information

Article

A Comparative Analysis of Supervised and Unsupervised Models for Detecting Attacks on the Intrusion Detection Systems

Tala Talaei Khoei * and Naima Kaabouch

School of Electrical Engineering and Computer Science, University of North Dakota, Grand Forks, ND 58202, USA
* Correspondence: tala.talaeikhoei@und.edu

Abstract: Intrusion Detection Systems are expected to detect and prevent malicious activities in a network, such as a smart grid. However, they are the main systems targeted by cyber-attacks. A number of approaches have been proposed to classify and detect these attacks, including supervised machine learning. However, these models require large labeled datasets for training and testing. Therefore, this paper compares the performance of supervised and unsupervised learning models in detecting cyber-attacks. The benchmark of CICDDOS 2019 was used to train, test, and validate the models. The supervised models are Gaussian Naïve Bayes, Classification and Regression Decision Tree, Logistic Regression, C-Support Vector Machine, Light Gradient Boosting, and Alex Neural Network. The unsupervised models are Principal Component Analysis, K-means, and Variational Autoencoder. The performance comparison is made in terms of accuracy, probability of detection, probability of misdetection, probability of false alarm, processing time, prediction time, training time per sample, and memory size. The results show that the Alex Neural Network model outperforms the other supervised models, while the Variational Autoencoder model has the best results compared to unsupervised models.

Keywords: intrusion detection systems; artificial intelligence; smart grid; supervised learning; unsupervised learning

Citation: Talaei Khoei, T.; Kaabouch, N. A Comparative Analysis of Supervised and Unsupervised Models for Detecting Attacks on the Intrusion Detection Systems. *Information* **2023**, *14*, 103. https://doi.org/10.3390/info14020103

Academic Editor: Krzysztof Szczypiorski

Received: 6 December 2022
Revised: 24 January 2023
Accepted: 4 February 2023
Published: 7 February 2023

1. Introduction

With the exponential development of computer networks and technologies, security has become a major concern due to the numerous cyber-attacks constantly targeting networks. To address this issue, one practical solution to improve the security of networks is to use Intrusion Detection Systems (IDS) and tools to detect and prevent such network threats. IDS is a promising system that monitors a network for malicious activities or violating policies [1]. For instance, IDS in a smart grid can prevent an adversary from exploiting the network's vulnerabilities to gain illegal access to the nodes. IDS also prevent the misuse of the available grid resources. In general, IDS can be classified into three categories, namely signature-based, specification-based, and anomaly-based. In signature-based IDS, cyber-attacks can be detected using patterns in their malicious behaviors. In contrast, specification IDS can only detect deviations from malicious activities. In anomaly-based IDS, statistical measures are used to differentiate malicious behaviors from legitimate activities [2].

The majority of studies have highlighted the advantages and disadvantages of signature, specification, and anomaly-based IDS. For instance, one of the significant benefits of anomaly-based IDS over other types of IDS is the high strength of these systems to detect zero-day or multi-stages attacks. These systems also can be widely used to detect real-time cyber threats in networks, such as smart grids [3–5]. In addition, anomaly-based IDS are a better choice in comparison with other types of IDS, due to their ability to detect multi-step, blended, and sophisticated attacks. Despite the benefits of anomaly-based IDS, they have several limitations that need to be addressed, including low detection rate and high false

alarm and misdetection rates. Therefore, several techniques have been proposed to improve the efficiency of anomaly-based IDS in detecting and classifying different networks, such as smart grids. One recommended solution to address this is the use of Artificial Intelligence techniques, including machine learning (ML) models [5,6].

However, most of these studies provided results with high misdetection and false alarm rates. In addition, the performance of these techniques has been made in terms of a limited number of metrics, for instance, accuracy. In addition, some of these models were not optimized and the used datasets were not appropriately preprocessed [7]. Furthermore, some of these studies compared a few ML models or evaluated their proposed models without comparing their performance to other existing techniques. Moreover, there are a limited number of studies that investigated and evaluated unsupervised ML model performance. There are also no studies that comprehensively compare supervised and unsupervised models and their ability to detect attacks on a network, such as a smart grid.

Therefore, this paper provides a comparative analysis of several supervised and unsupervised ML models in detecting and classifying cyber-attacks on IDS systems. The supervised models investigated are Gaussian Naive Bayes, Classification and Regression Decision Trees, C-Support Vector Machines, Logistic Regression, Alex Neural Network, and Light Gradient Boosting. Unsupervised models selected for the investigation were K-means, Principal Component Analysis, and Variational Autoencoder. To train, test, and validate the supervised models, we used the dataset CICDDOS 2019. This same dataset was used for the training process of unsupervised models after removing the labeled class of data. The results were evaluated in terms of Accuracy (ACC), Probability of Detection (PD), Probability of Misdetection (PMD), Probability of False Alarm (PFA), Processing Time (PRT), Prediction Time (PT), Training Per Sample (TPS), and Memory Size (M).

The remainder of this paper is as follows: Section 2 outlines the related works. Section 3 highlights the dataset, the features used, and the methods applied in this study. Section 4 provides and discusses the results. A conclusion is given in Section 5.

2. Related Work

Several recent studies have applied Artificial Intelligence techniques, specifically supervised machine learning (ML), to improve smart grid security. In the following, we discuss these studies in more detail.

2.1. Supervised Techniques

The authors of [3] compared the performance of three supervised models, namely Bagging, Boosting, and Stacking models, in detecting cyber-attacks on smart grids. Their results show that the Stacking classifier yielded better results than the other techniques. The authors of [4] applied several supervised Boosting ensembles and conventional models, including K nearest neighbor, support vector machine, Adaptive Boosting, Naïve Bayes, Categorial Boosting, and Gradient Boosting to detect intrusions on the smart grid. The Boosting ensemble classifiers yielded better performance than conventional classifiers. The authors of [5] compared the performance of four known supervised ML models for detecting intrusions in smart grids, namely Naive Bayes, Support Vector Machine, Decision Tree, and Random Forest. The results show that the Random Forest classifier provides better results than other known techniques. The authors of [6] compared the effectiveness of Decision Tree, Simple Logistic Regression, Naïve Bayes, Multi-layer perceptron, Support Vector Machine, Random Forest, and Zero Rule. Their results show that the Decision Tree classifier outperforms the other models in detecting intrusions. The authors of [8] compared Neural Networks and different types of Decision Trees for detecting network intrusions. The Classification and Regression Tree classifier yielded better results than the other models in detecting network intrusions.

The authors of [9] developed a hybrid supervised model using Extreme Boosting and Long Short-Term Memory to detect intrusions in a smart grid and compared the results to other ML models, including Classification and Regression Tree, Iterative Dichotomiser 3,

Random Forest, K nearest neighbor, and Cervical Segment 4/5. Their results indicate that the hybrid model's effectiveness is higher than that of the other models. Another study [10] compared several supervised models, including Random Forest, Naïve Bayes, Support Vector Machine, and Extreme Boosting, and their ability to detect intrusions on the smart grid. The authors indicated that Random Forest and Extreme Boosting models perform better than the other models. The authors of [11] compared several supervised models to detect cyber-attacks, including Support Vector Machine, Decision Tree, Artificial Neural Networks, K-Nearest Neighbors, Naive Bayes, and Random Forrest. The results show that random forest fairly provided better results in comparison with the other models in terms of accuracy, false alarm rate, UN-detection rate, true positive rate, and receiver operating characteristic diagram.

Few other studies focused on supervised deep learning techniques to detect intrusions on smart grids. For instance, the authors of [12] proposed a detection technique using a convolutional neural network and a long short-term memory. In [13], the authors proposed a supervised improved convolutional neural network to detect network abnormalities. In [14], the authors proposed a hybrid model using Kalman Filter and Recurrent Neural Network to detect attacks in a smart grid. This technique consists of two levels to predict and fit linear and nonlinear data and uses a fully connected module to combine the results.

2.2. Unsupervised Techniques

Few studies have been proposed to evaluate the impacts of unsupervised models on detecting cyberattacks. For example, the authors of [15] used a stacked autoencoder to detect false data injection attacks. The performance of this technique was evaluated and compared with those of the Support Vector Machine and K Nearest Neighbor. The authors of [16] used the K-means model to cluster the data and create an outlier detection model for data transmission between smart homes and power centers. The authors of [17] proposed an unsupervised technique based on the Isolation Forest model for detecting attacks on the smart grid. They extracted features using Principle Component Analysis and Isolate Forest for training, testing, and validating non-labeled data. The authors of [18] proposed anomaly-based intrusion detection using a Generative Adversarial Network. This model consists of three detection layers, network flows, Modbus/Transmission Control Protocol packets (TCP), and operational data to detect attacks.

The authors of [19] used an unsupervised deep learning model, Restricted Boltzmann Machine, to detect cyber-attacks on large-scale smart grids. The proposed model uses feature extraction and symbolic dynamic filtering to decrease the computational burden with casual interactions between subsystems. The results indicate high accuracy and true positive rates, as well as low false positive rates. The authors of [20] proposed Hierarchical Temporal Memory for real-time anomaly detection and compared their results to those of Random Cut Forest, Bayesian Change, and Relative Entropy, in terms of accuracy and scoreboard. The results indicated that their model outperforms the other models for real-time anomaly detection. The authors of [21] proposed an unsupervised model using Autoencoder and random forest to detect cyber-attacks on a smart grid. The proposed model yielded satisfactory results for classification among benign operations, natural events, and malicious vulnerabilities.

3. Methodology

Figure 1 depicts the supervised and unsupervised model workflow. As one can see in Figure 1A, the supervised model workflow consists of several steps: data acquisition, dataset assessment, model training, and optimization. Supervised models require labeled data; hence, several techniques were used for their data assessment, including data balancing, imputation, normalization, and encoding. Supervised models, namely Gaussian Naive Bayes, Classification and Regression Decision Trees, C-Support Vector Machines, Logistic Regression, Alex Neural Network, and Light Gradient Boosting are trained to detect and classify network attacks and optimized using the optimization techniques, such

as grid search and ADAM optimizer. In contrast, as shown in Figure 1B, the unsupervised models are used with an unlabeled dataset, resulting in the use of fewer data assessment techniques. Unsupervised models, namely K-means, Principal Component Analysis, and Variational Autoencoder, are evaluated based on an unknown data pattern after optimization techniques are applied. Details of materials and techniques are summarized in the following section.

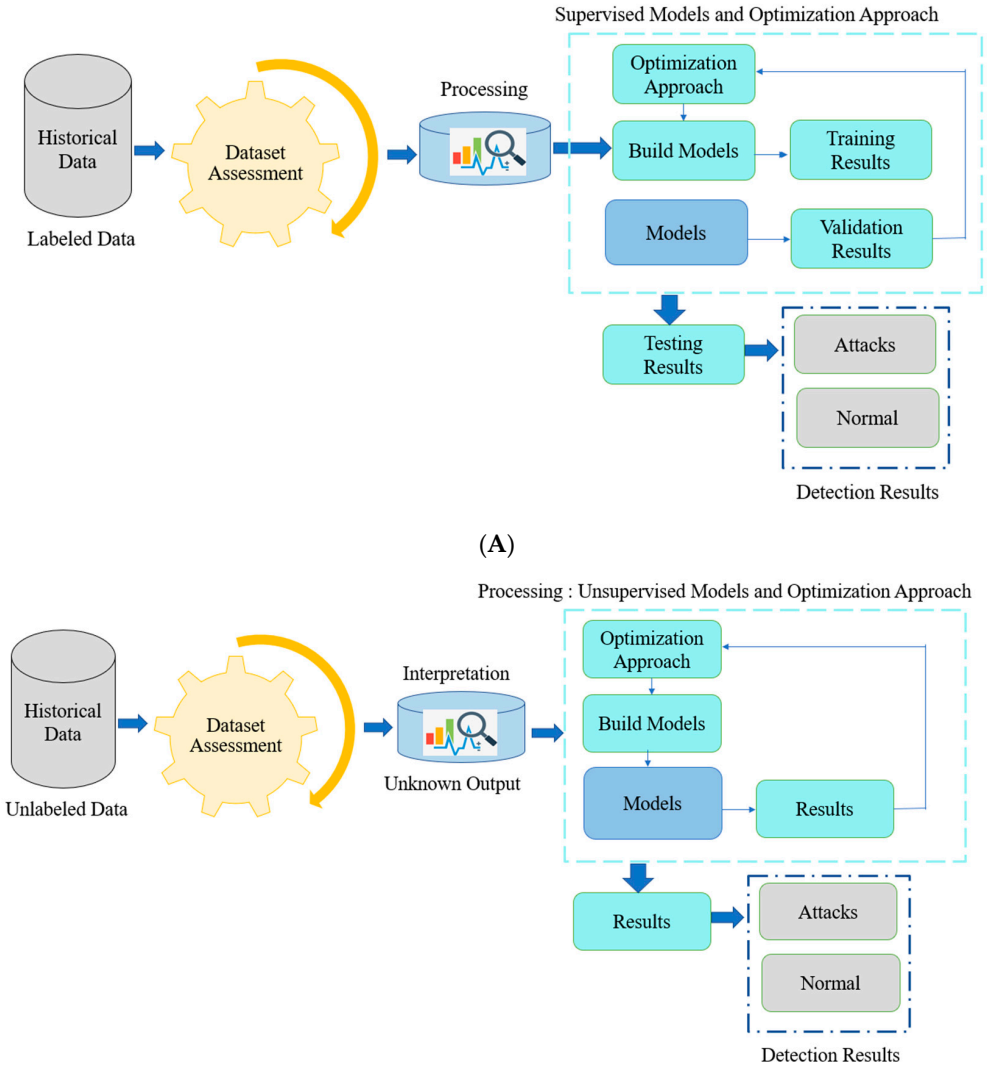

Figure 1. Supervised and Unsupervised Learning Working Flow. (**A**) Supervised Working Flow (**B**) Unsupervised Working Flow.

3.1. Dataset

We used the dataset CICDDOS 2019 [22] developed by the Canadian Institute of Cyber-Security and the University of New Brunswick. This dataset includes normal traffic samples and samples from 10 attack types. These attacks and their corresponding numbers of samples are listed in Table 1. As one can see in this table, attack classes are not balanced, which can result in inaccurate detection. To address such an issue, the lowest number of attack samples (366,461) that belongs to the UDP-lag attacks was used as a threshold for all attacks; therefore, each attack category was limited to this number. For the normal samples, we randomly selected 4,031,071, resulting in a dataset with 8,062,142 samples.

Table 1. List of Attacks.

Attacks	Number of Samples
Total Normal	5,693,110
Domain Name System (DNS)	5,071,011
Simple Network Management Protocol (SNMP)	5,159,870
Trivia File Transfer Protocol (TFTP)	20,082,580
Lightweight Directory Access Protocol (LDAP)	2,179,930,232
Network Basic Input/Output System (Netbios)	4,092,937
Microsoft SQL To Server (MSSQL)	5,781,928
Simple Service Discovery Protocol (SSDP)	2,610,611
Network Time Protocol (NTP)	1,202,649
Simple Service Discovery Protocol (SSDP)	2,610,611
User Datagram Protocol Link Aggregation (UDP-Lag)	366,461

The original dataset has 88 features and many of these do not contribute to the detection of attacks. The authors of [3] removed redundant features using Pearson's Correlation and Tree-based feature selection, resulting in 21 features, as shown in Table 2. The resulting balanced dataset with labeled samples was used for training supervised models; however, for training unsupervised models, this labeled column was removed from the dataset.

Table 2. List of Selected Features.

Features	Abbreviations
Total Length of Forward Packets	Total Length of Fwd Packets
Flow Byte(s)	Flow Byte
Flow Packet(s)	Flow Packet
Flow Inter Arrival Time Mean	Flow IAT Mean
Flow I Inter Arrival Time Std	Flow IAT Std
Flow Inter Arrival Time Max	Flow IAT Max
Forward Packets	Fwd Packets
Backward Packets	Bwd Packets
Min Packet Length	Min Packet Length
Max Packet Length	Max Packet Length
Packet Length Variance	Packet Length Variance
Total Forward Packets	Total Fwd Packets
Total Backwards Packets	Total Bwd Packets
Forward Packets Length Min	Fwd Packets Length Min
Forward Packets Length Mean	Fwd Packets Length Mean
Forward Inter Arrival Time Mean	Fwd IAT Mean
Backward Inter Arrival Time Total	Bwd IAT Total
Backward Inter Arrival Time Min	Bwd IAT Min
Backward Inter Arrival Time Mean	Bwd IAT Mean
Packet Length Mean	Packet Length Mean
Forward Packet Length Std	Fwd Packet Length Std

3.2. Data Pre-Processing

This step is used to improve the quality of data. In supervised models, this step consists of several techniques, namely missing data imputation, transformation, and encoding; however, in unsupervised learning models, data pre-processing techniques focused only on missing data imputation and transformation. A mean imputation technique was used to address the issue of null or missing values contained in the dataset. This technique replaces a missing value with the mean of all available values of that particular feature in the given dataset. The given data must also be normalized and standardized using a feature scaling technique. The features were rescaled according to the Yeo-Johnson Power Transformer, which shapes the data to appear more Gaussian and handles zero, positive, and negative values.

3.3. Machine Learning Models

We investigated several supervised and unsupervised models. Figure 2 provides a classification of these learning approaches. We selected the best model from each category for this study.

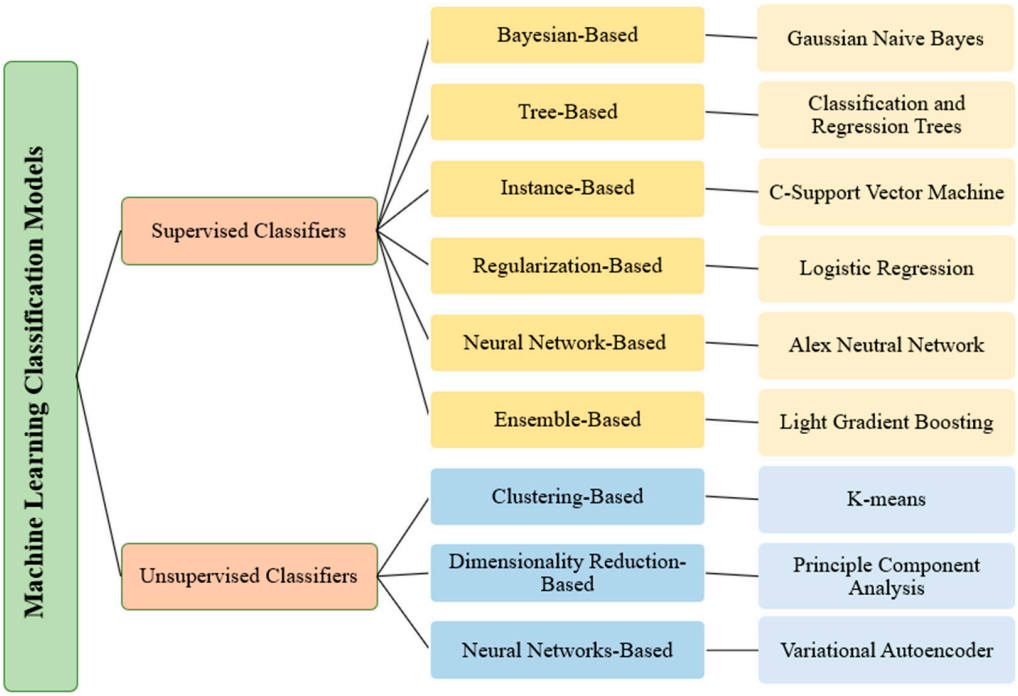

Figure 2. Classification of ML models used in this study.

3.3.1. Supervised Models

The selected supervised models are Gaussian Naïve Bayes (GNB), Classification and Regression Tree (CART), C-Support Vector Machine (C-SVM), Logistic Regression (LR), Alex Neural Network (AlexNet), and Light Gradient Boosting (LightGBM). From the Bayesian-based category, GNBwidely used data with a Gaussian normal distribution is selected [23,24]. The selected CART model, from the Tree-based category, uses a Gini index as a splitting criterion and cost-complexity pruning to decrease overfitting problems and improve accuracy [25,26]. C-SVM, from the instance-based category, stores the training data without preprocessing the target function [27,28]. The LR model, from the Regularization-based category, can be used for appropriately fitting a function on the training set and preventing overfitting problems by adding extra information to the models [29,30]. Alex Neural Network (AlexNet), from the neural-network-based category, consists of 25 layers, input, rectified linear units (ReLU), convolutional, max pooling, normalization, dropout, SoftMax, and output layers [30]. The ReLU activation function enables a faster training

process compared to other activation functions. In addition, such a function has lower computational costs without losing any generalization abilities [31–36]. Light Gradient Boosting (LightGBM), from the ensemble-based category, is based on three models, providing higher efficiency and faster training, lower memory usage, and better accuracy than other boosting models [37,38].

3.3.2. Unsupervised Models

From the unsupervised models, as highlighted in Figure 2, K-means clustering (K-means), Principle Component Analysis (PCA), and Variational Autoencoder (VA-Encoder) were selected. The K-means model, clustering-based, aims to select centroids that minimize within-cluster sum-of-square criterion (inertia). Principle Component Analysis (PCA), dimensionality reduction-based, is widely used to increase the performance of models on highly correlated data [39]. Variational Autoencoder (VA-Encoder), neural network-based, uses a compressed representation of the raw data [40]. VA-Encoder comprises three components: encoder, decoder, and loss function. This model yields a probabilistic approach to explain an observation in latent space. One of the primary benefits of using the VA-Encoder is its ability to prevent overfitting issues that guarantee that the latent space has good features with the generative process [41,42].

3.4. Optimization Approaches

Optimization approaches are necessary to obtain optimal results and decrease the ML model costs. We used two techniques in this study. These optimization techniques must be compatible with the main ML model characteristics. The grid search optimization approach was used for convolutional and ensemble models. Different combinations were investigated with a cross-validation technique in the grid search. The final result is the combination of parameters with the highest average score [36,37]. An adaptive moment estimation (ADAM) optimizer was selected for the neural network-based techniques AlexNet and VA-Encoder [38]. ADAM, as an extension of the Gradient Descent Optimization algorithm, can provide more efficient neural network parameters by running repeated cycles. This optimization technique can solve non-convex problems quicker with lower numbers of parameters. It is also an efficient technique to provide optimal results with large datasets [39,43]. In general, this optimizer usually provides better than any other optimizers and has a faster processing time and fewer tuned parameters.

3.5. Evaluation Metrics

The models were evaluated in terms of Accuracy (ACC), Probability of Detection (PD), Probability of Misdetection (PMD), and Probability of False Alarm (PFA). These metrics are defined as follows:

$$ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

$$PFA = \frac{FP}{(TN + FP)} \tag{2}$$

$$PD = \frac{TP}{(TP + FN)} \tag{3}$$

$$PD = \frac{TP}{(TP + FN)} \tag{4}$$

where TP is the number of correctly predicted malicious, TN denotes the number of correctly predicted normal signals, FP is the number of incorrectly predicted malicious signals, and FN is the number of incorrectly predicted normal signals.

The models were also evaluated in terms of Processing Time (PRT), Prediction Time (PT), Training Per Sample (TPS), and Memory Size (M). These metrics are defined as follows:

- *PRT* refers to the total time necessary to train, test, and validate the models.
- *PT* denotes the time taken to predict malicious signals over non-malicious signals.
- *TPS* denotes the time each sample takes to train the ML model.
- *M* is the amount of memory the ML models use during the entire period.

4. Results and Discussion

We used a 5-fold cross-validation approach to train 80% of the data and test the remaining 20%. The training data is split into five equal parts, and the model is fit into four parts in every iteration. This process is repeated five times using a subset of the dataset.

Table 3 lists the best hyperparameters based on grid search and ADAM optimizer that was used for training, testing, and validating the ML models.

Table 3. Best Parameters Settings Based on Optimization Techniques.

Model	Best Parameters
GNB	var_smoothing = 0.001
CART	Criterion = 'gini', max-depth = 36, splitter = 'best', max_features = 'log2'.
C-SVM	C = 4, penalty = 'l2'
LR	Max_iter = 12, penalty = 'l2'
AlexNet	Epoch = 100, momentum = 0.9, Batch size = 128, learning_rate = 0.01.
LightGBM	Boosting_type = 'gbdt', max_depth = 10, learning_rate = 0.1, n_estimators = 100
PCA	max-depth = 10, Max-features = 'sqrt', splitter = 'best', Criterion = 'entropy'.
K-means	n-clusters = 2, algorithm = 'auto', random-state = 0.
VA-Encoder	Loss = 'mse', Activation = 'Relu', Epoch = 100

Figures 3 and 4 present the results of the ML models in terms of accuracy, probability of detection, probability of misdetection, and *PFA*. The AlexNet model yielded the best results in terms of the selected metrics among supervised models (Figure 3). LightGBM yielded a slightly lower *ACC* and *PD* and higher *PMD* and *PFA* compared to the AlexNet model. The other supervised models, CART and C-SVM, also had satisfactory results. However, the LR and GNB models yielded the worst results among the supervised models.

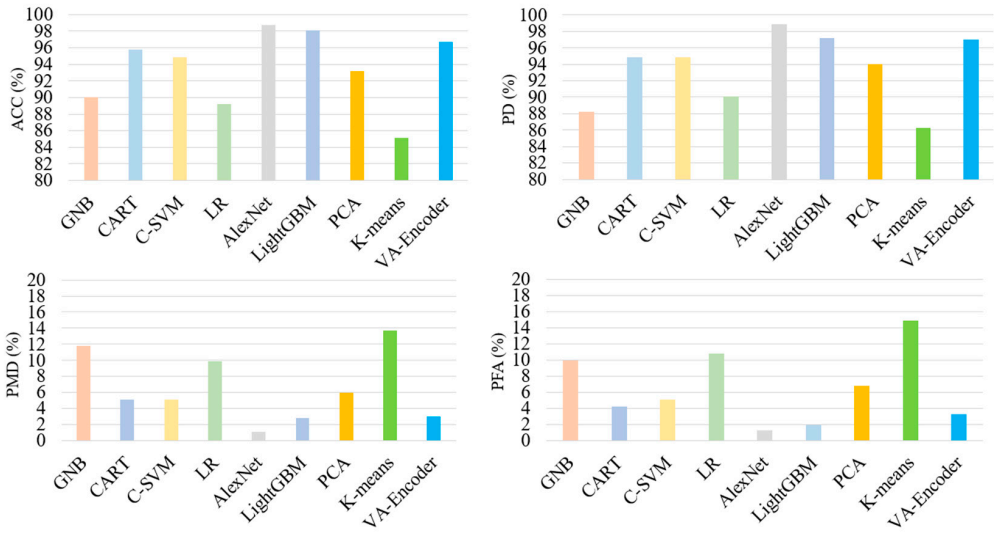

Figure 3. Performance evaluation of the ML models in terms of *ACC*, *PD*, *PMD*, and *PFA* for Test Data.

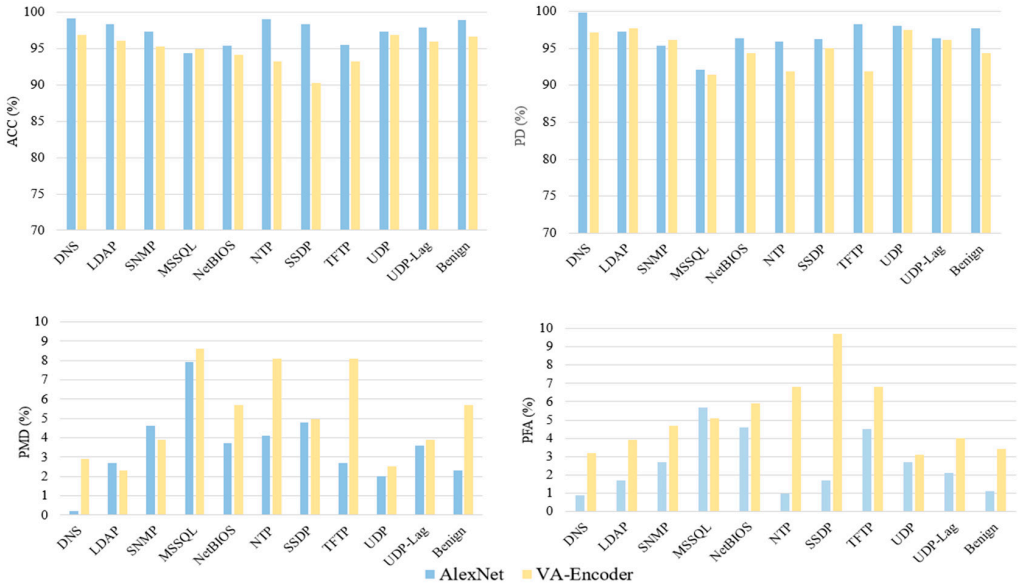

Figure 4. Performance evaluation of cyber-attacks based on best ML models in terms of ACC, PD, PMD, and PFA.

In contrast, the unsupervised models exhibited significantly lower performance in terms of the same metrics. The VA-encoder model yielded the highest performance compared to the other unsupervised models. The PCA model yielded considerably lower performance than the VA-Encoder. The K-means model had the lowest *ACC* and *PD* and the highest *PMD* and *PFA*.

Comparing the supervised and unsupervised models, the AlexNet model yielded the best results, followed by LightGBM, VA-Encoder, CART, C-SVM, PCA, GNB and LR, and K-means.

Table 4 illustrates the model results in terms of the other four metrics. The AlexNet model has the best *PRT*, *PT*, *TPS*, and *M* compared to the other supervised and unsupervised models, while the GNB model had the worst performance in terms of these metrics. The CART model yielded slightly higher results in terms of *PRT*, *PT*, *TPS*, and *M* than the AlexNet model. The VA-encoder model yielded the best performance among the unsupervised models, while K-means has the lowest performance.

Table 4. The ML models' performance in Terms of PRT, PT, TPS, and M for Test Data (best performances are in bold).

Model	PRT (S)	PT (S)	TPS (S)	M (MiB)
GNB	4.33	4.15	0.82	245
CART	1.2	1.1	0.2	132
C-SVM	2.9	1.8	0.39	236
LR	1.6	1.2	0.51	223
AlexNet	**1.01**	**1**	**0.01**	**102**
LightGBM	1.4	1.3	0.09	112
PCA	1.9	0.91	0.89	164
K-means	1.9	1.4	0.81	180
VA-Encoder	**1.77**	**1.2**	**0.5**	**144**

Therefore, the AlexNet model has the best results among the supervised models, whereas the VA-Encoder yielded the best results among the unsupervised models in terms of *ACC*, *PD*, *PMD*, *PFA*, *PRT*, *PT*, *TPS*, and *M*.

Figure 4 represents the detection results of individual attacks for the two best-selected models. AlexNet and VA-Encoder, in terms of *ACC*, *PD*, *PMD*, and *PFA*. AlexNet outperformed the VA-Encoder model when detecting cyber-attacks. For example, the DNS attacks were detected with better performance using AlexNet compared to VA-Encoder. AlexNet detected these attacks with an *ACC* of 99.13%, a *PD* of 99.81%, a *PMD* of 0.19%, and a *PFA* of 0.93%. The VA-Encoder detected the same attacks with considerably lower performance with an *ACC* of 96.83%, a *PD* of 97.11%, a *PMD* of 2.89%, and a *PFA* of 3.23%. The VA-encoder detected and classified UDP attacks with the highest performance. AlexNet detected MSSQL attacks with a slightly lower performance than the VA-Encoder; however, this last model detected SSDP, NTP, and TFP with the lowest performance. In general, AlexNet outperforms the VA-Encoder in detecting most attacks.

Table 5 presents the results of AlexNet and VA-Encoder in terms of processing time, prediction time, training time per sample, and memory size. AlexNet outperformed VA-Encoder in detecting cyber-attacks. For example, DNS attacks could be detected and classified using AlexNet with significantly lower *PRT*, *PT*, *TPS*, and *M* than VA-encoder. AlexNet detected the DNS attacks with a *PRT* of 1.1 s, a *PT* of 0.9 s, a *TPS* of 0.3 s, and an *M* of 149 MiB. AlexNet detected NetBIOS attacks with the highest *PRT*, *PT*, *TPS*, and *M* compared to those of other attacks; however, VA-Encoder detected SSDP attacks with the highest *PRT*, *PT*, *TPS*, and *M* compared to those other attacks.

Table 5. Performance of the ML Models in terms of PRT, PT, TPS, and M for TEST data.

Models	Attacks	PRT (S)	PT (S)	TPS (S)	M (MiB)
AlexNet	LDAP	1.4	1.2	0.7	125
	DNS	1.1	0.9	0.3	149
	SNMP	1.9	1.2	0.4	123
	MSSQL	1.3	1.2	0.1	177
	NetBIOS	1.9	1.4	0.9	191
	NTP	1.2	1.7	0.6	182
	SSDP	1.1	1	0.5	173
	TFTP	1.4	1.1	0.7	167
	UDP	1.8	1.2	0.5	166
	UDP-Lag	1.3	1.1	0.2	161
	DNS	1.4	1.2	0.7	125
	Benign	1.8	1.4	0.4	180
VA-Encoder	LDAP	3.5	3.1	0.4	290
	DNS	3.4	2.9	0.4	278
	SNMP	3.2	2.3	0.9	276
	MSSQL	2.9	2.3	0.3	254
	NetBIOS	2.9	2.3	0.4	246
	NTP	2.9	1.3	0.6	277
	SSDP	3.9	2.1	0.5	297
	TFTP	3.1	2.9	0.3	289
	UDP	3.8	3.1	0.7	290
	UDP-Lag	2.9	2.7	0.2	214
	Benign	3.1	2.9	0.2	212

To demonstrate the efficiency of the proposed techniques, our results are compared with several existing studies in the literature, as provided in Table 6. As one can see, these existing studies used different datasets, such as NSL KDD, and KDDCup99, as shown in Table 6. As can be seen, these studies evaluated and analyzed their proposed models in terms of a limited number of metrics. It is also worth mentioning that the majority of these studies only focused on supervised models, while it is necessary to study the performance of unsupervised models in detecting intrusions on a smart grid. For this purpose, our study fills this gap by evaluating the performance of the best supervised and unsupervised models in detecting intrusions on a smart grid. This table shows that AlexNet and VA-Encoder outperform the other models in the literature in terms of accuracy, probability of detection, probability of misdetection, probability of false alarm, processing time, prediction time, training time per sample, and memory size.

Table 6. Comparison of related works with the proposed techniques.

References	Best Used Models	Category Models		Accuracy (%)	Detection Rate (%)	Misdetection Rate (%)	False Alarm Rate (%)	Processing Time	Training Time Per Sample (S)	Prediction Time (S)	Memory Usage (MiB)	Datasets					
		Supervised	Unsupervised									KDDCup99	NSL-KDD	DARPA	CICDDOS 2019	CICD 001/002/2018	Self-Collected
[1]	Stacking	✓	-	97.3	96	4.1	8.9	-	-	-	-	-	-	-	-	✓	-
[2]	Categorical Boosting	✓	-	97.71	96.8	5.06	3.98	-	-	-	-	-	-	-	-	✓	-
[8]	Long Short-term Memory with Extreme Boosting	✓	-	88	98	-	-	-	-	-	-	✓	-	✓	-	-	-
[9]	Random Forest	✓	-	97.01	99.7	-	-	-	-	-	-	-	-	-	-	✓	-
[11]	Isolation Forest	✓	✓	93.01	-	-	-	-	-	-	-	-	-	-	-	-	-
[12]	K-means	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-
[14]	Generative Adversarial Network	-	✓	93	87.5	-	-	-	-	-	-	-	-	-	-	-	✓
[15]	Hierarchical Temporal Memory	-	✓	96	-	-	-	-	-	-	-	-	-	-	-	-	-
Proposed Models	**Alex Net**	✓	-	**98.71**	**98.9**	**1.1**	**1.29**	**1.16**	**1.06**	**0.10**	**104.2**	-	-	-	-	✓	-
	VA-Encoder	-	✓	**96.7**	**97**	**3**	**3.3**	**1.7**	**1.23**	**0.11**	**143.2**						

To summarize, AlexNet could detect LDAP, DNS, SNMP, MSSQL, NetBIOS, NTP, SSDP, TFTP, UDP, UDP-Lag attacks, and Benign traffic better than VA-Encoder.

The key points of this study are:

- The AlexNet model yielded the best results of all supervised and unsupervised learning techniques in terms of the highlighted metrics.

- GNB and LR models yielded the worst results of the supervised models.
- The VA-Encoder model yielded the highest-performance results of the unsupervised models.
- The worst performance model among the unsupervised models was K-means.
- Several models, such as CART, C-SVM, and PCA, yielded satisfactory results.

5. Conclusions

Intrusion Detection Systems are expected to monitor and detect abnormalities on the networks. In general, studies have been performed to detect and classify attacks on Intrusion Detection Systems; however, most of these studies have focused only on supervised machine learning models. We have provided a comprehensive comparison of supervised and unsupervised models in terms of accuracy, probability of detection, probability of misdetection, probability of false alarm, processing time, prediction time, training time per sample, and memory size. Models were classified as Bayesian, Tree, Instance, Regularization, Neural Network, and Ensemble categories, and one model was chosen from each category. We used Gaussian Naive Bayes, Classification and Regression Decision Trees, C-support vector machines, logistic regression, Alex neural networks, and Light Gradient Boosting for supervised models. We used Principal Component Analysis, K-means, and Variational Autoencoder for unsupervised models. The results indicate that the Alex Neural Network outperforms other supervised and unsupervised models; however, VA-Encoder provided the best results compared to other unsupervised models. In addition, cyber-attacks can be detected better with better performance in comparison with the same attacks using Variational-Encoder. Future works include investigating the performance of supervised and unsupervised deep learning models in detecting attacks on intrusion detection systems.

Author Contributions: Conceptualization, T.T.K.; software, T.T.K.; data curation, T.T.K.; formal analysis, T.T.K.; investigation, T.T.K.; methodology, T.T.K.; visualization, T.T.K.; writing—original draft, T.T.K. and N.K.; writing—review and editing, T.T.K. and N.K.; project administration, N.K.; validation, N.K.; supervision, N.K.; funding acquisition, N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is available in a publicly accessible repository.

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

References

1. Smadi, A.A.; Ajao, B.T.; Johnson, B.K.; Lei, H.; Chakhchoukh, Y.; Abu Al-Haija, Q. A Comprehensive Survey on Cyber-Physical Smart Grid Testbed Architectures: Requirements and Challenges. *Electronics* **2021**, *10*, 1043. [CrossRef]
2. Tazi, K.; Abdi, F.; Abbou, M.F. Review on Cyber-physical Security of the Smart Grid: Attacks and Defense Mechanisms. In *International Renewable and Sustainable Energy Conference (IRSEC)*; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6.
3. Khoei, T.T.; Aissou, G.; Hu, W.C.; Kaabouch, N. Ensemble Learning Methods for Anomaly Intrusion Detection System in Smart Grid. In Proceedings of the 2021 IEEE International Conference on Electro Information Technology (EIT), Mt. Pleasant, MI, USA, 14–15 May 2021; pp. 129–135. [CrossRef]
4. Khoei, T.T.; Ismail, S.; Kaabouch, N. Boosting-based Models with Tree-structured Parzen Estimator Optimization to Detect Intrusion Attacks on Smart Grid. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 1–4 December 2021; pp. 0165–0170.
5. Mrabet, Z.E.; Ghazi, H.E.; Kaabouch, N. A performance comparison of data mining algorithms-based intrusion detection system for smart grid. In *Conference on Electro Information Technology (EIT)*; IEEE: Piscataway, NJ, USA, 2019; pp. 298–303.
6. Anthi, E.; Williams, L.; Słowińska, M.; Theodorakopoulos, G.; Burnap, P. A supervised intrusion detection system for smart home IoT devices. *Internet Things J.* **2019**, *6*, 9042–9053. [CrossRef]
7. Talaei Khoei, T.; Ismail, S.; Shamaileh, K.A.; Devabhaktuni, V.K.; Kaabouch, N. Impact of Dataset and Model Parameters on Machine Learning Performance for the Detection of GPS Spoofing Attacks on Unmanned Aerial Vehicles. *Appl. Sci.* **2022**, *13*, 383. [CrossRef]
8. Thapa, N.; Liu, Z.; Kc, D.B.; Gokaraju, B.; Roy, K. Comparison of machine learning and deep learning models for network intrusion detection systems. *Future Internet* **2020**, *12*, 167. [CrossRef]

9. Song, C.; Sun, Y.; Han, G.; Rodrigues, J.J. Intrusion detection based on hybrid classifiers for smart grid. *Comput. Electr. Eng.* **2021**, *93*, 107212. [CrossRef]
10. Roy, D.D.; Shin, D. Network Intrusion Detection in Smart Grids for Imbalanced Attack Types Using Machine Learning Models. In Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 16–18 October 2019; pp. 576–581.
11. Arora, P.; Kaur, B.; Teixeira, M.A. Evaluation of Machine Learning Algorithms Used on Attacks Detection in Industrial Control Systems. *J. Inst. Eng.* **2021**, *102*, 605–616. [CrossRef]
12. Yao, R.; Wang, N.; Liu, Z.; Chen, P.; Sheng, X. Intrusion Detection System in the Advanced Metering Infrastructure: A Cross-Layer Feature-Fusion CNN-LSTM-Based Approach. *Sensors* **2021**, *21*, 626. [CrossRef]
13. Yang, H.; Wang, F. Wireless Network Intrusion Detection Based on Improved Convolutional Neural Network. *IEEE Access* **2019**, *7*, 64366–64374. [CrossRef]
14. Wang, Y.; Zhang, Z.; Ma, J.; Jin, Q. KFRNN: An Effective False Data Injection Attack Detection in Smart Grid Based on Kalman Filter and Recurrent Neural Network. *IEEE Internet Things J.* **2022**, *9*, 6893–6904. [CrossRef]
15. Majidi, S.; Hadayeghparast, S.; Karimipour, H. FDI attack detection using extra trees algorithm and deep learning algorithm-autoencoder in smart grid. *Int. J. Crit. Infrastruct. Prot.* **2022**, *37*, 100508. [CrossRef]
16. Ahmed, S.; Lee, Y.; Hyun, S.; Koo, I. Unsupervised Machine Learning-Based Detection of Covert Data Integrity Assault in Smart Grid Networks Utilizing Isolation Forest. *IEEE Trans. Inf. Secur.* **2019**, *14*, 2765–2777. [CrossRef]
17. Menon, D.M.; Radhika, N. Anomaly detection in smart grid traffic data for home area network. In Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 18–19 March 2016; pp. 1–4.
18. Grammatikis, P.R.; Sarigiannidis, P.; Efstathopoulos, G.; Panaousis, E. ARIES: A Novel Multivariate Intrusion Detection System for Smart Grid. *Sensors* **2020**, *20*, 5305. [CrossRef] [PubMed]
19. Karimipour, H.; Dehghantanha, A.; Parizi, R.M.; Choo, K.R.; Leung, H. A Deep and Scalable Unsupervised Machine Learning System for Cyber-Attack Detection in Large-Scale Smart Grids. *IEEE Access* **2019**, *7*, 80778–80788. [CrossRef]
20. Barua, A.; Muthirayan, D.; Khargonekar, P.P.; Al Faruque, M.A. Hierarchical Temporal Memory Based Machine Learning for Real-Time, Unsupervised Anomaly Detection in Smart Grid: WiP Abstract. In Proceedings of the ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS), Sydney, Australia, 21–25 April 2020; pp. 188–189.
21. Hu, C.; Yan, J.; Liu, X. Adaptive Feature Boosting of Multi-Sourced Deep Autoencoders for Smart Grid Intrusion Detection. In Proceedings of the 2020 IEEE Power & Energy Society General Meeting (PESGM), Virtual, 3–6 August 2020; pp. 1–5.
22. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In Proceedings of the IEEE 53rd International Carnahan Conference on Security Technology, Chennai, India, 1–3 October 2019.
23. Altwaijry, H. Bayesian based intrusion detection system. In *IAENG Transactions on Engineering Technologies*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 29–44.
24. van de Schoot, R.; Depaoli, S.; King, R.; Kramer, B.; Märtens, K.; Tadesse, M.G.; Vannucci, M.; Gelman, A.; Veen, D.; Willemsen, J.; et al. Bayesian statistics and modelling. *Nat. Rev. Methods Prim.* **2021**, *1*, 1. [CrossRef]
25. Jahromi, A.H.; Taheri, M. A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features. In Proceedings of the Artificial Intelligence and Signal Processing Conference (AISP), Shiraz, Iran, 25–27 October 2017; pp. 209–212. [CrossRef]
26. Song, Y.; Ying, L. Decision tree methods: Applications for classification and prediction. *Shanghai Arch. Psychiatry* **2015**, *27*, 130.
27. Singh, S.; Gupta, P. Comparative study ID3, cart and C4. 5 decision tree algorithm: A survey. *Int. J. Adv. Inf. Sci. Technol. (IJAIST)* **2014**, *27*, 97–103.
28. Zhang, M.L.; Zhou, Z.H. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognit.* **2007**, *40*, 2038–2048. [CrossRef]
29. Musavi, M.; Ahmed, W.; Chan, K.; Faris, K.; Hummels, D. On the training of radial basis function classifiers. *Neural Netw.* **1992**, *5*, 595–603. [CrossRef]
30. Yang, X.; Zhang, G.; Lu, J.; Ma, J. A Kernel Fuzzy c-Means Clustering-Based Fuzzy Support Vector Machine Algorithm for Classification Problems With Outliers or Noises. *IEEE Trans. Fuzzy Syst.* **2011**, *19*, 105–115. [CrossRef]
31. Izeboudjen, N.; Larbes, C.; Farah, A. A new classification approach for neural networks hardware: From standards chips to embedded systems on chip. *Artif. Intell. Rev.* **2014**, *41*, 491–534. [CrossRef]
32. Wang, D.; He, H.; Liu, D. Intelligent Optimal Control With Critic Learning for a Nonlinear Overhead Crane System. *IEEE Trans. Ind. Inform.* **2018**, *14*, 2932–2940. [CrossRef]
33. Wang, S.C. Artificial Neural Network. *Interdiscip. Comput. Java Program.* **2003**, *743*, 81–100. [CrossRef]
34. 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. [CrossRef]
35. Khoei, T.T.; Hu, W.C.; Kaabouch, N. Residual Convolutional Network for Detecting Attacks on Intrusion Detection Systems in Smart Grid. In Proceedings of the 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA, 19–21 May 2022; pp. 231–237.
36. Gunturi, S.K.; Sarkar, D. Ensemble machine learning models for the detection of energy theft. *Electr. Power Syst. Res.* **2021**, *192*, 106904. [CrossRef]

37. Ismail, S.; Khoei, T.T.; Marsh, R.; Kaabouch, N. A comparative study of machine learning models for cyber-attacks detection in wireless sensor networks. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 1–4 December 2021; pp. 0313–0318.
38. Khoei, T.T.; Kaabouch, N. Densely Connected Neural Networks for Detecting Denial of Service Attacks on Smart Grid Network. In Proceedings of the 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 26–29 October 2022; pp. 0207–0211.
39. Pham, D.T.; Dimov, S.S.; Chi, N.D. Selection of K in K-means clustering. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2005**, *219*, 103–119. [CrossRef]
40. Jolliffe, T.I.; Jorge, C. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2016**, *374*, 20150202. [CrossRef] [PubMed]
41. Bock, S.; Weiß, M. A Proof of Local Convergence for the Adam Optimizer. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
42. Slimane, T.T.K.H.O.; Kaabouch, N. Cyber-Security of Smart Grids: Attacks, Detection, Countermeasure Techniques, and Future Directions. *Commun. Netw.* **2022**, *14*, 119–170.
43. Jafari, F.; Dorafshan, S. Comparison between Supervised and Unsupervised Learning for Autonomous Delamination Detection Using Impact Echo. *Remote Sens.* **2022**, *14*, 6307. [CrossRef]