electronics

MDPI

Article

Developing Cybersecurity Systems Based on Machine Learning and Deep Learning Algorithms for Protecting Food Security Systems: Industrial Control Systems

Hasan Alkahtani [1,2] and Theyazn H. H. Aldhyani [1,3,*]

[1] Al Bilad Bank Scholarly Chair for Food Security in Saudi Arabia, The Deanship of Scientific Research, The Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Al Ahsa 31982, Saudi Arabia; hsalkahtani@kfu.edu.sa
[2] College of Computer Science and Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia
[3] Applied College in Abqaiq, King Faisal University, Al-Ahsa 31982, Saudi Arabia
* Correspondence: taldhyani@kfu.edu.sa

Abstract: Industrial control systems (ICSs) for critical infrastructure are extensively utilized to provide the fundamental functions of society and are frequently employed in critical infrastructure. Therefore, security of these systems from cyberattacks is essential. Over the years, several proposals have been made for various types of cyberattack detection systems, with each concept using a distinct set of processes and methodologies. However, there is a substantial void in the literature regarding approaches for detecting cyberattacks in ICSs. Identifying cyberattacks in ICSs is the primary aim of this proposed research. Anomaly detection in ICSs based on an artificial intelligence algorithm is presented. The methodology is intended to serve as a guideline for future research in this area. On the one hand, machine learning includes logistic regression, k-nearest neighbors (KNN), linear discriminant analysis (LDA), and decision tree (DT) algorithms, deep learning long short-term memory (LSTM), and the convolution neural network and long short-term memory (CNN-LSTM) network to detect ICS malicious attacks. The proposed algorithms were examined using real ICS datasets from the industrial partners Necon Automation and International Islamic University Malaysia (IIUM). There were three types of attacks: man-in-the-middle (mitm) attack, web-server access attack, and telnet attack, as well as normal. The proposed system was developed in two stages: binary classification and multiclass classification. The binary classification detected the malware as normal or attacks and the multiclass classification was used for detecting all individual attacks. The KNN and DT algorithms achieved superior accuracy (100%) in binary classification and multiclass classification. Moreover, a sensitivity analysis method was presented to predict the error between the target and prediction values. The sensitivity analysis results showed that the KNN and DT algorithms achieved R2 = 100% in both stages. The obtained results were compared with existing systems; the proposed algorithms outperformed existing systems.

Keywords: industrial control systems; intrusion detection system; machine learning; deep learning; cyberattack

check for updates

Citation: Alkahtani, H.; Aldhyani, T.H.H. Developing Cybersecurity Systems Based on Machine Learning and Deep Learning Algorithms for Protecting Food Security Systems: Industrial Control Systems. *Electronics* **2022**, *11*, 1717. https://doi.org/10.3390/electronics11111717

Academic Editors: Leandros Maglaras, Helge Janicke and Mohamed Amine Ferrag

Received: 7 May 2022
Accepted: 24 May 2022
Published: 27 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In critical infrastructures that supply crucial services such as water, electricity, or communications, industrial control systems (ICSs) are at the heart of the operation. ICSs provide the foundational services for monitoring and controlling industrial operations. The monitoring section uses sensors to collect data, keep track of the processes, and ensure that they run properly [1]. On the one hand, the monitoring section oversees operations and ensures that they run correctly. On the other hand, the controlling portion manages the processes and makes decisions that cause actions to be carried out by actuators. If this workflow is disrupted as a result of technological difficulties or cyberattacks, many citizens may be adversely affected, for example, as a result of interruptions in electrical power or communications [2].

Information technology stacks (ITS) and remote connections are commonly used to link ICS components. An increase in the likelihood of deliberate assaults on physical plants might result from reliance on communication networks to transmit measurements. Authentication, data encryption, and message integrity procedures are just a few methods for keeping network traffic safe. Despite this, these solutions cannot defend all layers of an ICS network from all types of invasions of privacy [3].

Operational technology (OT) processes, which are critical components of infrastructure, are routinely targeted by criminal organizations. In the past, OT and information technology (IT) networks were kept separate or "air-gapped" from one another [4]. However, due to increased efficiency gained via digitalization, new business requirements are emerging, increasing the time and money spent on digitalization. However, due to digitalization's increased efficiency, new business requirements are emerging, increasing the number of organizations using the technology [5]. Over the Internet of Things (IoT), sensor and actuator data, and multimedia data such as images and videos, are transmitted. It is vital to put security measures in place to protect against malicious behavior and cyber risks. On the one hand, cybersecurity approaches, which are critical to the long-term health of supply chains, are required to ensure the safety of workers and commodities and to protect information passing via their networks, among other things. On the other hand, a cyberattack on an ICS might result in a malfunction, which could cause physical harm to other physical components or even humans [6]. A cyberattack may result in the theft of confidential information about a company's business activities; it may also have the unintended consequence of decreasing the degree of competitiveness of the industry in the long term. The percentage of malware threats to ICSs over the last four years is presented in Figure 1.

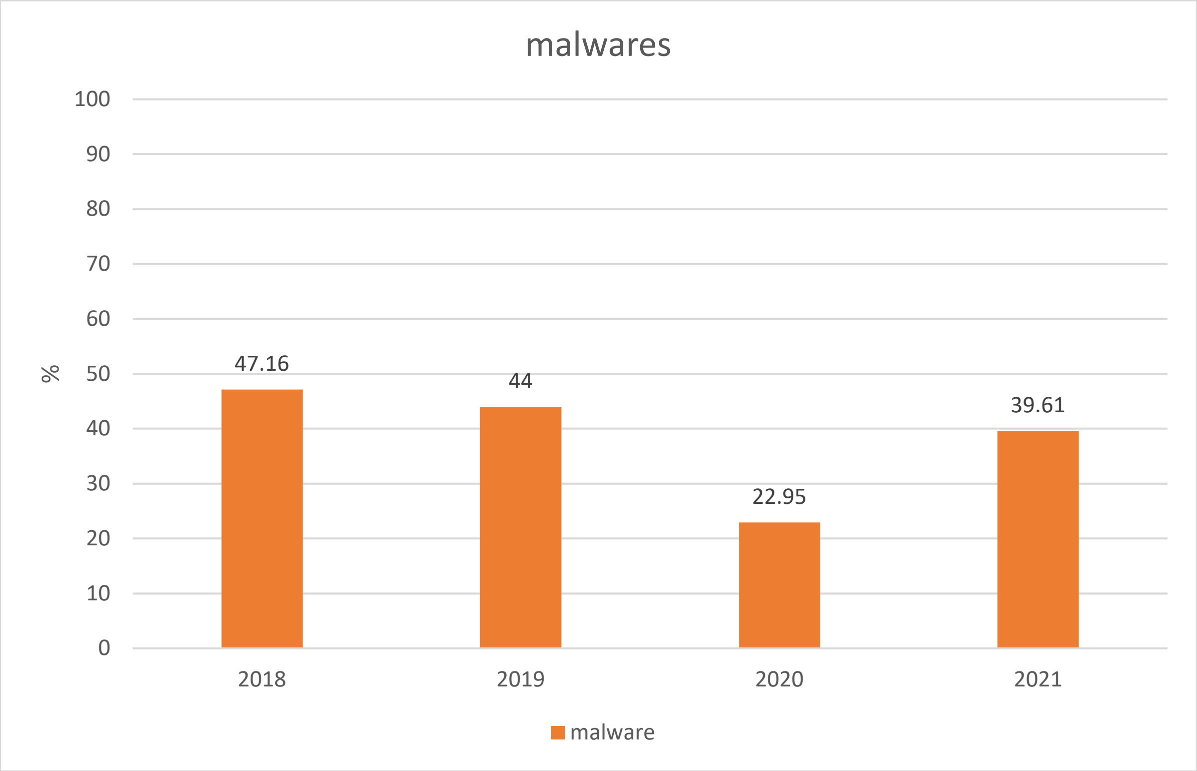

Figure 1. Malware threats to ICSs over the last four years.

An intrusion detection system (IDS) is one method of dealing with this problem. The detection methodologies used by signature-based and anomaly-based IDSs are distinct [7]. The signature-based technique instructs the system to seek specific anomalies. In contrast, the anomaly-based strategy instructs the system to look for any deviation from a previously defined standard of behavior. Most of the time, an IDS examines the network traffic of an ICS to detect irregularities in the incoming data packets. It is possible to safeguard a network from unwanted infiltration attempts by implementing a network intrusion detection system (NIDS), also known as packet filtering. The NIDS model utilized in various studies [8–12] was developed using machine learning approaches to detect network traffic intrusions. Because of its capacity to identify and quantify attacks in the network flow, even if its performance against encrypted data packets, fake IP packets, and regular false-positive alerts are not guaranteed, it is becoming increasingly popular.

Anomaly detection in industrial control systems (MADICSs) is a comprehensive technique for identifying abnormalities in ICSS and is presented as a solution. To detect cyber risks in ICSS, MADICSs seeks to provide a consistent and unified approach for comparing data, which any researcher can utilize to compare data from different sources. Although these processes are based on a standard machine learning/deep learning methodology [13,14], they have been tailored for industrial settings. They can deal with the unique challenges of these types of scenarios. The statistically significant relationship between the characteristics of an ICS and the repetitiveness of its actions is one of the peculiarities that distinguishes it from other scenarios, such as 5G networks [15] or clinical information systems [16]. The primary contributions of the proposed research are as follows:

- The development of an intelligence system based on machine learning and deep learning approaches to detect serious attacks on ICSs.
- The use of a sensitive analysis to find the critical patterns in an entire dataset.
- The primary motivation for the research was to compare the results of the proposed system with existing systems for this dataset. We concluded that the proposed system achieved a high level of accuracy.
- The primary goal is the development of a mechanism for detecting anomalies to protect ICSs from any cybersecurity threat to food security.

The remainder of this paper is arranged as follows: In Section 2, we provide the background for the study; in Section 3, we explain the methods of data collection; in Section 4, we describe the analytical findings of the proposed system for detecting ICS attacks; in Section 5 we discuss the results as well as a comparison with existing ICS systems; in Section 6, we present the conclusions of the proposed research.

2. Background Studies

Previous research has focused on detecting cyberattacks on ICSs, which has resulted in several publications. An IDS based on rules and deterministic finite automata (DFA) are two examples of this system. The majority of current research has focused on developing new approaches that have taken advantage of cutting-edge technologies, such as big data and machine learning/deep learning. A growing number of machine learning (ML) and deep learning (DL) approaches are being used to detect cyberattacks in the industrial sector. The most significant publications on ML and DL anomaly detection in industrial contexts are evaluated here [16].

Many academic ICSs use publicly available datasets to investigate ML algorithms, which is becoming increasingly common. The following are some of the drawbacks of the public database system in use today. The architecture proposed is limited to materials based on the Modbus/TCP protocol suite [17]. There was very little data acquired during online testing activities by [18], and only multi-ML algorithms were developed due to those activities. A similar issue existed in [19], where the database used was outdated and did not represent current threats. In contrast, the dataset in [20] was small (around 1000 occurrences) and was restricted to a single cyberattack. According to another report, the database was out of date and the assaults were linked to the field of IT [21]. The Singapore University of

Technology developed a water treatment testbed that included supervisory control and data acquisition (SCADA) network traffic and assault scenarios for use in water treatment [22]. To train and test ML systems, real-time datasets, including common cyberattacks and 35 different types of cyberattacks, have recently been generated for training and testing ML systems. The only strategies that have been used to compare the performances of different algorithms are supervised ML techniques. The results have shown that algorithms have a considerable probability of false detection, which is supported by the literature. The performance of ML algorithms on a public dataset may result in a decent output depending on the dataset [23]. However, the payload/data frame is controlled using dataset labels, and the assaults are manually randomized and parameterized to imitate the operational/attack situation [24].

Many DL and ML approaches have been reported in the literature, in addition to convolution neural network and long short-term memory (CNN-LSTM) networks. According to [25], unsupervised ML may be used to identify irregularities in cyber-physical systems (CPS). Among 23 methods, they tested deep neural networks (DNNs) and support vector machine (SVM) techniques, which were both designed specifically to work with time-series data. They used the test dataset's mean and standard deviation to scale the dataset to a new size. Autoencoders (AE) and 1D convolutional neural networks (CNN) were proposed in [26] as a DL approach for identifying ICS anomalies. Additionally, the authors recommended filtering features to choose those most suited for anomaly detection from among the DL models they presented. They developed a feature extraction method that used the discrete Fourier transform (DFT) to compute features in the frequency domain. When utilizing DFT to extract features, important information from the remaining signal was lost instead of using only the highest energy bands. In addition, they used a threshold for anomaly detection based on the test dataset's mean and standard deviation. Unsupervised anomaly detection was presented by the authors of [27] that focused on large-scale and real-time data processing. The three pillars of this strategy were update triggers, tree growth, and mass weighting methods. Thanks to this combination, random trees could be generated and updated in real time. They described the use of clustering analysis to reveal the dataset's underlying patterns for another unsupervised anomaly detection. Next, the researchers used cluster intra-distances and inter-distances to extract features from the clusters [28]. Finally, an inference method was used to determine whether an irregularity was sparked. The authors in [29] devised an unsupervised learning technique based on stacked denoising autoencoders. Because the original network stream was used in their solution, it did not require any special skills.

Furthermore, several surveys on IDSs have been conducted on IoT networks and their lightweight devices. Nevertheless, the majority of these surveys did not address the deployment of ML or DL approaches as detection mechanisms in IoT networks and their networks in any depth. In several recent studies, it was observed that the emphasis was on studying IoT security difficulties in general and categorizing them into multiple layers related to applications, network security, encryption, authentication, and access controls [30–35]. There is still more work to be done in ML and deep learning-based techniques for intrusion detection systems in IoT networks, which is the primary emphasis of this study.

3. Materials and Methods

In this section, we present the components of an intelligent system based on ML anda DL approach. The mechanism for detecting cyberattacks on ICSs is displayed in Figure 2. The system can achieve high performance in detecting various types of ICS attacks.

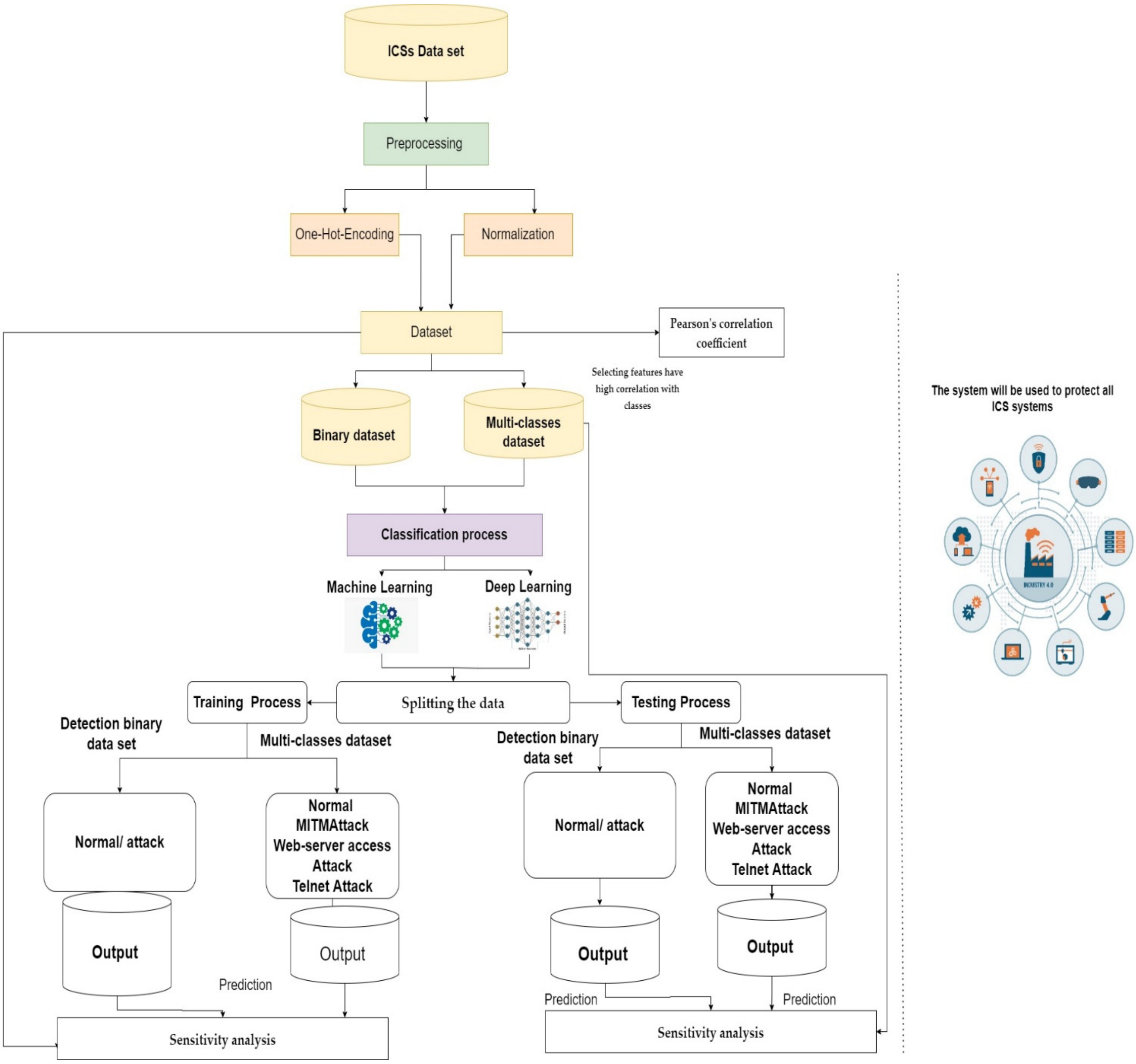

Figure 2. Framework of the ICS security system.

3.1. The Dataset

The standard dataset was collected from network traffic in ICS systems. The standard dataset was operated in collaboration with industrial partners Necon Automation and IIUM. IIUM has developed an ICS cyber system for evaluating and testing the proposed system. The research produced a system for gathering data from the Necon Automation system. The Institute of Information Technology and Management (IIUM) has created an in-house revolutionary portable ICS cyber test kit for the purposes of research and teaching [13]. The package includes a PLC system, an HMI system, modules for process simulation, an Ethernet switch, a physical sensor, and an attacker system. Real industrial network flow data were provided by the ICS portable kit package, which may be used for research and training, as well as the developing of machine learning and deep learning methods. Figure 3 shows the system architecture.

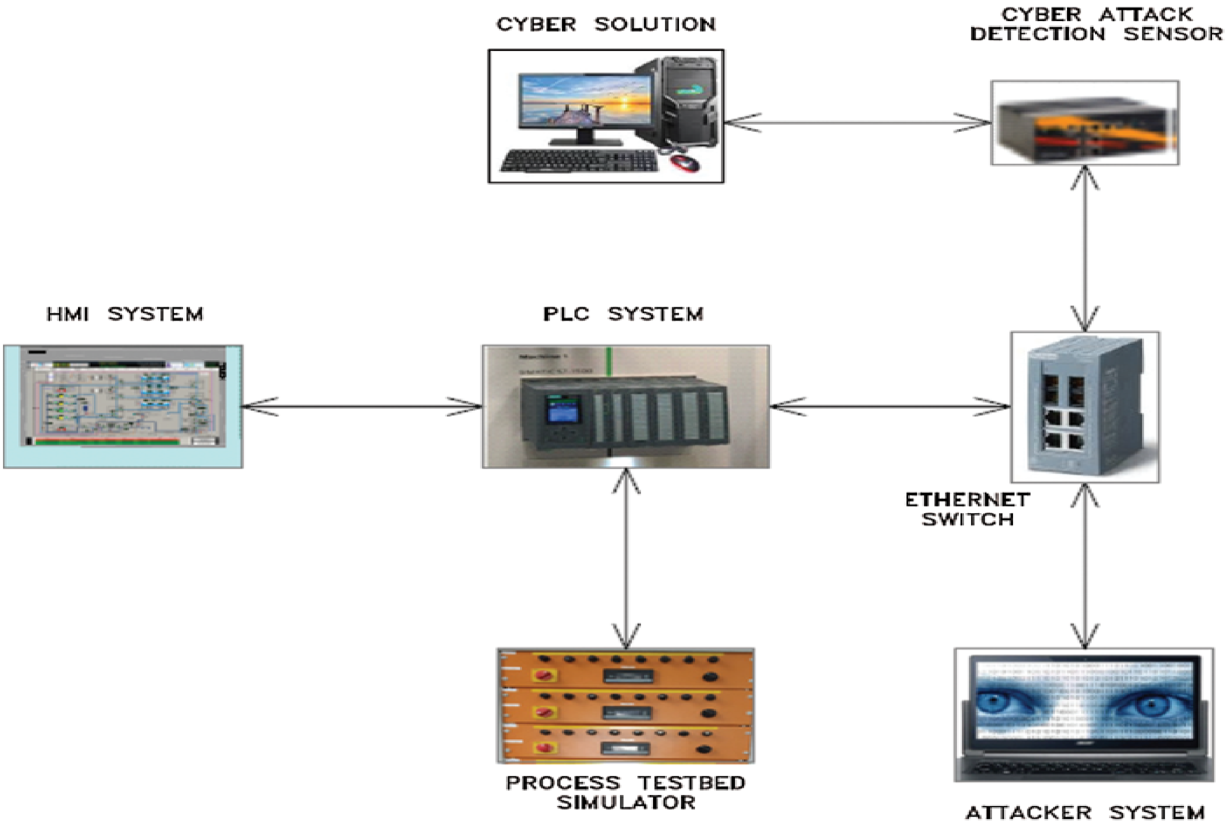

Figure 3. ICSs portable testbed prototype [36].

The dataset contains six features: the timestamp of each packet, source IP, destination IP, OT protocols, and a summary packet of information for DPI. Table 1 shows the MITM attack, Telnet attack, and Web-server access attack based on OT ICS protocols.

Table 1. Description of the dataset attacks.

Attacks	Description	OT Protocol
MITM attack	A man-in-the-middle attack is one type of eavesdropping attack which interrupts an ongoing communication or data transfer.	MITM
Web-server access attack	Web-server attacks have many forms of attack such as DoS, DDoS, and DNS server hijacking used to misconfigure web servers.	Web-server access
Telnet attack	This type of attack allows the hacker to remotely access the router or switch off the network.	Telnet
Normal	S7 is a protocol run on programmable logic controllers.	S7, TCP

Table 2 shows the volumes of each attack on the dataset. Note that the MIMT has height instance values as compared with other classes.

Table 2. Input datasets for each attack label.

#Labels	Volume
MIMT	14,594
Telnet attack	89
Web Access PLC attack	21,435
Normal packets	62,533

3.2. Preprocessing Method

An IDS cannot function well without first preparing the data for analysis. Therefore, data preprocessing is vital. The preprocessing step comprises four units: one-shot encoding, feature selection and data standardization, imbalance handling, and normalization.

3.2.1. One-Hot Encoding Method

Using a single one-hot encoding operation is one of the most commonly utilized approaches for the numeralization of categorical characteristic ICSs. It turns each character type characteristic into a binary vector and assigns a value of 1 to the associated category while assigning a value of 0 to the others. For example, the attribute protocol type and source and destination.

3.2.2. Normalization Method

A possible overlap in the training process caused by handling big datasets was avoided by employing maximum-minimum normalization methods after the categorical variables had been transformed. We utilized a scaling range from 0 to 1 in the normalization procedure to scale the dataset in the same range.

$$z_n = \frac{x - x_{min}}{x_{max - x_{min}}}(New_{max_x} - New_{min_x}) + New_{min_x} \quad (1)$$

where, x_{min} is the minimum of the data, x_{max} is the maximum of the data; New_{min_x} is the minimum number (0); New_{max_x} is the maximum number (1).

3.3. Machine Learning Approaches

The ML algorithms, namely KNN and decision tree (DT), were employed to detect the ICS attacks. A detailed description of this algorithm is presented in the following subsection.

3.3.1. K-Nearest Neighbor (KNN) Algorithm

KNN is an ML algorithm based on the supervised learning technique and is one of the most fundamental ML algorithms. The KNN algorithm compares new instances/data to existing examples and sorts them into the most comparable categories, depending on how similar they are to the previous cases. To classify new data points, the KNN algorithm compares them to previously stored data points and determines their similarity. It is possible to utilize the KNN approach to swiftly categorize new data into one of the relevant categories when it is first introduced. However, the KNN approach is more typically used for classification problems than for regression problems [37,38]. When utilizing the KNN approach, no assumptions about the underlying data are made, resulting in it being classified as a nonparametric approach. A lazy learner algorithm is sometimes termed as such because it does not immediately learn from the training set but instead stores the dataset and performs an action on it until it comes time to classify the data using the algorithm. In this study, we used the Euclidean distance function ($\mathrm{E_i}$) to find the distance between the classes of ICS network data. The mathematical expression of this Euclidean distance function is as follows:

$$\mathrm{E_i} = \sqrt{(\mathrm{c_1} - \mathrm{c_2}) + (\mathrm{d_1} - \mathrm{d_2})^2} \quad (2)$$

where c_1, c_2, d_1, and d_2 are the input data variables.

3.3.2. Decision Tree Algorithm

Classification and regression issues are frequently addressed using the ML technique's DT. A root node is at the top of a DT model and branches based on the data's core characteristic ICSs are at the bottom. The output of a feature is represented by a branch, while a child node represents the output of a category. Relying on sample training, one may learn the classification model using a classification DT, an example of supervised learning. Ultimately, the classification work is completed by the incoming data, which are evaluated by each node. ID3, C4.5, and CART are three forms of decision trees that may be categorized based on the parameters used to determine branch properties. ID3 implements a greedy algorithm and uses information entropy as a branch criterion [39]:

$$Entropy = (S) = \sum_{i=1}^{C} p_i \, log_2 \, p_i \tag{3}$$

$$entropy\ (S\,|B) = \sum_{j=1}^{j} \frac{|s_i|}{|S_i|}\, entropy\ (S_i) \tag{4}$$

$$\text{Gain}\ (S\,|B) = entropy(S) - entropy(S\,|B) \tag{5}$$

where S is the training dataset, C is the class of dataset which is attacks and normal, P_i is the probability of the sample that indicates class C, S_i is the samples of subsets of the class in features B.

3.3.3. Logistic Regression Algorithm

When categorizing dependent categorical data, binary classification is commonly used [40]. There are several applications for this kind of guided learning. Based on the values of the dependent variables, the algorithms forecast the outcome. Logistic regression uses an S-shaped logistic curve to separate data points for the separation process. To predict classification probabilities, logistic regression is used to construct a decision border, which is known as drawing the logistic curve. Some people refer to the logistic function as a sigmoid function:

$$S(x) = \frac{1}{1 + e - x} \tag{6}$$

An integer is sent through the sigmoid function, which returns a 0–1 as the outcome of the operation. The sigmoid function returns the likelihood of categorization in each case. When $S(x)$ is less than 0.5, the data is classified as class A, and when $S(x)$ is more than 0.5, the data is classified as class B.

3.3.4. Linear Discriminant Analysis

When dealing with high-dimensional applications, the linear ML method known as linear discriminant analysis (LDA) comes in handy. It is used to model and convert data from a high-space dimension to a low-space dimension by categorizing the data into regular and harmful packets and transforming the data between the two groups [41].

3.4. Deep Learning Approach

A DNN is a well-known DL approach among scientists. The DNN topology consists of three layers: the input, the hidden, and the output layers, all of which are connected. There are no connections between any neurons in the layers above or below a layer, however, every neuron in a layer, above or below the layer, is connected to every other neuron. The efficiency of the network learning effect is increased by adding an activation function to the output of each layer of the network. Consequently, a DNN may be viewed as an enormous perceptron consisting of many perceptrons working together [42–45].

A CNN is an artificial neural network commonly used in a DL approach to identify and classify images and objects. The CNN structure comprises three layers: convolutional, pooling, and fully linked layers. In this structure, feature extraction and dimensionality reduction are accomplished using convolutional and pooling layers, respectively. Each layer is attached to the preceding layer after being folded and connected. The fundamental structure of the CNN model utilized to detect fraudulent Android apps is shown in Figure 4.

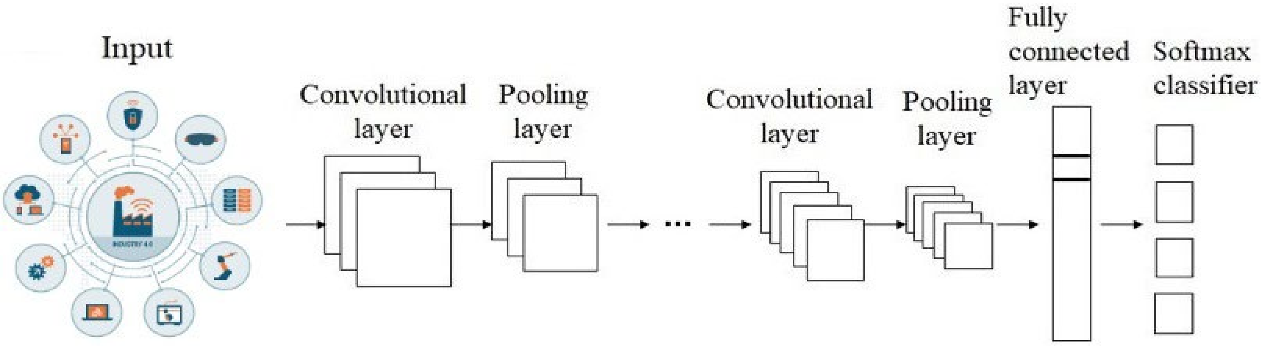

Figure 4. Structure of the CNN model based on ICS attack detection.

Long short-term memory (LSTM) is a recurrent neural network, although it performs far better in terms of memory than ordinary recurrent neural networks. LSTM performs significantly better when the learner has a firm comprehension of the patterns to be learned. LSTM and other neural networks differ because LSTM may have several hidden layers. As it progresses through each layer, valuable information is kept, and all irrelevant information is eliminated in each cell. An LSTM block building is shown in Figure 5. To control how much value is sent through the cell, each of the cell's input, forget, f_t, and output gates can be employed independently. The four LSTM block gates are as follows: cell state gate C_t, which remembers information over time; forget gate f_t; input gate i_t; and output gate o_t. The cell state gate C_t is responsible for remembering information over time. The activation functions for each gate are integrated into the gate layer. In addition to the three inputs, the LSTM block has three outputs: the cell state C_t, the previously concealed cell state h_t, and the current input X_t. The LSTM block contains three inputs and three outputs. It is possible to create the current output after the disguised state has been discovered. The following is a mathematical formulation for the LSTM unit, which is defined as follows:

$$f_t = \sigma\left(W_f \,.\, X_t + W_f . \, h_{t-1} + b_f\right) \tag{7}$$

$$i_t = \sigma(W_i . \, X_t + W_i \,.\, h_{t-1} + b_i) \tag{8}$$

$$S_t = \tan\mathrm{h}(W_c . \, X_t + W_c \,.\, h_{t-1} + b_c) \tag{9}$$

$$C_t = (i_t * S_t + f_t * S_{t-1}) \tag{10}$$

$$o_t = \sigma(W_o + X_t + W_o \,.\, h_{t-1} + \ V_o . C_t + b_o) \tag{11}$$

$$h_t = o_t + \tan\mathrm{h}(C_t) \tag{12}$$

The proposed CNN-LSTM model comprises two LSTM layers and four fully connected (FC) layers. It also has input and soft-max output layers, among other features. In addition, there are four convolutional layers and one pooling layer. The network architecture of the CNN-LSTM model in the proposed system is shown in Figure 6.

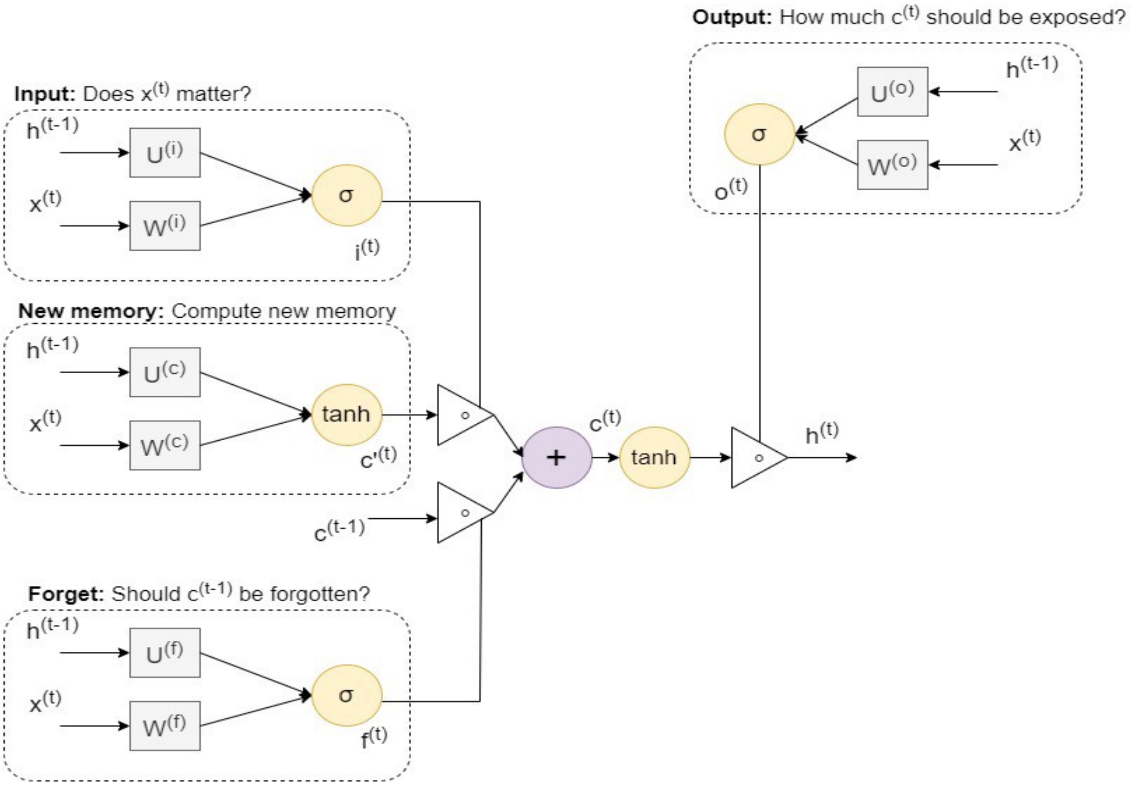

Figure 5. Structure of the LSTM technique.

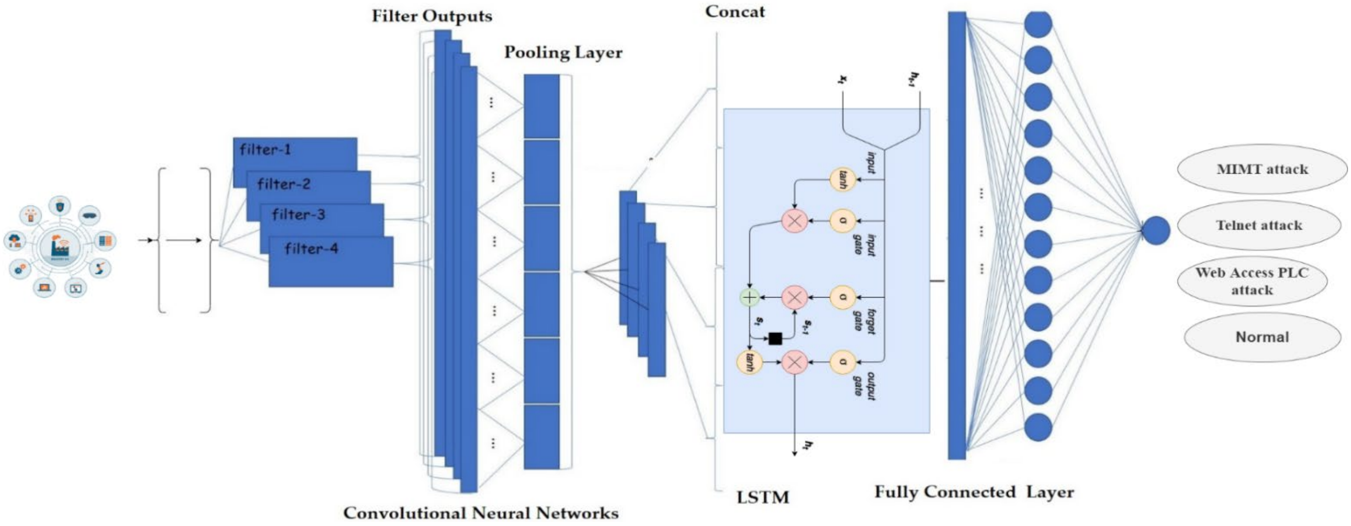

Figure 6. The srchitectre of the CNN-LSTM model for detecting ICS attacks.

The suggested model directly incorporates the 1D ICS network data into the computation for the first time. The supplied data are in the following format, consisting of four features: (1) a convolutional layer extracts abstract characteristic from the raw ICS data using 512 1D convolutional kernels, each having a 5-by-1 shape, one stride in the conv layer1, and one stride in the conv layer2 (the first convolutional layer); (2) an activation layer with rectified linear units (ReLUs) follows the convolutional layer; this layer may introduce nonlinearity into the proposed model. The following is a mathematical formulation for the 1D convolutional operation, and the ReLU activation in the case of the ReLU is shown below:

$$y_j^l = \sigma\left(\sum_{i=1}^{N_{l-1}} conv\left(w_{i,j}^l,\ x_i^{l-1}\right)\right) + b_j^l \tag{13}$$

where N_{l-1} is the number of the feature map; y_j^t is the feature map of ICS data; $w_{i,j}^t$ is the convolutional kernel; b_j^t is the bias of feature map.

The $\sigma()$ denotes the ReLU activation function, which we used to avoid the overfitting issue in the obtained data:

$$\sigma(x) = \begin{cases} 0, & x \leq 0 \\ x, & 0 > 0 \end{cases} \tag{14}$$

The convolution kernel was used to pass the training data into the max pool to extract significant features to improve the classification actuary. A function expression of the max pool is defined below:

$$Q_j = Max\left(P_j^0, P_j^1, P_j^2 P_j^3 \ldots . P_j^t\right), \tag{15}$$

where Q_j denotes the output from the max pool, and P_j^t is the feature map before max. The significant parameter indicator values of the proposed CNN-LSTM model are presented in Table 3.

Table 3. Parameter values of the proposed CNN-LSTM model.

#Parameters Indicators	#Values
Convolution kernel size	5
The size of max pooling	5
Drop out	0.50
The size of the FC layer	128
Activation function	ReLU
Optimizer	Adam
Epochs	20
Batch size	120

4. Experimental Analysis

In this section, we apply ML and DL algorithms to detect ICS cyberattacks. The effectiveness of each algorithm was tested using a well-known ICS IDS dataset. The questions for the proposed research are as follows:

How can ML and DL algorithms detect anomalies in an ICSs environment?

What are the appropriate algorithms for detecting ICSs attacks?

How can the developed, robustness, and efficiency models protect the ICSs system?

4.1. Splitting the Data

A validation method is essential for evaluating a system. In this study, we divided the data into training and testing the proposed system. Table 4 shows the split ICS dataset from artificial intelligence algorithms detecting intrusion.

Table 4. Volume of datasets.

Datasets	Total Volume	Training	Testing
ICSs	98,651	69,055	29,596

4.2. Experimental Environments

The proposed system was developed in a specific hardware and software environment because we knew the network data were very complex. The platform used to detect intrusion in Android applications is presented in Table 5.

Table 5. Environmental requirements of the proposed model.

Hardware	Software	Version
RAM size 8 GB	Python	3.6
Intel(R) Core(TM) i7	Panda	1.4.2
CPU 1.80 GHz	TensorFlow library	2.8.0
	Keras library	2.8.0
	Matplotlib	3.1
	NumPy library	1.11.0

4.3. Performance Measurements

In order to evaluate the high performance of the ICS security system, evaluation metrics were proposed such as mean square error (MSE), the Pearson's correlation coefficient (R), the root-mean-square error (RMSE), accuracy, precision, recall and F1 score,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_{i,exp} - y_{i,\ pred}\right)^2 \tag{16}$$

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{\left(y_{i,exp} - y_{i,pred}\right)^2}{n}} \tag{17}$$

$$R^2\, bn1 - \frac{\sum_{i=1}^{n}\ (y_{i,\ exp} - y_{i,\ pred})^2}{\sum_{i=1}^{n}\ (y_{i,\ exp} - y_{avg,\ exp})^2} \tag{18}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{19}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{20}$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{21}$$

$$\text{Fscore} = \frac{2 * preision{*}\text{Sensitivity}}{preision + \text{Sensitivity}} \times 100\% \tag{22}$$

$$R\% = \frac{n\left(\sum_{i=1}^{n} y_{i,exp} \times y_{i,\ pred}\right) - \left(\sum_{i=1}^{n} y_{i,exp}\right)\left(\sum_{i=1}^{n} y_{i,\ pred}\right)}{\sqrt{\left[n\left(\sum_{i=1}^{n} y_{i,exp}\right)^2 - \left(\sum_{i=1}^{n} y_{i,exp}\right)^2\right]\left[n\left(\sum_{i=1}^{n} y_{i,pred}\right)^2 - \left(\sum_{i=1}^{n} y_{i,pred}\right)^2\right]}} \times 100 \tag{23}$$

where, the confusion metrics of ICS system such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are used as parameters for examining the model, where $y_{i,exp}$ is ICSs input data and $y_{i,pred}$ is output of the developing ICS system.

4.4. Results

In this section, we present the results of the ML and DL approaches. The proposed system was tested in two stages: binary classification (normal and attacks) and multiclass classification in four classes (MITM attack, Telnet attack, Web-server access attack, and normal). The system was tested using real ICS network datasets, including different types of attacks.

4.4.1. Binary Classification Results

The proposed ML and DL algorithms were applied to test their effectiveness in binary classification. We classified the datasets as normal or attacks. ML algorithms, such as KNN, DT, and logistic function, were considered for classifying ICS attacks. Table 6 shows the results of these algorithms for detecting ICS attacks. Based on the empirical results, most of the algorithms achieved superior accuracy, but the KNN and DT algorithms both achieved an accuracy of 100%.

Table 6. Results of approaches in binary classification for detecting ICS attacks.

Algorithms	Classes	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Logistic regression	Normal	99	98	100	99
	Attacks		100	99	99
KNN	Normal	100	100	100	100
	Attacks		100	100	100
Linear discriminant	Normal	99	98	99	99
	Attacks		100	99	99
Decision tree	Normal	100	100	100	100
	Attacks		100	100	100

The confusion matrix of the ML logistic regression, KNN, linear discriminant, and DT approaches are shown in Figure 7. The matrix reports the results of this algorithm using different metrics, such as false negatives, true positives, and true negatives. The binary classification is either normal or attacks (0, 1). The logistic regression results were 62.94%, classified as TP, where the TN was 36.18%, and the false-positive was 0.85%. The KNN and DT approaches showed the correct classification of 36.18 classified as TN and 63.80 classified as TP. The linear discriminant was 0.64% FP, 36% normal, and 63.15% attacks.

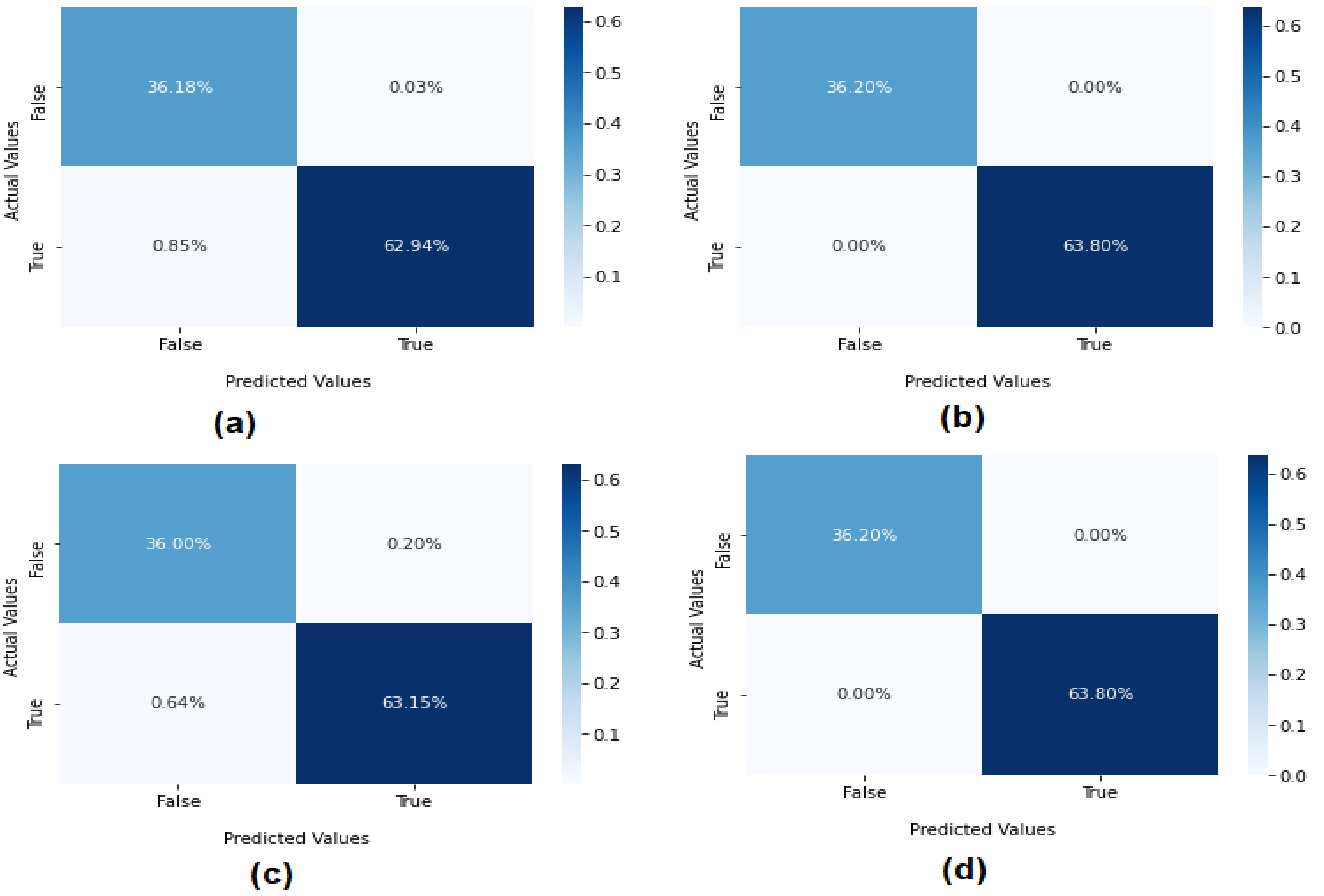

Figure 7. Confusion metrics of ML models: (**a**) logistic regression, (**b**) KNN, (**c**) LDA, and (**d**) DT.

Table 7 shows the results of the DL CNN-LSTM model for detecting and classifying ICS attacks using the binary dataset. The DL approach achieved an accuracy of 98.89%.

Table 7. Results of the DL CNN-LSTM model for detecting ICS attacks using binary classification.

Algorithms	Classes	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
CNN-LSTM	Normal	98.89	99.83	98.42	99.12
	Attacks		100	99	99

The performance of the CNN-LSTM model for predicting ICS attacks using binary classification is shown in Figure 8. The accuracy of the CNN-LSTM model in the training process was 99%, whereas the performance of the CNN-LSTM model for detecting ICS attacks as the validation step was 98.89%. The training model varied in each epoch. The validation loss of the CNN-LSTM model was 750–250.

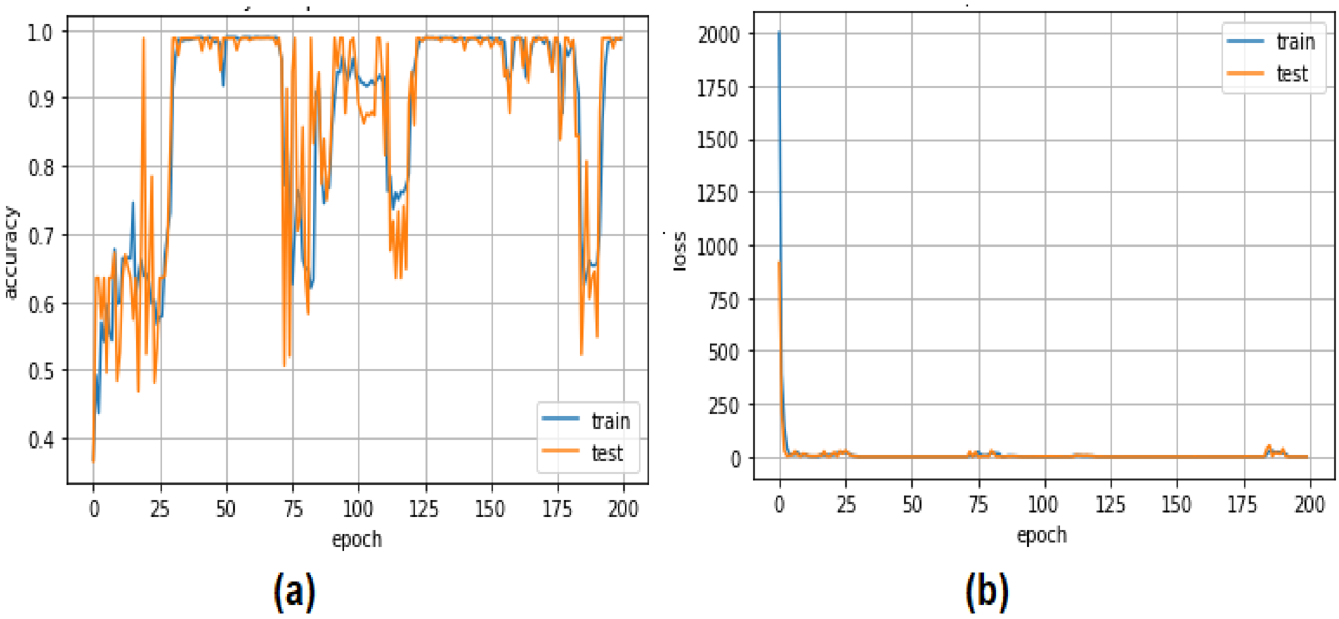

Figure 8. Performance and loss accuracy of the CNN-LSTM model using the binary dataset. (**a**) model accuracy (**b**) model loss.

Figure 9 shows the confusion metrics of the CNN-LSTM model using binary classification. The CNN-LSTM model showed promising results; the percentage of the true negative was 36.11%, and the false-positive was 62.79%. The misclassification percentage of the CNN-LSTM model was 1.01%.

4.4.2. Results of Multiclass Classification

In this experiment, ML and DL approaches were examined using the multiclass classification dataset: MITM attack, web-server access attack, Telnet attack, and normal. Table 8 shows the results of the logistic regression algorithm to detect ICS attacks. The results of the logistic regression method were not satisfactory. The accuracy value in all classes was 30%. We recommend that this algorithm be used for detecting multiclass attacks on ICSs.

Table 8. Results of logistic regression using the multiclass classification dataset.

Attacks	Precision (%)	Recall (%)	F1 Score (%)
Normal	67	46	54
MITM attack	0.0	0.0	0.00
Web-server access attack	82	0.01	0.02
Telnet attack	0.00	0.00	0.00
	Accuracy 30%		
Weighted average	60	30	35

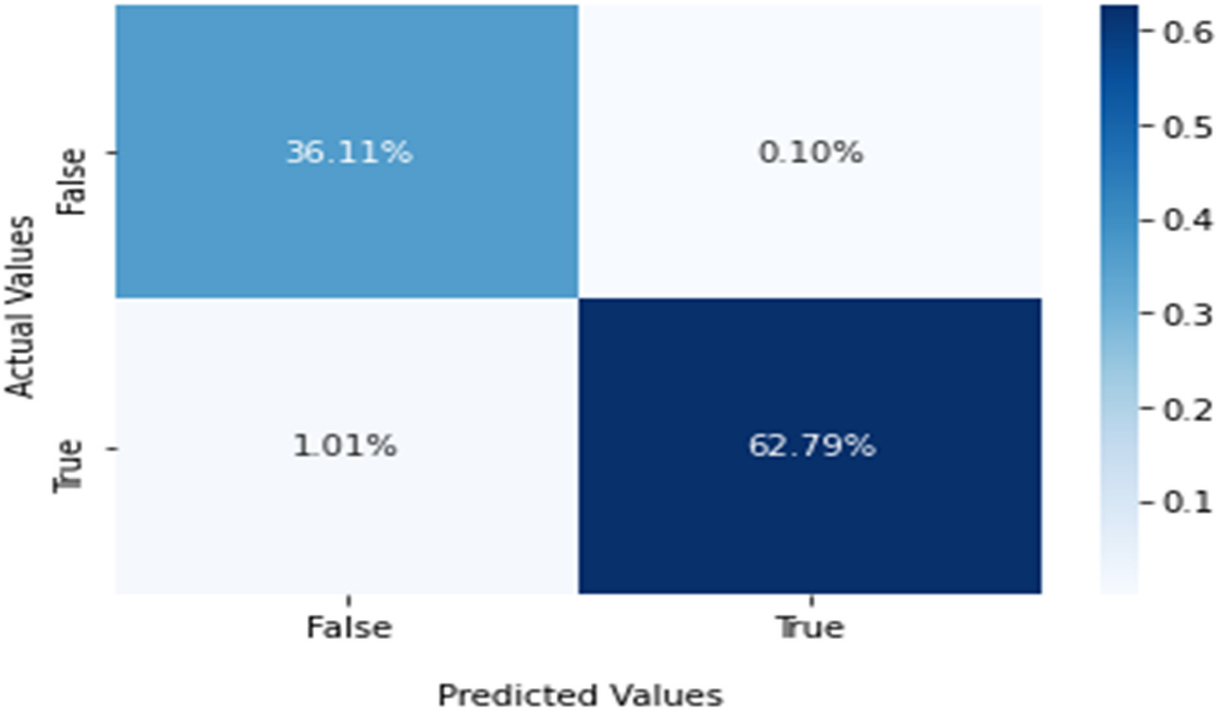

Figure 9. Confusion metrics of the CNN-LSTM model using the binary dataset.

The detection results of the KNN algorithm for the multiclass classification on the ICS platform are presented in Table 9. Based on the confusion metric ICSs, the results were superior, with an accuracy of 100% for the KNN.

Table 9. Results of the KNN approach using the multiclass classification dataset.

Attacks	Precision (%)	Recall (%)	F1 Score (%)
Normal	100	100	100
MITM attack	100	100	100
Web-server access attack	100	100	100
Telnet attack	100	100	100
	Accuracy 100%		
Weighted average	100	100	100

The results of the LDA model for detecting ICS attacks using multiclass classification datasets are shown in Table 10. The accuracy of the LDA model was 94.37%. The weighted averages of the precision, recall, and F1 scores were 95%, 94%, and 94%, respectively.

Table 10. Results of the LDA model using the multiclass classification dataset.

Attacks	Precision (%)	Recall (%)	F1 Score (%)
Normal	98	93	95
MITM attack	79	100	88
Web-server access attack	99	94	96
Telnet attack	0.00	0.00	0.00
Accuracy 94.37			
Weighted average	95	94	94

Table 11 shows the results of the DT algorithm, which achieved superior performance. The accuracy percentage of the DT in all classes was 100%. The weighted average performance of the DT was 100% for all classes.

Table 11. Results of the decision tree using the multiclass classification dataset.

Attacks	Precision (%)	Recall (%)	F1 Score (%)
Normal	100	100	100
MITM attack	100	100	100
Web-server access attack	100	100	100
Telnet attack	100	100	100
Accuracy 100%			
Weighted average	100	100	100

Figure 10 shows the confusion metric indicators such as actual negative false-positive rate, valid positive rate, and false negative. The logistic regression model results showed 26.36% correctly classified as normal, whereas the model scored 0.26% and was correctly classified as a Telnet attack. The false-positive rate was very high in the MITM attack at 14.74%, with 21.04% for the Web-server attack. The results of the KNN approach showed 63.84% detected as normal; the true positives were 14.74%, 21.31%, and 0.09% correctly classified as MITM attack, Web-server access attack, and Telnet attack, respectively. The misclassification (FP) was 0.00, demonstrating that the KNN algorithm was appropriate for detecting ICS attacks. The confusion metric results for the LDA model were 59.59% correctly classified as normal packets. In contrast, the true positive metrics showed 14.75% classified as MITM attack, 20.03% classified as Web-server attack, and 0.00 classified as Telnet attack. The false positives of the LDA model were slight at 0.01% uncorrected classified MITM attack, 1.28% uncorrected classified as Web-server attack, and 0.091% uncorrected classified as Telnet attack. The decision tree algorithm showed the same performance as the KNN algorithm, with 63.84% detected as normal. The true positives were 14.74%, 21.31%, and 0.09%, correctly classified as MITM attack, Web-server access attack, and Telnet attack, respectively.

A hybrid DL CNN-LSTM model is proposed to detect ICS cyberattacks. The performance of the DL CNN-LSTM classifier according to evaluation metrics such as accuracy, precision, recall, F-score, and classification performance was evaluated using the four classes, i.e., MITM attack, Web-server access attack, Telnet attack, and normal. The results of the CNN-LSTM model were compared with ML approaches. Table 12 shows the performance of the CNN-LSTM model using the multiclass classification dataset. The CNN-LSTM model achieved the highest accuracy, i.e., 98%. The weighted average performance of the CNN-LSTM model for all four classes was 98%.

Table 12. Results of the CNN-LSTM model using the multiclass classification dataset.

Attacks	Precision (%)	Recall (%)	F1 Score (%)
Normal	99	98	99
MITM attack	100	97	98
Web-server access attack	88	50	64
Telnet attack	92	99	96
Accuracy 98%			
Weighted average	98	98	98
Loss 0.076			

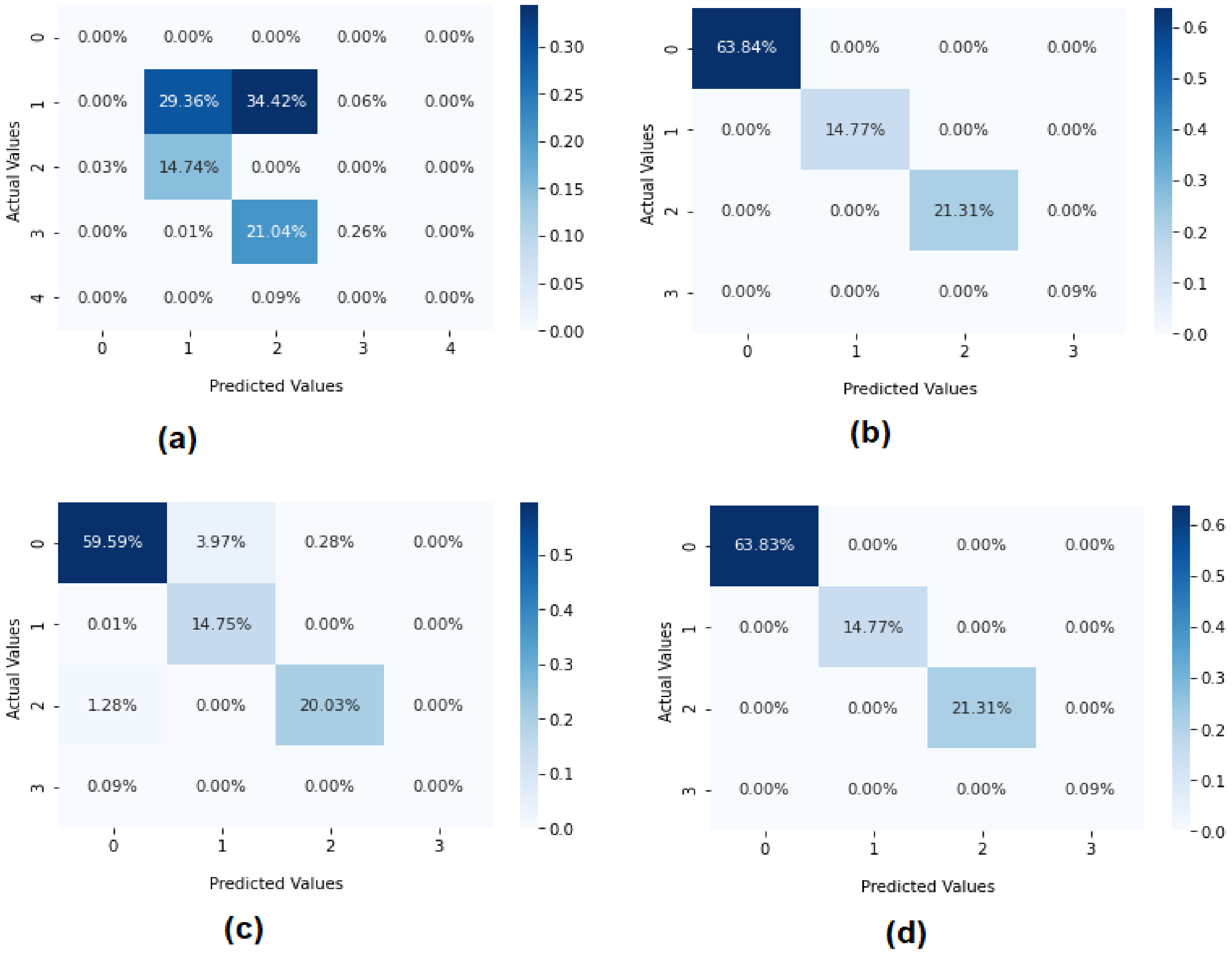

Figure 10. Confusion metrics of ML models using a multiclass classification dataset: (**a**) logistic regression; (**b**) KNN; (**c**) LDA; (**d**) DT.

Figure 11 shows the confusion metrics of the CNN-LSTM model for detecting anomalies in the ICS system. The graphical representation of the CNN-LSTM model shows that 14.06% (TN) was correctly classified as normal; 62% was correctly classified as MITM attacks, 0.04% was correctly classified as Telnet attacks, and 21.39% was correctly classified as Web-server access attacks. The false-positive rate was high at 0.05% on the Telnet attack. The false-negative rate was high at 0.13% with the normal.

The accuracy performance and loss of the CNN-LSTM model for detecting multiple classes are shown in Figure 12. The plot shows that the accuracy of training and volition increased from 88% to 98%, whereas the accuracy loss decreased from 0.30 to 0.10 in the testing phase with 20 epochs.

4.5. Sensitivity Analysis

A sensitivity analysis is an approach for measuring the influence of uncertainties of input data variables. Analyzing the input data is very useful for extracting patterns from a dataset. A sensitivity analysis determines the effects of fluctuations in network features with different attacks on the outputs or performance of a mathematical model or system. In other words, a sensitivity analysis may be used to assign changes in system outputs to distinct sources of uncertainty in system inputs, as opposed to a traditional approach. In this study, we determined that there was a strong association between input attributes and class membership using Pearson's correlation coefficient. Certain traits had strong connections (normal and MITM attacks, Web-server access attacks, and Telnet attacks).

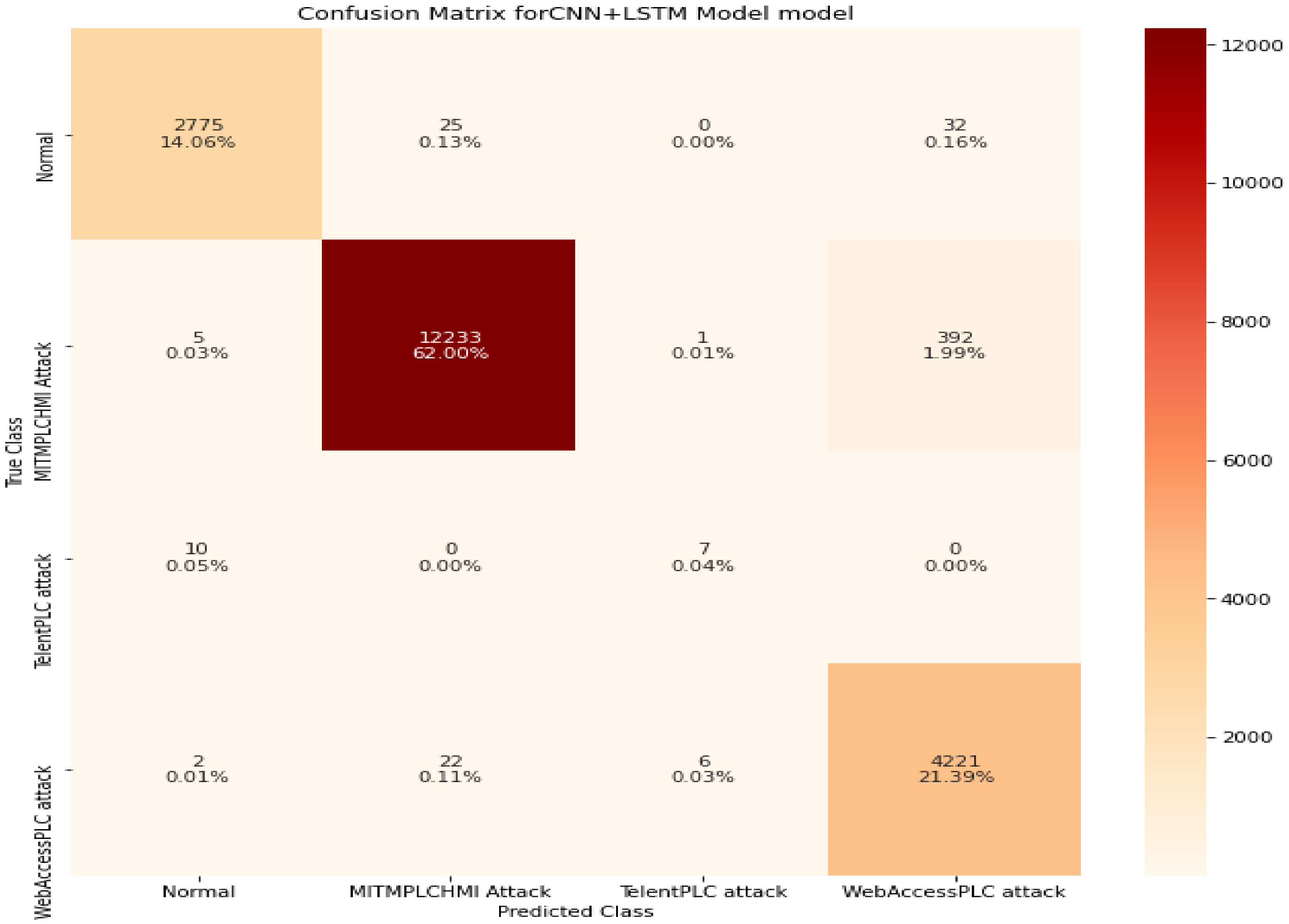

Figure 11. Confusion metrics of the CNN-LSTM model using the multiclass classification dataset.

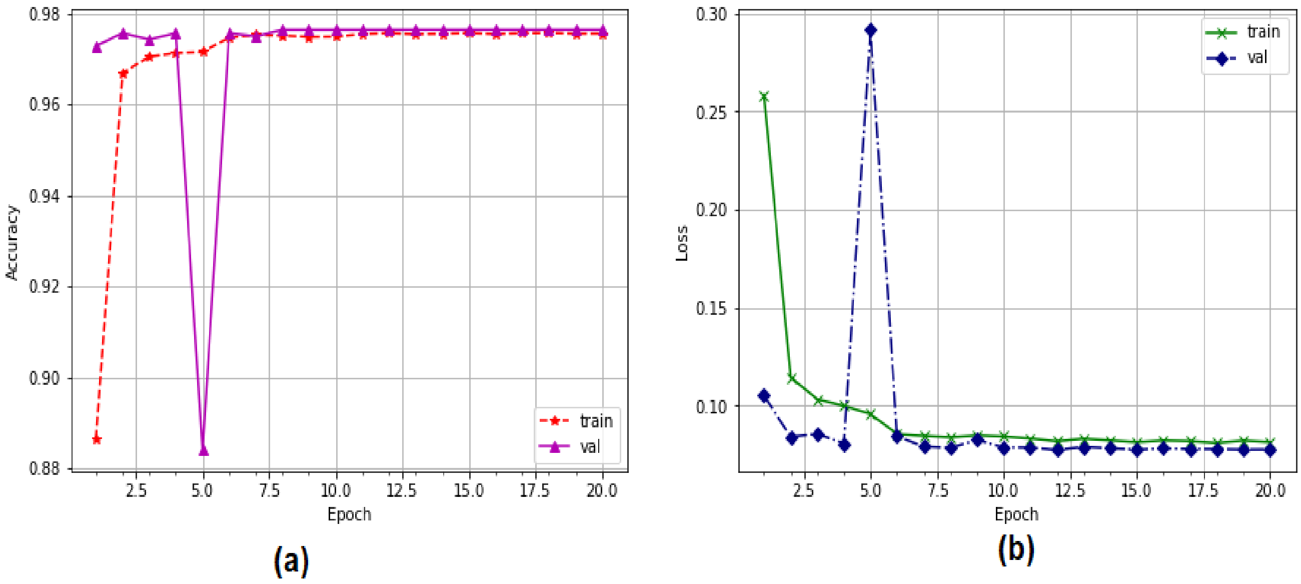

Figure 12. Performance of the CNN-LSTM model for detecting ICS attacks using the multiclass classification dataset: (**a**) Accuracy; (**b**) loss.

We selected features with high relationships with classes. Figure 13 shows the results of the Pearson's correlation coefficient method to determine the significant elements. The destination and source addresses are critical features, with superior corrections among classes of 92%, whereas the length features do not have good connections among the classes.

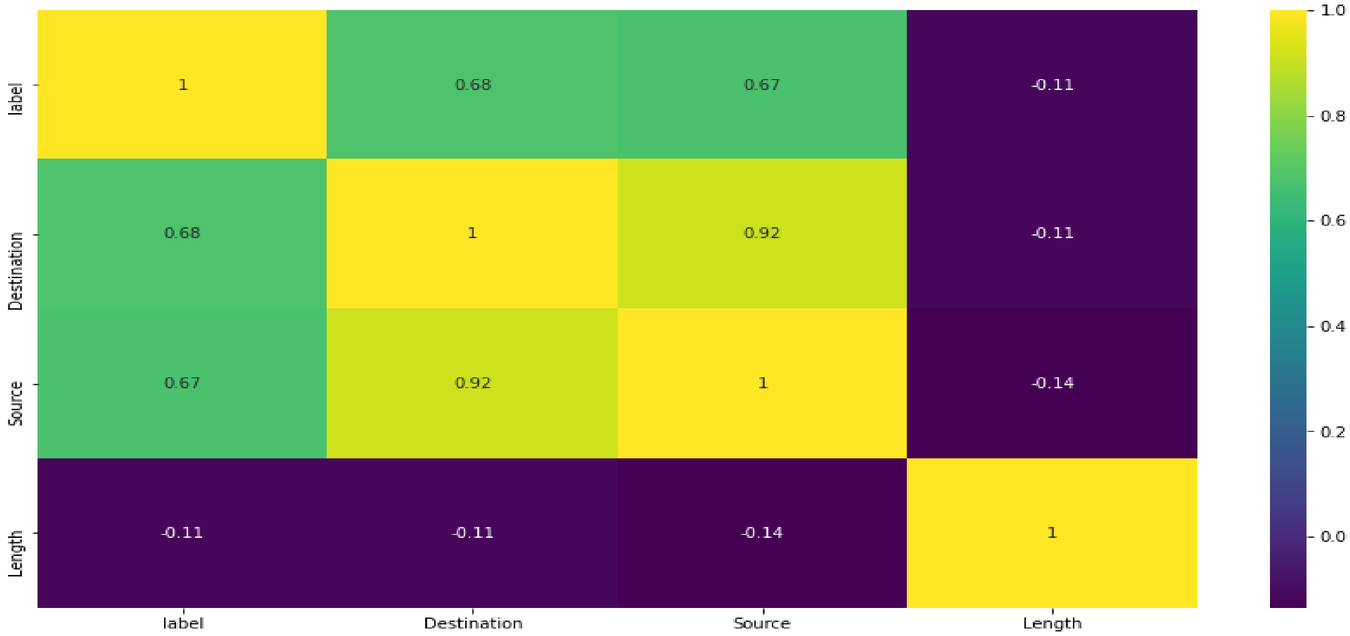

Figure 13. The correlation coefficients of the ICS datasets.

Moreover, the mean absolute error statistical analysis (MAE), MSE, RMSE, and R^2 were applied to measure the percentage error between the target and prediction values. Table 13 summarizes the statistical analysis of ML and DL with binary classification. The KNN and DT algorithms had the highest correlations between the target and prediction values, i.e., R^2 = 100% and prediction output of 0.00.

Table 13. Sensitivity analysis of ML using binary classification.

Models	MAE	MSE	RMSE	R^2 (%)
LG	0.0088	0.0088	0.093	96.21
KNN	0.0	0.0	0.0	100
LDA	0.0084	0.0084	0.0919	99.53
Decision tree	0.0	0.0	0.0	100
CNN-LSTM	0.0111	0.0106	0.1030	95.43

Table 14 compares the prediction results among the target values, ML and DL, using multiclass classification. The KNN and DT algorithms scored the highest (R^2 = 100%), with the prediction error between the target values and prediction output being 0.00. The prediction error of the logistic regression method was very high. Overall, the KNN and DT algorithms were effective in predicting ICS attacks.

Table 14. Sensitivity analysis of ML and DL using multiclass classification.

Models	MAE	MSE	RMSE	R^2 (%)
LG	0.705	0.7096	0.8424	72
KNN	0.0	0.0	0.0	100
LDA	0.073	0.110	0.331	83.73
Decision tree	0.0	0.0	0.0	100
CNN-LSTM	0.043	0.089	0.299	90.53

5. Results and Discussion

ICSs are part of contemporary critical infrastructures, such as water treatment facilities, oil refineries, power grids, and nuclear and thermal power plants. An ICS is a system developed by merging computational and communication components to govern a physical process. An ICS includes devices and subsystems, such as sensors, actuators, programmable logic controllers (PLC), human-machine interfaces (HMI), and SCADA systems.

A steady rise in the frequency of successful attacks on ICSs has led to an urgent need to develop security systems that can accurately and quickly identify irregularities. A new breed of anomaly detectors based on data-centric methods is gaining traction in this effort. Such techniques may automatically understand the dynamic ICSs of the process and the control strategies used in an ICS using ML and DL algorithms. Anomaly detectors can be created more quickly and easily using these techniques than using design-centric approaches based on plant physics and design.

ML, including logistic regression, KNN, linear discriminant, and DT algorithms, and the CNN-LSTM model, are designed to detect ICS attacks and protect the infrastructure of ICSs. The proposed system was examined using the Sung real dataset from industrial partner Necon Automation. This study conducted two binary classification and multiclass classification experiments to detect and classify ICS attacks. Overall, the KNN and DT algorithms achieved the highest performance with binary classification and multiclass classification of 100% concerning the accuracy metric. Figure 14 shows a regression plot of the KNN and DT algorithms, including the correlation between input and prediction values. These algorithms scored 100% with respect to R.

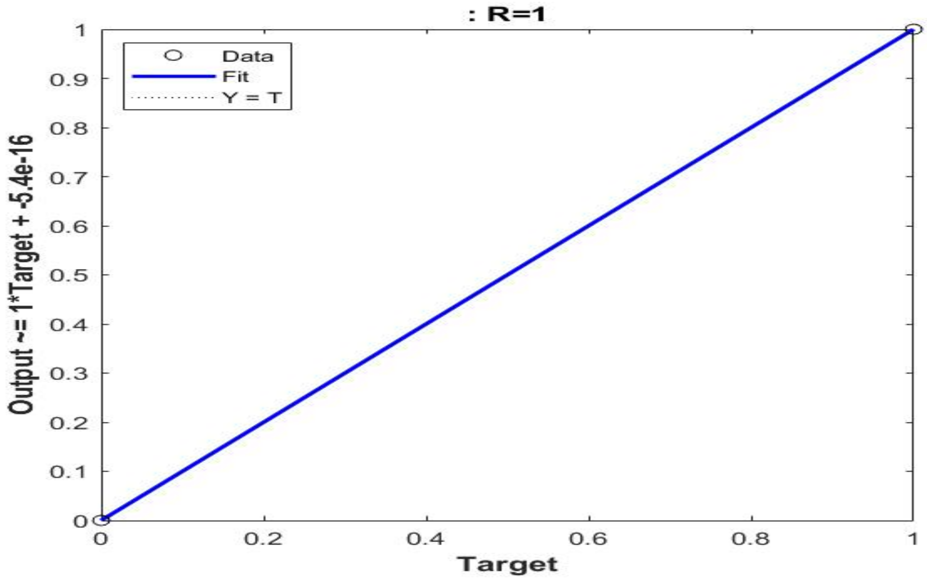

Figure 14. Regression plots for the KNN and DT methods.

The proposed system was compared with that of Sinil Mubarak et al. [36], who developed the dataset. The authors used logistic regression, KNN, naive Bayes, random forest, and ANN algorithms. However, cross-validation was used to evaluate the algorithm. The results of these algorithms were: logistical regression 95.18%, KNN 95.24%, naive Bayes 94.75%, random forest 95.52%, and ANN = 95.14, according to the accuracy metric. In this study, the KNN and DT algorithms achieved accuracies of 100%; the hybrid DL CNN-LSTM model achieved 98% accuracy. Table 15 shows the empirical results of the proposed ML and DL model against existing security systems developing the dataset. Figure 15 shows a graphical depiction of the results obtained by our system and those obtained by other current methods based on accuracy metrics. In general, the approach we recommend has the highest level of accuracy available.

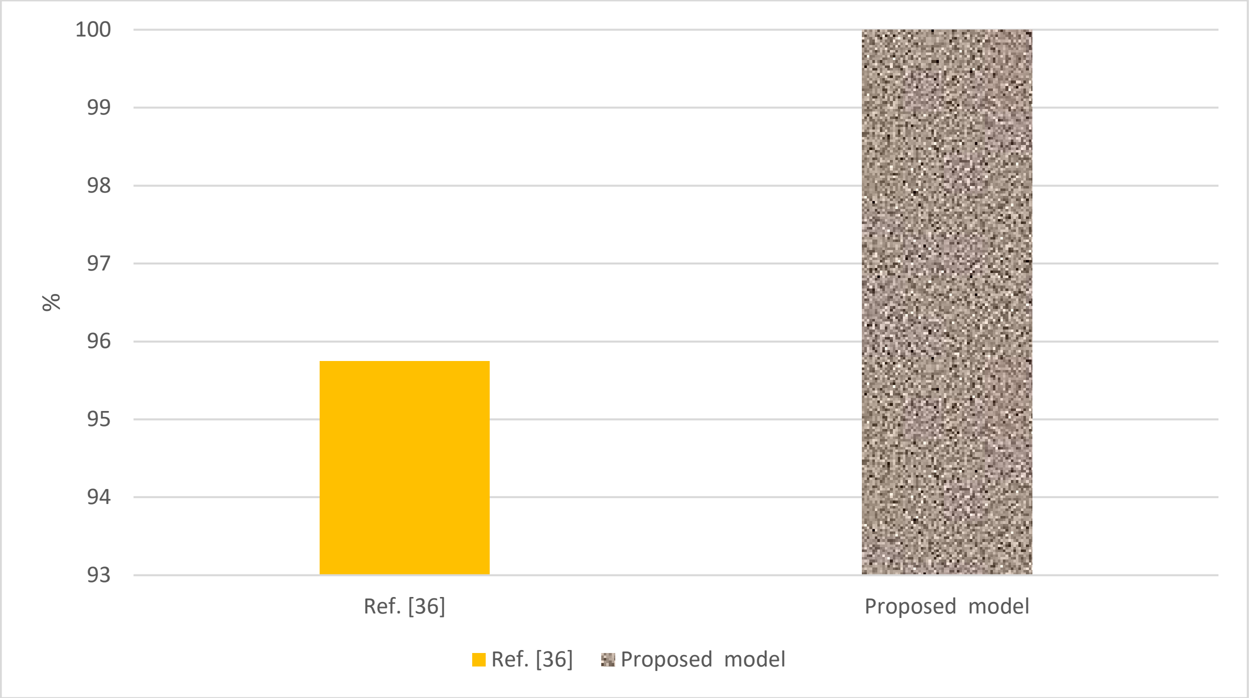

Figure 15. Comparison of the proposed system with the system that developed the dataset.

Table 15. Comparison of the proposed system with the system that developed the dataset.

Reference	Year	Datasets	Model	Accuracy (%)
Ref. [36]	2022	Industrial partner (Necon Automation)	Logistic regression KNN Naive Bayes Random forest ANN	95.18 95.24 94.75 95.52 95.14
Proposed model	2022	Industrial partner (Necon Automation)	KNN Decision tree CNN-LSTM	**100** **100** **98**

A comparison between the proposed system and the existing system using a different dataset is presented in Table 16. Overall, the proposed system achieved superior accuracy as compared with the existing system for protecting the ICS's environment. Figure 16 displays the performance of the proposed system as compared with the existing systems with respect to accuracy metrics.

Table 16. Comparison of the proposed system to existing systems using different ICS datasets.

Reference	Year	Datasets	Model	Precision (%)
Ref. [46]	2018	ICSs	CNN	96
Ref. [47]	2018		MPL	96
Ref. [48]	2019		LSTM	93
Ref. [49]	2017		DNN	98
Ref. [50]	2019		GAN	70
Proposed model	2022		KNN	**100**
			Decision tree	**100**
			CNN-LSTM	**99**

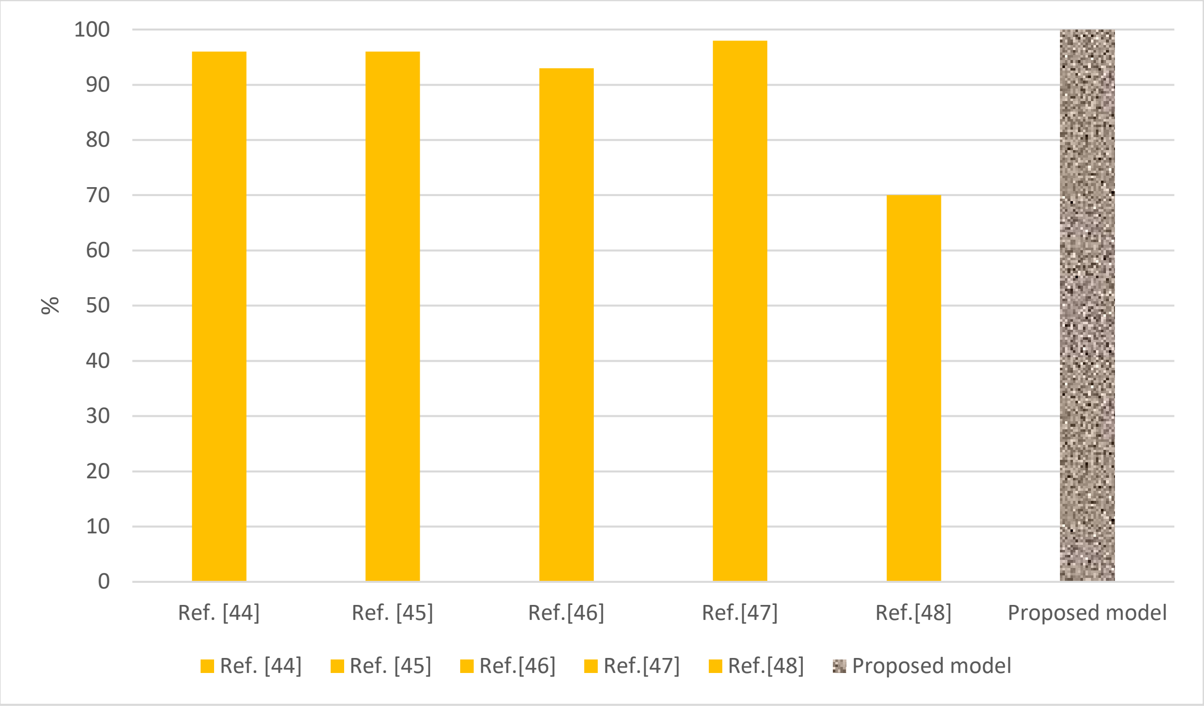

Figure 16. Comparative results of the proposed system with systems using different ICS datasets.

6. Conclusions

Increasing effective assaults on industrial control systems (ICSs) have prompted the creation of protection measures for accurate and quick process anomaly detection. In contrast to other types of cyberattacks, ICS-oriented intrusions have the potential to disrupt critical infrastructure operations, cause significant economic losses, pollute the environment, and even cost human lives. In this study, a security system was developed to protect and prevent cyberattacks from threatening ICSs. The artificial intelligence algorithms namely regression, KNN, LDA, DT, and the CNN-LSTM model are designed to detect malicious ICS attacks. The following conclusions stem from the encouraging findings of this study:

- The machine learning and DL algorithms were tested using a standard dataset collected from industrial partners Necon Automation and IIUM. The algorithms contained MITM attacks, Web-server access attacks, Telnet attacks, and normal packets.
- Testing was conducted in two stages, namely binary classification and multiclass classification, to achieve the high-performance mode for detecting ICS attacks.
- The ML algorithms KNN, LDA, and DT were examined to find the appropriate algorithm for protecting ICS systems. The KNN and DT algorithms achieved the highest levels of accuracy.
- The hybrid DL CNN-LSTM model proved to be the most effective and efficient algorithm for successfully detecting malware in ICS systems.
- The inaccuracies between the anticipated output and the target values were discovered during the validation phase using a sensitivity analysis that examined the metrics MSE, RMSE, and R2. The KNN and DT algorithms provide fewer prediction errors in binary classification and multiclass classification.
- The ML and DL approaches performed well in the validation phase, with the KNN and DT algorithms outperforming the competition by a wide margin. This study's findings were compared with those of other recent studies, confirming the validity and efficacy of our findings. We developed ML and DL algorithms and conducted experiments with them to get the best malware detection possible in ICSs. Even though both of the suggested classifiers obtained high accuracy, the KNN and decision accuracy were 100%, demonstrating that they can beat other state-of-the-art classifier models.
- In the future, we aim to design our system with a real ICS system for protecting food security.

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

Funding: This research and the APC were funded by the Al Bilad Bank Scholarly Chair for Food Security in Saudi Arabia, The Deanship of Scientific Research, The Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, and Al Ahsa, Saudi Arabia, grant chair number 24. The authors are grateful for financial and moral support.

Data Availability Statement: The data presented in this study are available at https://data.mendeley.com/datasets/k76xhm22yj/1 (accessed on 15 February 2022).

Acknowledgments: The authors express their deepest gratitude to Al Bilad Bank Scholarly Chair for Food Security in Saudi Arabia, The Deanship of Scientific Research, The Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, and Al Ahsa, Saudi Arabia, grant chair number 24. The authors are grateful for financial and moral support.

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

References

1. Oliver, E.; Philipp, K.; Tavolato, P. Identifying S7comm Protocol Data Injection Attacks in Cyber-Physical Systems. In Proceedings of the 2018 Proceedings of the 5th International Symposium for ICSS & SCADA Cyber Security Research, Hamburg, Germany, 29–30 August 2018.
2. Kargl, F.; van der Heijden, R.W.; König, H.; Valdes, A.; Dacier, M.C. Insights on the Security and Dependability of Industrial Control Systems. *IEEE Secur. Priv.* **2014**, *12*, 75–78. [CrossRef]
3. Threats against Industrial Control Systems on the Rise in H2 2020, Growing by Nearly 8 Percentage Points in the Engineering Sector. Available online: https://www.kaspersky.com/about/press-releases/2021_threats-against-industrial-control-systems-on-the-rise-in-h2-2020 (accessed on 19 April 2022).
4. George, G.; Thampi, S.M. A Graph-Based Security Framework for Securing Industrial IoT Networks from Vulnerability Exploitations. *IEEE Access* **2018**, *6*, 43586–43601. [CrossRef]
5. Fan, X.; Fan, K.; Wang, Y.; Zhou, R. Overview of cyber-security of industrial control system. In Proceedings of the 2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC), Shanghai, China, 5–7 August 2015; pp. 1–7.
6. Jiang, B.; Yang, J.; Ding, G.; Wang, H. Cyber-physical security design in multimedia data cache resource allocation for industrial networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6472–6480. [CrossRef]
7. Wang, Y.; Meng, W.; Li, W.; Li, J.; Liu, W.X.; Xiang, Y. A fog-based privacy-preserving approach for distributed signature-based intrusion detection. *J. Parallel Distrib. Comput.* **2018**, *122*, 26–35. [CrossRef]
8. Aloqaily, M.; Otoum, S.; Al Ridhawi, I.; Jararweh, Y. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Netw.* **2019**, *90*, 101842. [CrossRef]
9. Vinayakumar, R.; Alazab, M.; Soman, K.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep learning approach for intelligent intrusion detection system. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
10. Manzoor, I.; Kumar, N. A feature reduced intrusion detection system using ANN classifier. *Expert Syst. Appl.* **2017**, *88*, 249–257.
11. Miller, B.; Rowe, D. A survey SCADA of and critical infrastructure incidents. In Proceedings of the 1st Annual Conference on Research in Information Technology, Calgary, AB, Canada, 11–13 October 2012; pp. 51–56.
12. Nicholson, A.; Webber, S.; Dyer, S.; Patel, T.; Janicke, H. SCADA security in the light of Cyber-Warfare. *Comput. Secur.* **2012**, *31*, 418–436. [CrossRef]
13. Fernández Maimó, L.; Perales Gómez, A.L.; García Clemente, F.J.; Gil Pérez, M.; Martínez Pérez, G. A Self-Adaptive Deep Learning-Based System for Anomaly Detection in 5G Networks. *IEEE Access* **2018**, *6*, 7700–7712. [CrossRef]
14. Fernández Maimó, L.; Huertas Celdrán, A.; Gil Pérez, M.; García Clemente, F.J.; Martínez Pérez, G. Dynamic management of a deep learning-based anomaly detection system for 5G networks. J. Ambient Intell. *Humaniz. Comput.* **2019**, *10*, 3083–3097.
15. Fernández Maimó, L.; Huertas Celdrán, A.; Perales Gómez, A.L.; García Clemente, F.J.; Weimer, J.; Lee, I. Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments. *Sensors* **2019**, *19*, 1114. [CrossRef] [PubMed]
16. Goh, J.; Adepu, S.; Junejo, K.N.; Mathur, A. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In *Critical Information Infrastructures Security*; Havarneanu, G., Setola, R., Nassopoulos, H., Wolthusen, S., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 88–99.
17. Almalawi, A.; Yu, X.; Tari, Z.; Fahad, A.; Khalil, I. An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems. *Comput. Secur.* **2014**, *46*, 94–110. [CrossRef]
18. Tomin, N.V.; Kurbatsky, V.; Sidorov, D.N.; Zhukov, A.V. Machine learning techniques for power system security assessment. In Proceedings of the IFAC Workshop on Control of Transmission and Distribution Smart Grids, Prague, Czech Republic, 11–13 October 2016; pp. 445–450.
19. Zaman, M.; Lung, C. Evaluation of machine learning techniques for network intrusion detection. In Proceedings of the IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–5.
20. Teixeira, M.A.; Salman, T.; Zolanvari, M.; Jain, R.; Meskin, N. SCADA system testbed for cybersecurity research using machine learning approach. *Future Int.* **2018**, *10*, 76. [CrossRef]
21. Almseidin, M.; Alzubi, M.; Kovacs, S.; Alkasassbeh, M. Evaluation of machine learning algorithms for intrusion detection system. In Proceedings of the IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 14–16 September 2017; pp. 277–282.
22. Mathur, A.; Tippenhauer, N. SWaT: A water treatment testbed for research and training on ICSS security. In Proceedings of the International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater), Vienna, Austria, 11 April 2016; pp. 31–36.
23. Perez, R.L.; Adamsky, F.; Soua, R.; Engel, T. Machine learning for reliable network attack detection in SCADA systems. In Proceedings of the 17th IEEE International Conference On Trust, Security And Privacy in Computing And Communications, New York, NY, USA, 1–3 August 2018; pp. 633–638.
24. Jicha, A.; Patton, M.; Chen, H. SCADA honeypots: An in-depth analysis of Conpot. In Proceedings of the IEEE Conference on Intelligence and Security Informatics (ISI), Tucson, AZ, USA, 28–30 September 2016; pp. 196–198.
25. Almomani, O. A hybrid model using bio-inspired metaheuristic algorithms for network intrusion detection system. *Comput. Mater. Contin.* **2021**, *68*, 409–429. [CrossRef]

26. Kravchik, M.; Shabtai, A. Efficient cyber attacks detection in industrial control systems using lightweight neural networks. *arXiv* **2019**, arXiv:1907.01216. [CrossRef]
27. Liu, L.; Hu, M.; Kang, C.; Li, X. Unsupervised Anomaly Detection for Network Data Streams in Industrial Control Systems. *Information* **2020**, *11*, 105. [CrossRef]
28. Tomlin, L.; Farnam, M.R.; Pan, S. A clustering approach to industrial network intrusion detection. In Proceedings of the 2016 Information Security Research and Education (INSuRE) Conference (INSuRECon-16), Huntsville, AL, USA, 30 September 2016.
29. Schneider, P.; Böttinger, K. High-performance unsupervised anomaly detection for cyber-physical system networks. In Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy, Toronto, ON, Canada, 19 October 2018; pp. 1–12.
30. Sfar, A.R.; Natalizio, E.; Challal, Y.; Chtourou, Z. A roadmap for security challenges in the Internet of Things. *Digit. Commun. Netw.* **2018**, *4*, 118–137. [CrossRef]
31. Keshk, M.; Moustafa, N.; Sitnikova, E.; Creech, G. Privacy preservation intrusion detection technique for SCADA systems. In Proceedings of the 2017 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 14–16 November 2017; pp. 1–6.
32. Zhao, K.; Ge, L. A survey on the internet of things security. In Proceedings of the 2013 Ninth International Conference on Computational Intelligence and Security, Leshan, China, 14–15 December 2013; pp. 663–667.
33. Kumar, J.S.; Patel, D.R. A survey on internet of things: Security and privacy issues. *Int. J. Comput. Appl.* **2014**, *90*. [CrossRef]
34. Suo, H.; Wan, J.; Zou, C.; Liu, J. Security in the internet of things: A review. In Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, China, 23–25 March 2012; Volume 3, pp. 648–651.
35. Kouicem, D.E.; Bouabdallah, A.; Lakhlef, H. Internet of things security: A top-down survey. *Comput. Netw.* **2018**, *141*, 199–221. [CrossRef]
36. Mubarak, S.; Habaebi, M.H.; Islam, M.R.; Balla, A.; Tahir, M. Industrial datasets with ICSs testbed and attack detection using machine learning techniques. *Intell. Autom. Soft Comp.* **2022**, *31*, 1345–1360. [CrossRef]
37. Aldhyani, T.H.H.; Alkahtani, H. Attacks to Automatous Vehicles: A Deep Learning Algorithm for Cybersecurity. *Sensors* **2022**, *22*, 360. [CrossRef] [PubMed]
38. Liu, G.; Zhao, H.; Fan, F.; Liu, G.; Xu, Q.; Nazir, S. An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs. *Sensors* **2022**, *22*, 1407. [CrossRef] [PubMed]
39. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [CrossRef]
40. Shah, R.A.; Qian, Y.; Kumar, D.; Ali, M.; Alvi, M.B. Network Intrusion Detection through Discriminative Feature Selection by Using Sparse Logistic Regression. *Future Internet* **2017**, *9*, 81. [CrossRef]
41. Rawat, W.; Wang, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Comput.* **2017**, *29*, 2352–2449. [CrossRef]
42. Alkahtani, H.; Aldhyani, T.H.H. Botnet Attack Detection by Using CNN-LSTM Model for Internet of Things Applications. *Secur. Commun. Netw.* **2021**, *2021*, 3806459. [CrossRef]
43. Alkahtani, H.; Aldhyani, T.; Al-Yaari, M. Adaptive anomaly detection framework model objects in cyberspace. *Appl. Bionics Biomech.* **2020**, *2020*, 6660489. [CrossRef]
44. Gul, F.; Mir, I.; Abualigah, L.; Sumari, P.; Forestiero, A. A Consolidated Review of Path Planning and Optimization Techniques: Technical Perspectives and Future Directions. *Electronics* **2021**, *10*, 2250. [CrossRef]
45. Agostino, F. Metaheuristic algorithm for anomaly detection in Internet of Things leveraging on a neural-driven multiagent system. *Knowl.-Based Syst.* **2021**, *228*, 107241.
46. Kravchik, M.; Shabtai, A. Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks. In Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy, Toronto, ON, Canada, 15–19 October 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 72–83.
47. Shalyga, D.; Filonov, P.; Lavrentyev, A. Anomaly detection for water treatment system based on neural network with automatic architecture optimization. *arXiv* **2018**, arXiv:1807.07282.
48. Zizzo, G.; Hankin, C.; Maffeis, S.; Jones, K. Intrusion Detection for Industrial Control Systems: Evaluation Analysis and Adversarial Attacks. *arXiv* **2019**, arXiv:1911.04278.
49. Inoue, J.; Yamagata, Y.; Chen, Y.; Poskitt, C.M.; Sun, J. Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 1058–1065.
50. Li, D.; Chen, D.; Jin, B.; Shi, L.; Goh, J.; Ng, S.K. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In *Artificial Neural Networks and Machine Learning*; Tetko, I.V., Kurková, V., Karpov, P., Theis, F., Eds.; ICANN 2019: Text and Time Series; Springer International Publishing: Cham, Switzerland, 2019; pp. 703–716.