures due to the significant roles that they play. However, the adoption of technologies in the function of critical infrastructures allows several attacks to exploit the vulnerabilities of the critical infrastructures [3]. The scientific community tries to detect the vulnerabilities, possible threats, and attacks in critical infrastructures to develop security systems that prevent those attacks. One of the most well-known and commonly applied types of attack is the intrusion attack.

Today, countries have to support the increased resources that societies need to remain functional and protect their economies from crises. For this reason, countries rely on infrastructures (assets, industries, and systems) to handle and provide the required resources such as energy, communications, transportation, etc. We characterize as critical infrastructures the group of infrastructures that handle important resources such as power grids, water treatment plants, health services, and any other service which relates to the maintenance of the national economy, health, and security. The most crucial of the critical infrastructures are energy, water, transportation, and communications [4] since these infrastructures handle resources for the majority of operations of modern societies. Moreover, all critical infrastructures are interconnected and interdependent between them and with the sectors of the economy. This strong association between the critical infrastructure sectors means that damage in one service of a sector or even worse the loss of one sector will undoubtedly affect, to the same or a greater degree, the other critical infrastructures.

The large necessity of countries to satisfy the aforementioned requirements caused a rapid inflow of technology in the critical infrastructures to control their operations and maximize their performances. However, these computer systems have vulnerabilities and there are threats to their security. The high rate of cyber attacks in critical infrastructures such as the well-known Stuxnet malware, whose goal was to damage a nuclear power plant in Iran [5], confirms the necessity to develop algorithms, techniques, and software to counteract such attacks.

This research focuses on intrusion detection in critical infrastructures. More specifically, we refer to some of the most common attacks that can be used by the attacker to take the control of a system or cause damage to it. Furthermore, this research contains a description of techniques and models which have been implemented in several intrusion detection systems. The contributions and novelty of the article are:

- We present some of the most used and well-known attacks which could harm a critical infrastructure and cause serious problems and damages;
- We present a short analysis of machine learning and deep learning models and methods which are used in intrusion detection systems, as shown in the literature;
- We conduct several experiments by generating DoS (Denial of Service Attacks) attacks in order to measure packet loss and response delay;
- We evaluate the efficiency of several machine learning techniques against several attacks by using a publicly available dataset;
- We discuss our findings and propose several future research directions.

The rest of this research is organized as follows: In Section 2, we present the related work with critical infrastructures' vulnerabilities and threats, types of attacks, e.g., phishing, SQL injection, etc. Section 3 presents some of the most used intrusion attack methods along with the proposed IDS models from the literature that use either machine or deep learning. In Section 4, we evaluate common DOS attacks using our experimental small network section and we also present the accuracy of common ML techniques against a dataset that includes a wide variety of intrusions simulated in a military network environment. Finally, Section 5.2 includes the conclusions that we draw from this research.

2. Related Work

Critical infrastructures are a significant sector of every country and, for this reason, it is crucial to know which are the threats and vulnerabilities in such systems and possible attacks in order to find a way to prevent and confront them. The research effort presented in [6] gives emphasis on the recent SCADA systems vulnerabilities and recommends ways for the improvement of the security in crucial components of SCADA systems in industrial infrastructures. Additionally, the authors of [7] present the main threats in critical infrastructures, security measures for these threats, and an overview of the categories of cyberattack techniques. In [8], the vulnerabilities and the threats in critical infrastructures are presented, and possible solutions are recommended. In [9], the authors present a review of the existing sniffing attacks, variations of these attacks, and prevention and detection techniques. Moreover, in [10], the authors presented a review of SQL injection attacks. The next paper introduces a survey of phishing attacks [11]. The research article in [12] investigates brute force attacks which aim to find the configurations of an IoT network.

The development of several types of systems to ensure the security of critical infrastructures is the result of the need to deal with the threats and attacks in critical infrastructures such as nuclear power plants [13]. Scholars have proposed a number of technical measures that include technological tools to prevent [14,15], defend [16], detect [17], mitigate [18], and respond to cyber attacks. One way to check if a system is being attacked or an intruder has gained access to it is to detect abnormal behavior. The authors of [19] present the similarities, differences, and limitations of the most used tools for fault diagnosis and cybersecurity. In [20], the authors present a real-time anomaly-based Intrusion Detection System (IDS), which has the goal to detect attacks in industrial process levels of critical infrastructures. In [21], the authors present a survey of data mining techniques adopted to detect anomalies in data or reveal if a system attacked. The authors of [22] introduce a new method for intrusion detection that relies on an incremental clustering algorithm and adopts the DBSCAN algorithm. The authors of [23] propose a new algorithm for attack detection based on an autoencoder. In [24], the authors present a new algorithm to prevent users from phishing. Again, the DBSCAN algorithm is adopted, but in combination with a technique named RD-TIA [25], which clusters the data based on their features as phishing or legitimate, in an effort to increase the accuracy of the algorithm. An extended version of an isolation forest was introduced by the authors of [26] for fault detection in hydroelectric plants. Furthermore, in [27], a deep learning approach using a Long Short Term Memory (LSTM) architecture and Recurrent Neural Network (RNN) aims to create an intrusion detection system. Additionally, ref. [28] studies the comparison between the naïve Bayes classifier and hidden Markov model. Both models are applied to detect spam emails. In [29], a detailed analysis is presented for seven deep learning models such as Convolutional Neural Networks (CNNs), recurrent neural networks, etc., and their performances for intrusion detection are tested. The authors of [30] evaluate some of the most well-known machine learning models for intrusion detection. Finally, the research [31] presents an evaluation of the performance of restricted Boltzmann machines when they were applied to detect intruders in an anomalous network intrusion detection system.

3. Intrusion Attack Methods

Critical infrastructures, as mentioned before, have a crucial role in the functioning of society. Hence, countries try to improve their efficiency and at the same time reduce the time and production cost. However, some of the improvements that the countries apply to achieve this goal have created vulnerabilities in critical infrastructure systems. Hence, these vulnerabilities allow attackers to overcome the security of those systems and gain access to systems with privileges that are not permitted to have. Below, we give a short description of intrusion attack methods.

Brute-force attack: This is one of the most well-known attacks. The operation of this attack type is simple and relies on the computing power of the attacker's computer system. Specifically, the attacker tries to find the correct combination of username and password using an exhaustive search of passwords and usernames. Usually, these attacks use tools to find the proper letters and symbols that constitute the password and the username. An attack that belongs to this category is a Dictionary attack which tries to find the correct username and password searching through a dictionary with common words and phrases that could be the password or username.

Buffer overload: This kind of attack has as goal to overwrite the data that exists in memory to gain control of the system. More specifically, the attacker gives as input to a program—more data than the buffer can handle. As a result, the data overcome the buffer boundary and the additional data stored in adjacent memory locations. The attackers use this attack in order to cause a Denial of Service (DoS) situation or in cases where the memory is well-defined can find the part of memory where the executable code of the system is stored and replace it with their own executable code. In the second case, the attacker can take the control of the system and intervene in the program operation.

Phishing: Phishing is a fraud type attack [32] that tries to delude the users by impersonating someone else, e.g., a company, which the user (victim) trusts. Often in these attacks, the attackers send an email that seems to be legitimate but it is not. The email that the user receives contains a malware file or insecure link. Hence, the attacker hopes the user (victim) will open the attachment file or link and the malware will be installed on the victim's computer. The previous process has the result that the attacker has access to sensitive data such as passwords, usernames, etc. Some of the most common types of phishing listed below are referred to in [33]:

- Clone phishing;
- Spear phishing;
- Social networking on mobile;
- Gaming phishing;
- DNS base phishing;
- Live chat;
- Whaling;
- Filter evasion.

SQL injection: In this type of intrusion attack, the attacker sets an SQL statement as an input in the application's input box in order to gain access to the database. The success of the attack results in the intruder gaining the permission to execute malicious code and harm the database or, even worse, retrieve sensitive data. The risks associated with SQL injection attacks and classification of SQL injection attack types are presented in detail.

Sniffer attack: Frequently, applications transmit packages over the transmission channel to exchange information with each other. A Sniffer attack is a process where the attacker captures, decodes, inspects and interprets the data in these packages. The context of these packages is usually passwords, usernames, etc.—i.e., important data. There are two types of Sniffer attacks, active and passive. In an active Sniffer attack, the attacker interacts with network traffic and the victim can detect if someone spies him and steal packages. On the other hand, in a passive Sniffer attack the victim does not has the ability to realize if someone performs a Sniffer attack because the attacker does not interact with network traffic.

Trojan horses: Trojan horses are programs that contain the attacker's malware file in order to camouflage the malware from the victim's systems defenses. In particular, the attackers use these programs attached to an email or in free downloaded files and programs to insert their malicious code in the victim's computer or to performs any other attack they want. Frequently, trojan horses are used by attackers as backdoors to gain access to a system.

4. Intrusion Detection Systems

Models and algorithms in intrusion detection systems: In this chapter, we present some of the most well-known algorithms and models which can be applied into an implementation of an intrusion detection system (IDS), since the existence of anomalies in data may denote that an intruder has access to our computer system. The models presented below belong to the areas of machine and deep learning.

4.1. Machine Learning Models

K-Means: K-Means is one of the most popular and used unsupervised algorithms. This algorithm tries to minimize the distance of points in a cluster with their centers. The K-Means takes as input a dataset of n data points, i.e., $D = d_0, d_1, \dots, d_n$ and the number of clusters (K) that we want to cluster the data need to be predefined. The steps of the K-Means algorithm are enumerated as follow:

- From the dataset, D are randomly chosen K data points to be the centers of the K clusters;
- Every data point in D are assigned in the cluster whose centers is nearest to the examined data point;
- After the completion of step 2, we recalculate the center of each cluster only based on the data point which belongs to the cluster;
- When the new cluster's centers are the same as the cluster's centers of previous iteration, the algorithm output the clusters. Otherwise, we iterate from step 2.

The K-Means algorithm is a simple and efficient method for intrusion detection systems because it has the ability to cluster and classify tremendous volumes of high-dimensional numerical data.

Naive Bayes classifier: The naive Bayes classifier is based on Bayes' theorem. The algorithm uses the Bayes theorem which has the ability to calculate the probability of an event occurring when we know the probability of another event that has already occurred. This probability can be calculated using the formula $P(c \mid x) = (P(x \mid c) * P(c))/(P(x))$.

In the above equation, $P(c \mid x)$ is the probability of the target class (C) given predictor (x, attributes), P(C) is the prior probability of target class (C), $P(x \mid c)$ is the likelihood which is the probability of the predictor given class and P(x) is the prior probability of the predictor. More specifically, the posterior probability is calculated constructing a frequency table for each attribute against the target class. Then, the frequency tables are transformed to likelihood tables and used in combination with Equation 1 to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction. The goal is to predict the correct class for a new instance.

K-nearest neighbors classifier: This algorithm is based on distance/similarity between two data. Specifically, the algorithm classifies a given data x using the following steps: (i) calculate the distance/similarity to all the data based on a function; (ii) sort the outcome of the function in ascending order; (iii) select the top K data of the ascending order; (iv) classify the given data x in the majority class of the top k nearest data. The most common approach of KNN uses the Euclidean distance between the data to detect the top K nearest neighbors.

DBSCAN: The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is one of the most widely used algorithms for performing clustering. DBSCAN processes the data based on the density that they have and for this reason it belongs in the category of density-based algorithms. The goal of DBSCAN is to assign the examined data points to subsets, i.e., clusters, and detect possible anomalies in the dataset. Assume a set of n points $P = p_0, p_1, \ldots, p_(n-1)$ and each data point p_i,i ϵ [0,n) belongs to some d-dimensional space R $\wedge$ d with d ϵ N. Hence, the DBSCAN, in order to cluster the points that belong in P, uses two parameters: The first is ϵ, which is the radius of the neighborhood around a data point (calculated by a distance metric) that determines the data points that are very close to a data point under consideration. The second parameter is the minimum number of points minPts that p_i has to be connected in its neighborhood to be characterized as a core point.

Decision trees: Decision trees are used for classification processes. A decision tree is designed upside down with the root at the top. It consists of the root node, the interval nodes, and leaves, and all these components are connected with branches. The root node is the beginning of the tree, each interval node represents an attribute/feature, and every leaf represents a class in which the data will be classified. The general step for the creation of a decision tree can be summarized in the following steps: (i) beginning from the root node, which contains the complete dataset, we defined as D; (ii) find the best attribute/feature in the dataset based on an Attribute Selection Measure (ASM); (iii) divide the D into subsets that contain possible values for the best attributes/features; (iv) construct the internal node, which contains the best attribute; (v) the algorithm repeats steps three and four recursively to construct new internal nodes using each time the subsets were created from the third step. The algorithm stops when it cannot further classify the data, abd the last node is the leaf node which contains the classes of the data. The paper [34] contains an anomaly-based intrusion detection system using a CART decision tree.

SVM: Support vector machine is a supervised machine learning classification algorithm which is usually used in IDS systems. SVM is mostly used for binary classification problems and has as goal to find the appropriate hyperplane to maximize the distance between two classes. A hyperplane is a threshold which separates the data into two classes in the best way. In a two-dimensional space, we can image the hyperplane as a line which separates the data into two groups/classes. Hence, SVM makes sure to select the appropriate hyperplane. The data points which are closest to the hyperplane are called support vectors. SVM is used in both linear separable cases and linear non-separable cases. In linear separable cases, SVM plots the data in a n-dimensional space where each feature of data corresponds to one dimension and tries to find the hyperplane that maximizes the distance between the two classes. On the other hand, in linear non-separable cases, SVM introduces two concepts: soft margin and kernel tricks. SVM uses the soft margin approach in an effort to balance the trade-off between the maximization of distance between the classes and the misclassification. Alternatively, SVM adopts kernel trick, where the kernel maps the data into a new higher-dimensional space so that the original non-linear data can be separated.

Isolation forest: This algorithm is used to identify the anomaly data creating decision trees over random attributes. The idea behind the algorithm is that if a forest of random decision trees produces shorter path lengths for some points, then these points are highly likely to be anomalies. The algorithm starts with the training phase, i.e., the construction of the isolation trees. In the first step, a subset of training dataset is selected. The second step of the construction is to randomly choose an attribute r and a random value of this attribute between its min and max values; this value is called the split value. In the third step, a data point is selected and if it has a value smaller than the "split value" for the attribute r, then that point is sent to the left branch. Otherwise, the point is sent to the right branch. The second and the third steps are repeated recursively over the subset until a complete data point isolation, or a predetermined tree depth limit is reached. After the training phase is completed, the testing phase begins. In this phase, every examined data x has to pass over all isolation trees to obtain its path length h(x). The anomaly score for the x is calculated as follows:

$$s(x,n) = 2^{-E(h(x))/c(n)} \tag{1}$$

where

$$E(h(x)) = \Sigma_{i=1}^{t} h_i(x)/t \tag{2}$$

is the average path length of x over t isolation trees and $c(n)$ is the average path length of unsuccessful search in Binary Search Tree.

$$c(n) = 2H(n-1) - (2(n-1)/n)$$

with

$$H(i) = ln(i) + \gamma$$

where γ is the Euler's constant. Generally, if the score is close to 1, then the examined data point is considered as an anomaly. Otherwise, if the anomaly score is smaller than 0.5, it is considered as a normal datum.

4.2. Deep Learning Models

Autoencoders: Autoencoders belong to unsupervised learning algorithms. An autoencoder in its simplest form consists of three components, an encoder, code, and decoder. An encoder is a feed-forward, fully connected neural network which has a goal to compress the input data vector into a latent space representation and encode it in a reduced dimension. The code contains the reduced data vector and sets it as input to the decoder. The decoder has the same structure but inversed, i.e., the first layer of the decoder has the same size as the last layer of the decoder. The operation of an autoencoder starts with the transformation of the input data vector into lower dimensions (encoder). The output of the encoder is stored in the code. After that, the autoencoder tries to reconstruct the initial input from the compressed data vector (decoder). There are many types of autoencoders. There are many types of autoencoders. Below we refer to some of the most used autoencoders types: convolutional autoencoders, variational autoencoders, denoising autoencoders and deep autoencoders.

RNN: Recurrent neural networks (RNNs) are a type of artificial neural network used in cases of processing sequential and time-series data. RNNs take as input a sequence of data and, from them, produce a sequence of outputs. The difference between the classical neural networks and the RNNs is depicted with the presence of a "hidden" state vector which represents the context based on prior input(s)/output(s). This means that the output depends on two parameters; the current input and the sequence of previous inputs. A simple implementation of RNN can be mathematically formulated by the following equations:

$$h_t = \sigma^{(h)}(W^{(h)}h_{t-1} + W^{(x)}x_t) \tag{3}$$

$$y_t = \sigma^{(y)}(W^{(y)}h_t) \tag{4}$$

In these equations, x_t represents the input for the current timestamp t, h_t and h_{t-1} represent the hidden state vector for the current and previous timestamps, respectively. Additionally, the dense matrices are defined by $W^{(h)}, W^{(x)}, W^{(y)}$ and the activation functions are represented by $\sigma^{(h)}, \sigma^{(y)}$. One important characteristic is that the RNNs have the same weight parameter within each layer of the network in contrast with traditional neural networks.

LSTM: Long short-term memory networks are a type of recurrent neural network (RNN) which have the ability to overcome the long-term dependency issue of recurrent networks. Many times, information from previous timestamps have important effects on the output of a model. Due to the ability of LSTMs to remember information for long time, LSTMs are a common choice for time-series models. A classical LSTM consists of four neural network layers. Each LSTM module contains a forget gate, an input gate, an output gate and a cell state. The basic component is the cell state which passes through the repeating modules. The forget gate decides how much of the memory from the previous module should be maintained. The input gate takes into consideration the input at current timestamp, the output of the previous output and combines them with an output activation function. The output gate decides based on the information from the input at current timestamp, the previous output and an output activation function the new output.

1D convolutional neural networks (CNN): CNNs are a deep learning model of feed-forward neural-networks. The most common version of CNNs is 2D CNNs which are applied in image processing. However, in intrusion detection systems they can be more effective than 1D CNNs due to the way they process the data. More specifically, 2D CNNs kernel move horizontally across the data whereas 1D CNNs move vertically. The architecture of CNNs contains two types of layers CNN-layers and MLP-layers. In CNN layers, the 1D convolutions and sub-sampling functions are applied. The list of hyper-parameters below forms the configuration of a 1D CNN.

- Number of hidden CNN and MLP layers;
- Kernel size in each CNN layer;
- Subsampling factor in each CNN layer;
- The chosen pooling and activation functions.

The paper [35] include a survey in 1D convolutional neural networks and present the applications in which they can be applied.

5. Evaluation of Attacks and Detection Mechanisms

The purpose of our section is to generate DoS (Denial of Service) attacks as well as deploy countermeasures against them. Both the attacks and their results will be presented in detail along with some detection mechanisms.

5.1. Attacks

Denial of service attacks aim to deny machine or network resources to certain users by interrupting the services of the host that are connected to the Internet. Such attacks are usually carried out by flooding a machine with unnecessary requests, overloading it, thus preventing the fulfillment of real requests. The scripts are located in Github Source Code (https://github.com/DeStC3/DosAttacksAndCountermeasures, accessed on 25 August 2021).

DDoS attacks are similar, except that incoming requests to the victim now come from multiple sources (distributed), making DDoS attacks more efficient, as it is almost impossible to block all target destinations of the attack.

All of these attacks are usually a derivative of some specialized tools available on many Linux operating systems, or programming language packs that take advantage of the basic principles of network structures and their principles used in an OS to easily generate scripts/attacks. In the simplest variations of these, the technique is usually to send a large volume of packets to the target, while in more complex ones, botnets or tools such as MyDoom or Slowloris can be used, along with sending packets of various protocols.

UDP Flood: A UDP Flood attack floods the target with a large number of User Datagram (Protocol) protocol packages. The general goal is to find random ports of the system, with the result that it repeatedly tries to check the application listening to this port; when none is found, then it sends back, in response, an ICMP package with the information "Destination Unreachable". Obviously in large volumes, network resources start to run out of downloads and send, possibly leading to denial of access. Our version of a UDP Flood follows this simple standard, creating a udp packet and then sending it to the target for a specific period of time. Here, we assume that we already know the IP of the target as well as which ports are open in its system.

SYN Flood: SYN Flood exploits a known vulnerability in the TCP connection sequence, the three way handshake, where a SYN request to start a TCP connection must receive an SYN-ACK response from the host and then send back an ACK response. In such an attack, the request sender sends many SYN requests, but either does not respond to SYN-ACK with its own ACK, or sends requests from fake IPs. In any case, the host system waits for the response to each request, freezing resources until it can not create new connections, leading to denial of service.

To create our own attack, we used Python's scapy package, which is widely used in such cases. Initially, fake IPs are created for a number of packets, as well as TCP packets that are sent. Due to problems using the scapy classes, the addresses and packet data were based on the template, but were inserted directly into the shipment function as it was the only suggested solution that bypassed the object creation problem.

ICMP Attack: Similar to UDP attacks, ICMPs send a large number of echo requests without expecting a response. They are able to consume incoming and outgoing bandwidth, as the system is forced to constantly send back reply packets, delaying it considerably. Our ICMP attack module creates and sends packets, the number of which is user-defined and determines the duration of the attack at a rate of about 10 to 1 (10 packs per second).

Ping of Death: An attack of this type typically involves sending malicious or malicious pings to a computer. The maximum size of an IP packet is about 64 bytes, including the header. However, the data link layer sets some upper limits on size, usually the maximum frame size. So, sometimes a large package is split into smaller fragments when sent, while the recipient reassembles them into a whole, which can create a buffer overflow on the memory pieces assigned to the package by creating denial of service for regular data packets. and allowing malware to be installed. For this attack, which runs on a number of packets, a false IP address is created each time it is used to create the packet before it is sent to the recipient.

5.2. Evaluation Results

The attacks were carried out on two machines which are in the same network, connected to two different router devices. For convenience, the target machine has its firewall turned off while the only processes running in the background are those that monitor the system network devices (nload, netstat, tcpdump) and the process that measures the device network response (ping in the address google.com). At the same time, the fact that the router/modem devices that provide the internet connection have their own firewalls that are locked to the active one by the provider must be taken into account. When the system is in a healthy state, the response time for the ping google.com command is 68–69 ms depending on the execution time with no data loss.

The results of the attacks vary, depending on the type of each (see Figures 1 and 2). The Ping of Death and SYN Flood attacks proved to be stronger, with the former creating a very large percentage of packet loss (80%), which makes sense considering its model as well as the size of the packages shipped, while SYN creates the longer network response delays (up to 500 ms) due to the commitment of network resources to validate the handshake for each TCP packet.

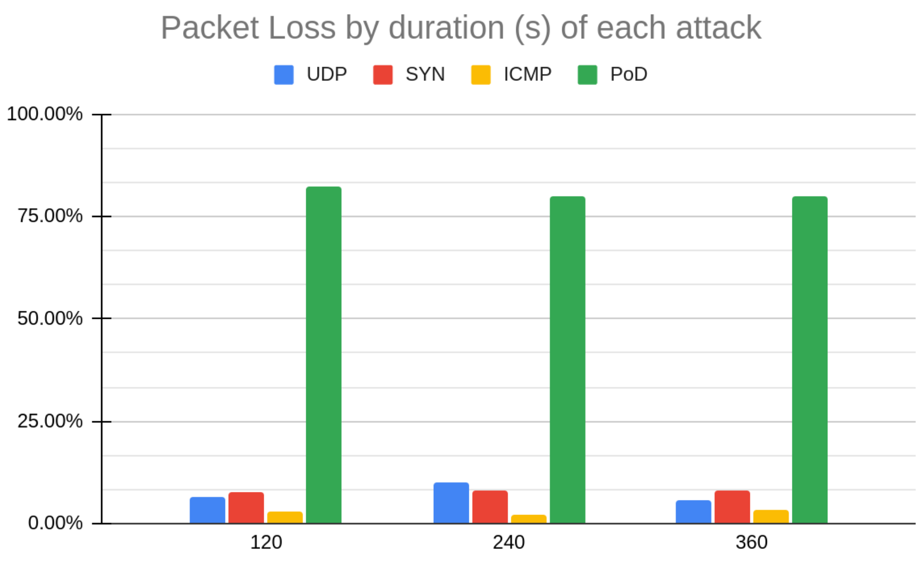

Figure 1. Packet loss during attacks.

The UDP and ICMP attacks proved to be less effective, a phenomenon based on the simplistic model they follow. Of course, the ICMP attack crisis must also be based on the settings of the target machine. A conventional machine (PC) has very few ports listening to the ICMP protocol and more TCP/UDP so most packets do not reach the target effectively. Of course, if the target was a configured mail-server, obviously the configured ports listening in the mail protocol would be higher in number and the attack would be more efficient.

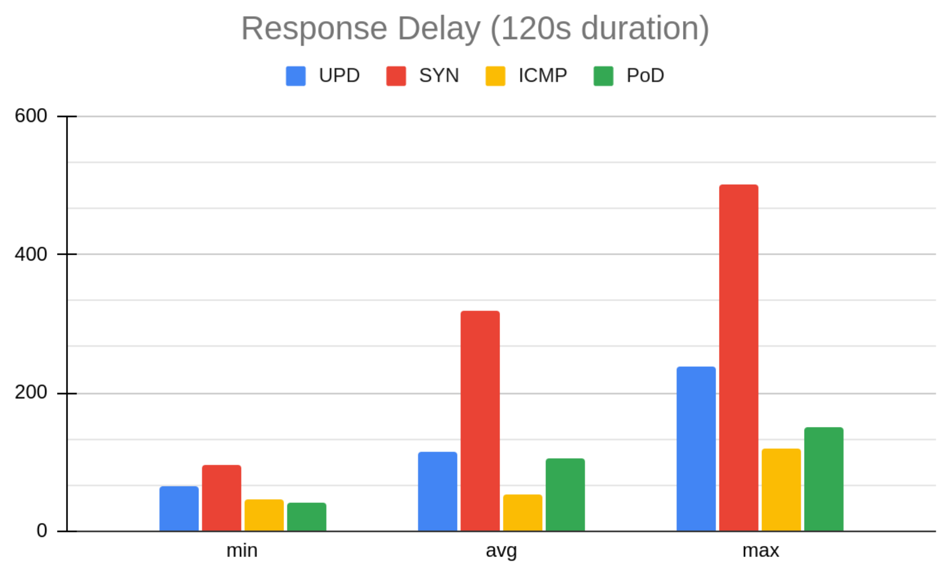

Figure 2. Response delay during attacks.

5.3. *Intrusion Detection Systems*

Finally, we decided to test several ML methods against several attacks since usually the implementation of an IDS system follows a more dynamic approach using different machine learning models. By choosing a dataset that contains a number of malicious and normal connections to a network, we can train a system so that by inputting a link with the same number of arguments as the dataset, it can decide and inform the administrator of a network for malicious connections and packets. We decided to implement an IDS which will use machine learning to detect suspicious links and test it on the KDD CUP 1999 Dataset (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, accessed on 25 August 2021) We used several algorithms (Gaussian naïve Bayes, decision tree, random forest, support vector classifier, logistic regression, gradient descent) and the findings are shown in Figure 3.

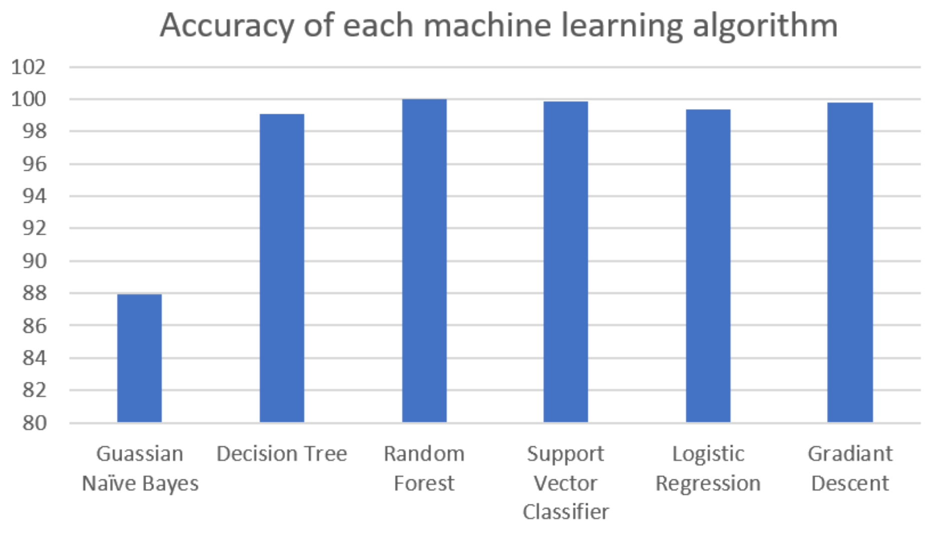

Figure 3. Detection accuracy.

6. Conclusions—Discussion

Over recent years, the need for shielding critical infrastructure has become more and more and more urgent. The security of the product or service produced, through one almost completely automated process, and the elimination of labor opportunity accidents caused by cyber attacks and at the same time their full cooperation with automations and automated processes that take place during the production process perform timely and valid detections of cyber attacks on and cyber intrusion into a critical infrastructure, which is a rather difficult achievement. The most modern studies discuss the application of intelligent algorithms using artificial intelligence (AI), which will be modeled on a set of rules set out in by the managers or the system itself and will evolve through productive processes using machine learning (ML). To date, it has not been established whether theory can meet practice in real conditions of an infrastructure that manages vital products or services and whether, during the production process, different suppliers of devices, as well as sensors of low technical specifications and computing power, are used.

In this research, we investigated intrusion detection in critical infrastructures. Intrusion detection systems are the first line of defense against intrusion attacks. More specifically, we referred to the definition of critical infrastructures and the significant role that they have for a country. Additionally, we mentioned some of the most used and well-known attacks which could harm a critical infrastructure and cause serious problems and damages to it and by extension to the sectors of a country such as the economy, etc. Furthermore, we present a short analysis of machine learning and deep learning models and methods which are used in intrusion detection systems, as shown in the literature. We comprehended the important role of the security of critical infrastructures from cyber attacks and other threats. However there are many challenges in the security of critical infrastructure and generally in cybersecurity. A significant part of the implemented solutions does not focus on unsolved important problems, such as the development of lightweight intrusion detection systems that are able to work in devices with limited power supply, false alarm control, the reduction in false positives and false negatives number and DoS attacks. Moreover, dynamic IDSs that can cope with altering conditions of the system must be further examined and analyzed [36]. Another important future direction in intrusion detection systems is the application of deep learning approaches in combination with a proper dataset to produce valid results.

Author Contributions: Conceptualization, F.P., K.T. and L.M.; methodology, K.T., K.G. and L.M.; software, K.T., K.G. and M.A.F.; validation, M.A.F. and L.M.; formal analysis, F.P., M.A.F. and L.M.; investigation, K.T., K.G., F.P. and M.A.F.; resources, K.T., K.G. and L.M.; data curation, K.G., K.T., F.P. and M.A.F.; writing—original draft preparation, K.T., K.G. and F.P.; writing—review and editing, M.A.F. and L.M.; visualization, M.A.F. and L.M.; supervision, M.A.F. and L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The scripts are located on Github Source Code (https://github.com/DeStC3/DosAttacksAndCountermeasures, accessed on 25 August 2021). Additional data that are presented in this study are available on request from the corresponding author.

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

References

1. Çimen, H.; Palacios-García, E.J.; Kolaek, M.; Çetinkaya, N.; Vasquez, J.C.; Guerrero, J.M. Smart-Building Applications: Deep Learning-Based, Real-Time Load Monitoring. *IEEE Ind. Electron. Mag.* **2020**, *15*, 4–15. [CrossRef]
2. Santiago, I.; Moreno-Munoz, A.; Quintero-Jiménez, P.; Garcia-Torres, F.; Gonzalez-Redondo, M. Electricity demand during pandemic times: The case of the COVID-19 in Spain. *Energy Policy* **2021**, *148*, 111964. [CrossRef]
3. Coffey, K.; Maglaras, L.A.; Smith, R.; Janicke, H.; Ferrag, M.A.; Derhab, A.; Mukherjee, M.; Rallis, S.; Yousaf, A. Vulnerability assessment of cyber security for SCADA systems. In *Guide to Vulnerability Analysis for Computer Networks and Systems*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 59–80.
4. Lewis, T.G. *Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation*; John Wiley & Sons: Hoboken, NJ, USA, 2019.
5. Kushner, D. The real story of stuxnet. *IEEE Spectr.* **2013**, *50*, 48–53. [CrossRef]
6. Upadhyay, D.; Sampalli, S. SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations. *Comput. Secur.* **2020**, *89*, 101666. [CrossRef]
7. Maglaras, L.; Ferrag, M.; Derhab, A.; Mukherjee, M.; Janicke, H.; Rallis, S. Threats, countermeasures and attribution of cyber attacks on critical infrastructures. *EAI Endorsed Trans. Secur. Saf.* **2018**, *5*, e1. [CrossRef]
8. Robles, R.J.; Choi, M.k.; Cho, E.s.; Kim, S.s.; Park, G.C.; Lee, J. Common threats and vulnerabilities of critical infrastructures. *Int. J. Control Autom.* **2008**, *1*, 17–22.
9. Prabadevi, B.; Jeyanthi, N. A review on various sniffing attacks and its mitigation techniques. *Indones. J. Electr. Eng. Comput. Sci* **2018**, *12*, 1117–1125. [CrossRef]
10. Halfond, W.G.; Viegas, J.; Orso, A.; others. A classification of SQL-injection attacks and countermeasures. In Proceedings of the IEEE International Symposium on Secure Software Engineering, Raleigh, NC, USA, 7–10 November 2006; Volume 1, pp. 13–15.
11. Alabdan, R. Phishing attacks survey: types, vectors, and technical approaches. *Future Internet* **2020**, *12*, 168. [CrossRef]
12. Stiawan, D.; Idris, M.; Malik, R.F.; Nurmaini, S.; Alsharif, N.; Budiarto, R.; others. Investigating brute force attack patterns in IoT network. *J. Electr. Comput. Eng.* **2019**, *2019*, 4568368. [CrossRef]
13. Tewari, A.; Gupta, B.B. Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework. *Future Gener. Comput. Syst.* **2020**, *108*, 909–920. [CrossRef]
14. Ferrag, M.A.; Maglaras, L.; Derhab, A.; Janicke, H. Authentication schemes for smart mobile devices: Threat models, countermeasures, and open research issues. *Telecommun. Syst.* **2020**, *73*, 317–348. [CrossRef]
15. Wen, H.; Tang, J.; Wu, J.; Song, H.; Wu, T.; Wu, B.; Ho, P.H.; Lv, S.C.; Sun, L.M. A cross-layer secure communication model based on Discrete Fractional Fourier Fransform (DFRFT). *IEEE Trans. Emerg. Top. Comput.* **2014**, *3*, 119–126. [CrossRef]
16. Mishra, A.; Gupta, N.; Gupta, B. Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. *Telecommun. Syst.* **2021**, *77*, 47–62. [CrossRef]
17. Hamedani, K.; Liu, L.; Atat, R.; Wu, J.; Yi, Y. Reservoir computing meets smart grids: Attack detection using delayed feedback networks. *IEEE Trans. Ind. Inform.* **2017**, *14*, 734–743. [CrossRef]
18. Bhushan, K.; Gupta, B.B. Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1985–1997. [CrossRef]
19. Ayodeji, A.; Liu, Y.k.; Chao, N.; Yang, L.q. A new perspective towards the development of robust data-driven intrusion detection for industrial control systems. *Nucl. Eng. Technol.* **2020**, *52*, 2687–2698. [CrossRef]
20. Clotet, X.; Moyano, J.; León, G. A real-time anomaly-based IDS for cyber-attack detection at the industrial process level of critical infrastructures. *Int. J. Crit. Infrastruct. Prot.* **2018**, *23*, 11–20. [CrossRef]
21. Agrawal, S.; Agrawal, J. Survey on anomaly detection using data mining techniques. *Procedia Comput. Sci.* **2015**, *60*, 708–713. [CrossRef]

22. Oh, S.H.; Lee, W.S. Anomaly intrusion detection based on dynamic cluster updating. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Nanjing, China, 22–25 May 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 737–744.
23. Bae, G.; Jang, S.; Kim, M.; Joe, I. Autoencoder-based on anomaly detection with intrusion scoring for smart factory environments. In Proceedings of the International Conference on Parallel and Distributed Computing: Applications and Technologies, Jeju Island, Korea, 20–22 August 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 414–423.
24. Jeong, S.Y.; Koh, Y.S.; Dobbie, G. Phishing detection on Twitter streams. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Auckland, New Zealand, 19–22 April 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 141–153.
25. Zhou, W.; Wen, J.; Koh, Y.S.; Alam, S.; Dobbie, G. Attack detection in recommender systems based on target item analysis. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 332–339.
26. de Santis, R.B.; Costa, M.A. Extended Isolation Forests for Fault Detection in Small Hydroelectric Plants. *Sustainability* **2020**, *12*, 6421. [CrossRef]
27. Kim, J.; Kim, J.; Thu, H.L.T.; Kim, H. Long short term memory recurrent neural network classifier for intrusion detection. In Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), Jeju, Korea, 15–17 February 2016; pp. 1–5.
28. Gomes, S.R.; Saroar, S.G.; Mosfaiul, M.; Telot, A.; Khan, B.N.; Chakrabarty, A.; Mostakim, M. A comparative approach to email classification using Naive Bayes classifier and hidden Markov model. In Proceedings of the 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Bangladesh, 28–30 September 2017; pp. 482–487.
29. Ferrag, M.A.; Maglaras, L.; Janicke, H.; Smith, R. Deep learning techniques for cyber security intrusion detection: A detailed analysis. In Proceedings of the 6th International Symposium for ICS & SCADA Cyber Security Research, Athens, Greece, 10–12 September 2019; pp. 126–136.
30. Almseidin, M.; Alzubi, M.; Kovacs, S.; Alkasassbeh, M. Evaluation of machine learning algorithms for intrusion detection system. In Proceedings of the 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 14–16 September 2017; pp. 277–282.
31. Aldwairi, T.; Perera, D.; Novotny, M.A. An evaluation of the performance of Restricted Boltzmann Machines as a model for anomaly network intrusion detection. *Comput. Netw.* **2018**, *144*, 111–119. [CrossRef]
32. Anwar, S.; Mohamad Zain, J.; Zolkipli, M.F.; Inayat, Z.; Khan, S.; Anthony, B.; Chang, V. From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. *Algorithms* **2017**, *10*, 39. [CrossRef]
33. Bhavsar, V.; Kadlak, A.; Sharma, S. Study on phishing attacks. *Int. J. Comput. Appl.* **2018**, *182*, 27–29. [CrossRef]
34. Radoglou-Grammatikis, P.I.; Sarigiannidis, P.G. An anomaly-based intrusion detection system for the smart grid based on cart decision tree. In Proceedings of the 2018 Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece, 23–25 October 2018; pp. 1–5.
35. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **2021**, *151*, 107398. [CrossRef]
36. Stewart, B.; Rosa, L.; Maglaras, L.; Cruz, T.J.; Simões, P.; Janicke, H. Effect of network architecture changes on ocsvm based intrusion detection system. In Proceedings of the International Conference on Industrial Networks and Intelligent Systems, Hanoi, Vietnam, 27–28 August 2016; Springer: Berlin/Heidelberg, Germany, 2016, pp. 90–100.