applied sciences

MDPI

Article

Lightweight Hybrid Deep Learning Architecture and Model for Security in IIOT

Ankita [1], Shalli Rani [1,*], Aman Singh [2,3], Dalia H. Elkamchouchi [4] and Irene Delgado Noya [2,3]

[1] Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, Punjab, India; ankitaanand2719@gmail.com
[2] Higher Polytechnic School, Universidad Europea del Atlántico, C/Isabel Torres 21, 39011 Santander, Spain; amansingh.x@gmail.com (A.S.); irene.delgado@uneatlantico.es (I.D.N.)
[3] Department of Project Management, Universidad Internacional Iberoamericana, Campeche C.P. 24560, Mexico
[4] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; dhelkamchouchi@pnu.edu.sa
* Correspondence: shalli.rani@chitkara.edu.in

Abstract: Remarkable progress in the Internet of Things (IoT) and the requirements in the Industrial era have raised new constraints of industrial data where huge data are gathered by heterogeneous devices. Recently, Industry 4.0 has attracted attention in various fields of industries such as medicines, automobiles, logistics, etc. However, every field is suffering from some threats and vulnerabilities. In this paper, a new model is proposed for detecting different types of attacks and it is analyzed with a deep learning technique, i.e., classifier-Convolution Neural Network and Long Short-Term Memory. The UNSW NB 15 dataset is used for the classification of various attacks in the field of Industry 4.0 for providing security and protection to the different types of sensors used for heterogeneous data. The proposed model achieves the results using Cortex processors, a 1.2 GHz processor, and four gigabytes of RAM. The attack detection model is written in Python 3.8.8 and Keras. Keras constructs the model using layers of Convolutional, Max Pooling, and Dense Layers. The model is trained using 250 batch size, 60 epochs, 10 classes. For this model, the activation functions are Relu and softmax pooling.

Keywords: industrial internet of things; deep learning; security; attacks; privacy

Citation: Ankita; Rani, S.; Singh, A.; Elkamchouchi, D.H.; Noya, I.D. Lightweight Hybrid Deep Learning Architecture and Model for Security in IIOT. *Appl. Sci.* **2022**, *12*, 6442. https://doi.org/10.3390/app12136442

Academic Editor: Faisal Jamil

Received: 20 May 2022
Accepted: 22 June 2022
Published: 24 June 2022

1. Introduction

The combination of Industrial Internet of Things applications and the manufacturing field acts as a big opportunity to enhance the base of the customer. Production-oriented applications, generally, improve the manufacturing process and reduce implementation complexity. They also enhance organizational communication and collaboration inside the company [1]. Likewise, the deployment of automated wireless communication technologies can ensure effective communication at any time and location, facilitating an industrial firm's successful growth and development. The importance of those networks based on the Internet of Things (IoT) in Industrial IoT applications is discussed in this section. Industrial organizations, on the other hand, continue to use conventional methods for continually improving industrial efficiency in tandem with adaptation in supervision, management, and control. This drives a tendency to look into new perspectives in industrial processes, as well as intellectual disciplines, to better understand the difficulties ahead [2]. As a result, the agile manufacturing applications, which allow information interchange, are in conflict with various businesses. There may also be challenges in bringing Industry 4.0 applications to industries. The examination of the elements that cause these issues allows businesses to better govern the transformation process. It also makes the transition easier. There are several solutions to analyze the criteria available in the literature. For this type of investigation, the decision-making procedures based on multiple criteria are useful [3]. The research related to Industrial IoT, also known as Industry 4.0, is generally undertaken in the field of manufacturing in developed countries, as the following conception was conceived

in the world's leading industrial economies [1]. The goal of this study is to assess the application of advanced digital technologies in Industrial Domains and in transition nations in the context of Industrial IoT, where a lot of attacks occur on the Industrial networks where security plays a crucial role. This assessment contains the security approach in comparison with the sophisticated digital technologies using a set of criteria. Meanwhile, the industrial sector has a gradual increase in the number of edge and control devices where various security and privacy issues have surfaced, posing a significant threat to the IIoT's security and credibility [4]. Intruders may be able to take advantage of these edge and control devices. A hacked IoT device might send fake data to cloud servers or allow unwanted access to sensitive corporate papers, financial projections, and company data. This could result in equipment failure, as well as economic damage. One of the biggest issues in the smart industry is ensuring cybersecurity in the IIoT. It comprises protection from malicious software, prevention from unauthorized access, and communication and physical privacy protection. By installing modern and comprehensive security procedures, the IIoT's security, privacy, and trustworthiness may be improved. Deep learning (DL) techniques for designing and developing solutions related to cybersecurity have gained a lot of interest in recent years from both industry and academics. DL techniques have a lot of potential for producing improved results from industries' huge data [5]. Therefore, developing more safe and robust attack detection techniques for the IIoT continues to be a challenge. The remaining paper is composed of the following sections: Section 2 presents the related work. Section 3 presents the proposed hybrid CNN-LSTM model. Section 4 presents the results and discussion part, while Section 5 which is the Conclusions part.

2. Background and Related Work

The authors [6] outlined the number 9 cornerstones of Industrial IoT as well as their associated problems. Big Data Analytics, Intelligent Robotics, Simulation, Longitudinal, and Combined Directional systems, etc. The researchers discussed the issues and challenges associated with Industry 4.0: Automation and self-managing systems must be present in adaptive production systems, contrasting the lack of knowledge of robotics in today's systems. Indian women's network procedures should be sufficient in terms of speed [7]. It is difficult to keep track of data and ensure that they are accurate. The system should be constructed in such a way that it can handle its manufacturing process. Defending against a cyber-attack requires security measures. The authors in [8] developed a theoretical framework for analyzing Industrial IoT adoption patterns. They conducted a study in 92 industrial sectors to assess the adoption of Industry 4.0 front-end and base technologies. The usual categorization of IT security goals is divided into confidentiality, integrity, and availability. The CIA-triad in [9] is the name given to these three categories. Only authorized users have access to certain information, ensuring confidentiality. Integrity refers to the intact information and consistently refers to an underlying occurrence or fact. When authorized users have instant access to information, availability is met in [10]. In recent years, science has progressed beyond additional IT security goals or standards, such as authenticity, accountability, and transparency, defined by the CIA-triad. This interconnection of cyber and physical systems, however, raises security and privacy concerns [11]. As a result, the initial step in maintaining security and privacy is to determine the vulnerabilities being faced. The authors of [12] suggested a zone partitioning-based anomaly detection mechanism for industrial cyber-physical systems based on the creation of a zonal functional model. Intrusion responses for industrial control systems were proposed by researchers in [13]. The authors came up with a supply chain in the field of Industrial IoT. The pros and cons of Industrial IoT concern supply chain, storage, and transportation [14]. The authors of [4] came up with an approach using a Convolutional network for detection of attack known as DoS, and it is 92.9% accurate. The authors in [11] demonstrated a deep learning-based sequential long short-term memory (WDLSTM) network with the CAIDA dataset [11] for anomaly detection, with 96.7% accuracy. A convolutional network is employed for feature extraction from large data for intrusion detection, and the KDD CUP 99 is used for the classification of attacks in the suggested technique. Compared with

existing systems, the proposed method demonstrated improved performance. A migration deep learning-based detection threat for IoT-connected smart cities was introduced in [15]. The researchers used the famous and initial dataset known as KDD CUP 99 to assess the performance of the model. The results of the experiments confirmed the improved accuracy, as well as the detection time for detecting malware attacks in the industrial field. A combined approach based on an improved genetic algorithm and deep belief networks was introduced in [16]. The genetic algorithm for determining hidden layer neurons and the deep belief networks are used to classify the attacks more accurately and efficiently. The proposed approach was evaluated using the NSL-KDD dataset, an attack detection system that works in the IIoT network. The authors created a deep neural network based on the Gaussian Bernoulli Restricted Boltzmann Machine (GBRBM) (DNN). This method turned fault detection into a challenge for classification [17]. A combined technique for binary classification was used to combine the deep network and the k-Nearest Neighbors algorithm (kNN). The CICIDS and NSL-KDD datasets were used to test the researchers' model for failure detection by using a bidirectional long short-term memory approach [18]. The Table 1 is showing comparison of attacks with different datasets.

Table 1. Comparison of Attacks with different datasets.

Ref.No	Dataset	ML/DL Classifier	Type of Attack	Accuracy
[4]	KDDCUP99	CNN	DoS	92.9%
[11]	CAIDA	WDLSTM	anomaly detection	96.7%
[15]	KDDCUP99	CNN	Malware	97.2%
[16]	NSL-KDD	DBN	DoS	96.3%
[17]	DARPA	GBRBM-DNN	Fault Detection	98.5%
[18]	CICIDS and NSL-KDD	Bi-LSTM	Failure Detection	95.5%
Our Model	UNSW NB 15	Hybrid Model	Various types of attacks	99%

3. Proposed Hybrid Model CNN-LSTM

Research works lag in terms of intrusion detection, according to the survey [18], since research models are targeted at detecting single attacks. There are few research models aimed at detecting repeated attacks, and the system's computing cost is quite high, making it inappropriate for a wide range of applications. This suggested research focuses on Convolutional Neural Networks (CNN), which have gained popularity in recent years. A deep learning classifier Convolutional Neural Network (CNN) is used for the classification of data that excels in managing a variety of database collections [19]. IDS based on CNN have only recently evolved, and their performance outperforms that of existing detection approaches. The proposed study model integrates the recurrent neural network (RNN) model-based Long Short-Term Memory (LSTM) to produce remarkable performance in identifying a wide range of network anomalies and malicious assaults, even though convolutional performs better. Figure 1 depicts the current hybrid CNN-LSTM prototype. A suitable dataset is selected, from which original and valuable features can be extracted by deleting redundant features and replacing them with improved ones using a deep learning LSTM model. The Intrusion Detection System can be used for classification and detection of assaults after the model has been trained using CNN and a corresponding weight set has been obtained. A well-known data classification model that excels at handling a wide range of database resources. The performance of convolutional network-based intrusion detection systems has recently improved, and it currently beats earlier detection methods [20]. Although convolutional performs better, the suggested research model

employs LSTM, an RNN diversification, to produce greater performance in the detection of attacks. As illustrated in Figure 1, the proposed model is divided into the following phases: data collection, Feature extraction, and Improved features; then the training can be performed by the convolutional network, and attack detection. The collecting of network vital data is an important step in this procedure. After collecting the intrusion dataset, the preprocessing is conducted following the data cleaning steps such as dealing with the missing and filling values along with the noisy data. Afterwards, features which are relevant can be extracted as per the requirement. The following step is used to make the training set, feature set, and testing set. The training model will obtain fine-tuned data using one or many convolutional layers, connection weights of the neuron, and outputs (see Figure 2). Finally, the identification step works with the data trained, and obtained information is merged for weight production, and the training period is employed for threat identification. The input layer with a matrix set p_0 q_0 and an output layer with a group of neurons for each label makes up the network structure. This research focuses on creating features that can be tweaked to capture the attacks and how they are effective in detecting in the industrial sector. Even though the feature can be any created signature based on any physical measurements, the proposed attack detection framework employs three types of features to capture both time and space attributes of physical systems: physics, learning, and statistics-based features. This is utilized in calculations involving space. The following study is used to calculate multiple and univariate characteristics, as well as independent (univariate) data. To capture the dynamics of the entire system or temporal holdings, the characteristics are computed on a sliding window [21]. Assume that m physical measurements exist: $a^1; a^2; ...a^m$. There are three measurement parameters, such as actuator, sensor, and perhaps periodically control change limitations, and the sliding window's width can be taken as s. This process is used to collect critical information from nodes, as well as infectious node identification.Long Short-Term Memory is used in the following step to learn all the patterns through the network [22]. The LSTM model, which is based on a recurrent neural network, uses time stamps to generate the output. However, using LSTM alone to identify network intrusion is unsuccessful due to a problem with gradient vanishing that prohibits it from learning information for a long period. However, the efficiency of the LSTM is far better for short periods, and the system's complexity is drastically decreased. A string of inputs is passed to the connected hidden layers, which produce the desired outputs in a traditional LSTM. Bidirectional LSTM models were eventually developed, which used two hidden layers to process input strings in both forward and backward orientations (see Figure 3).

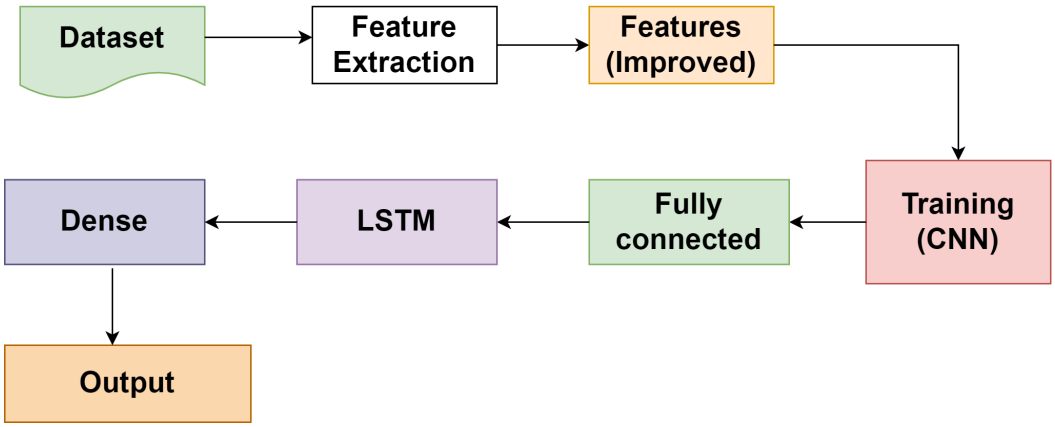

Figure 1. The Proposed hybrid model.

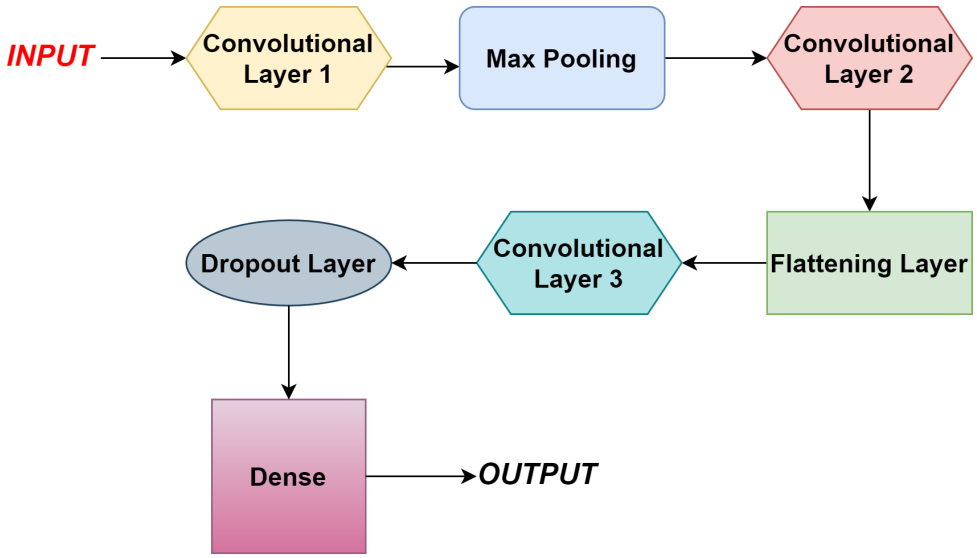

Figure 2. CNN architecture for classification of attacks in IoT Networks.

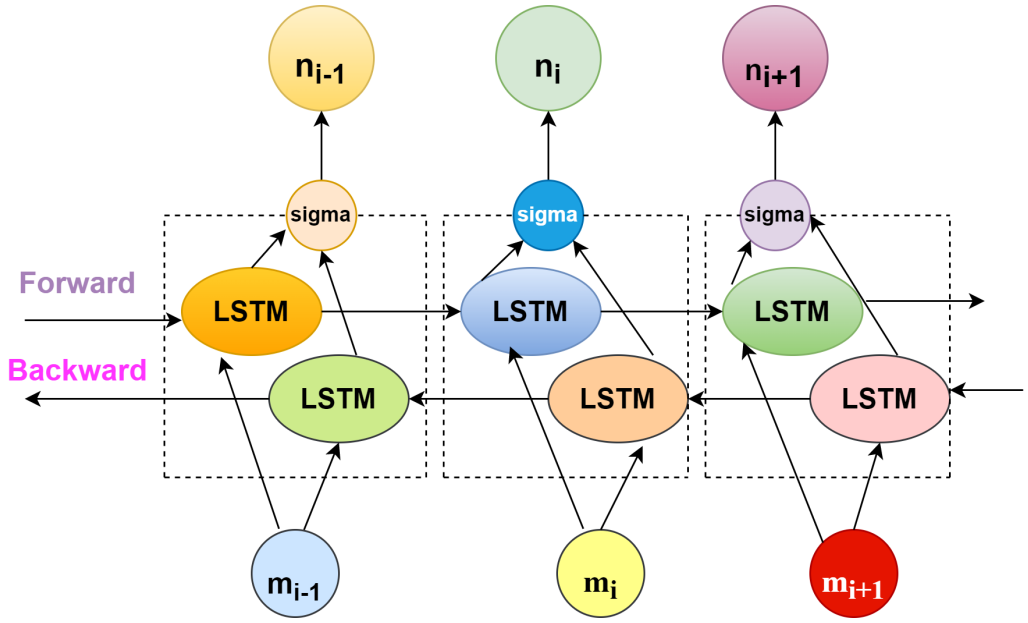

Figure 3. Working of LSTM.

These parameters have an impact on data transmission and retrieval through the network. The input features are labeled in the training phase, and a neuron-based feature matrix is assigned to each element in the input level. The connection weights of a neuron and matching weight matrix for such a convolutional layer can be constructed after the inputs have been processed. The configuration has p convolutional layers, with vector layers denoted by f, output neurons denoted by q, and the weight function denoted by w. The weight functions for the output layer q can be computed as in Equation (1):

$$w_y = \sum_{n=1}^{N} \sum_{m=1}^{M} (p_n * f_n + 1) + p * f \tag{1}$$

where the total weight of the network is w_y, the size of a filter of the window is $p_n * f_n$, and the number of convolutional filters can be denoted by M [23]. The input with its related weights generates the loss function-based label throughout the training process.

The loss function is given as and is derived from the compact size and validation data of the corresponding model as in Equation (2).

$$L_f = L_c + \phi * L_d \tag{2}$$

The scaling parameter ϕ is utilized to evaluate the precedence of nodes to calculate the next two metrics' compact size and descriptiveness [24]. The important features are extracted from the supplied dataset and transformed into a feature matrix in the detection step. The matrix function assists in calculating the weight set in the training phase, and the output layer determines neuron labeling. The outputs are monitored using the activation function, and are given as in Equation (3):

$$f_m(y) = \frac{e_m^0}{\sum_{m-1}^{n} e_n^0} \tag{3}$$

The probability of each neuron's output and weight of a fully connected layer can be calculated by the activation function. In general, the activation function's value ranges from 0 to 1, with the largest value representing the output label.

4. Results and Discussion

4.1. Preprocessing and Testing

The data are obtained and exported in readable format, therefore, before the execution process, the data are to be preprocessed. The proper format of data is taken. The sheer volume and dimensionality of the data make it compulsory to perform proper data preprocessing to clean and format data before any training. The first thing to account for is that some of the data may be missing or corrupt. After this initial cleansing and formatting of the following dataset, the horizontal complexity imposes some preliminary feature selection to guide the test. The suggested proposed system has been tested in the lab and compared to a CNN and RNN-based IDS. The data set for the proposed methodology is used for the ratio of 70% and 30%. The 70% is used for training and the other 30% is used for testing. The developed scheme analyzes the dataset for features and distinguishes between attack and normal conditions.

This is one of the latest IIoT datasets, which is widely used to assess the effectiveness of cybersecurity systems based on machine learning. There are a total of 257,673 samples in this dataset, with 164,673 inaccurate results and 93,000 normal values. There are 49 features and 9 types of threat in this dataset. Following are the types of attacks such as backdoor, analysis, reconnaissance, exploit, generic, fuzzer, DoS, worm, and shellcode assaults. The authors in [12] identify the network size of a packet, IP addresses of both source and destination, ports, set of rules, and so on, that can be used to forecast harmful malicious activity. The UNSW-NB15 dataset is used in the cited paper to train and test their model. For creating their model and assessing predictability, the four machine learning classifiers are used. The results revealed that the machine learning classifiers' decision tree has better results, with a 93 percent accuracy rate. Some of the datasets given did not prefer a realistic testbed, and some of the threat scenarios were not varied. The model requires the following dataset for the tests due to its attributes such as regular updates, substantial attack heterogeneity, and integration of the traffic generated by IoT. To begin, both the training and testing datasets are loaded and the fundamental libraries needed to run the program are imported. The dataset is checked for missing values during the preprocessing step. The resulting dataset is well organized, therefore, there are no null values in the dataset. As a result, utilizing functions separates the different attacks from normal traffic data. The fundamental problem with this particular dataset is that data entries of the malicious traffic are much larger than the regular traffic data entries. The legitimate and malignant traffic data pieces were reduced to a 70–30 ratio as a result of the trimmed or pruned data of any attack. As a result of huge positive and few negative examples during training in binary classification, the model will prefer to forecast. In that situation, the model is quite accurate.

4.2. Steps to Load Data

The following model achieves the results using processors, with four gigabytes of RAM. The attack detection model is written in Python 3.8.8 and Keras. Keras constructs the model using layers of Convolutional, Max Pooling, and Dense Layers. The use of a dense layer is to alter the dimensions of the vector. This layer additionally manipulates the vector's rotation, scaling, and translation parameters. Following that, the hyperparameters are optimized, as shown below, and the model is trained using the values of the hyperparameters with 20 epochs and a batch size of value 64. The number of classes required is a value of 25, to analyze the detection of attacks. The activation function in the following hybrid model is ReLU.

The ReLU sequence is followed by the convolution layer, which is used for mapping the features present in the output layer. The ReLU function is defined as follows in Equation (4):

$$f(a_0) = max(0, a_0) \tag{4}$$

4.3. Analysis and Discussion

This section discusses the various settings as well as the outcomes of using the hybrid CNN-LSTM model. The following are the performance metrics on which the dataset is based. The efficiency of the dataset utilized to provide the best and safest findings affects performance metrics.

The first step is to train the model using a batch size of 250, epoch 60, and some classes 10, the activation function used is relu along with the softmax pooling. Table 2 discusses the various parameters along with their values.

Table 2. Hyperparameters with their Values.

Hyperparameters	Values
size of batches	250
epoch	60
number of classes	10
activation function	relu
pooling	softmax

After concluding the model, it is trained with the preprocessed data. For the training phase, epochs and batch size are specified. When approaching the minimum of a loss function, the learning rate regulates the step size at each cycle. An epoch is a complete presentation of the data set to be taught to a learning machine, whereas a batch size is merely the number of training cases used in one iteration. The correlation matrix between different features of the dataset is obtained and shown in Figure 4. Following the model's training, testing is carried out as described in the preceding section. As a result, a confusion matrix with Predicted Labels and True Labels is obtained. The confusion matrix of all the attacks is represented in Figure 5. Both confusion matrix and correlation matrix relate to each other as the confusion matrix is the relationship between some features of the correlation matrix. The ROC curve is a graph that shows how well The Receiver Operating Characteristic curve depicts the productivity and effectiveness of a classification model at each classification level. Figure 6 depicts the ROC curve for multi-class which is the plot between the True positive and the False positive rate. The various characteristics such as accuracy (Acc), recall (Rec), and precision (Prec) are calculated and the results are shown in Figures 7 and 8, with statistical results including mean absolute error, mean squared error, and root mean squared error.

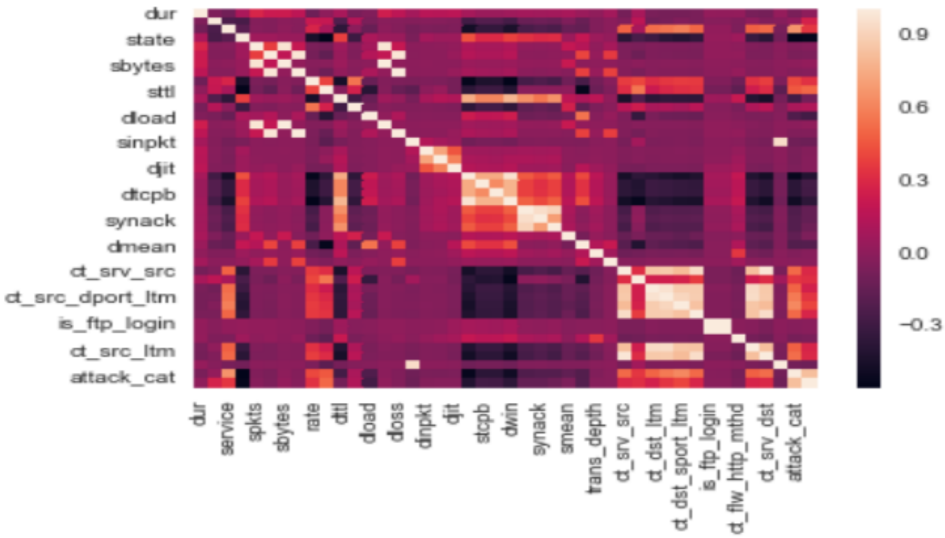

Figure 4. Correlation Matrix.

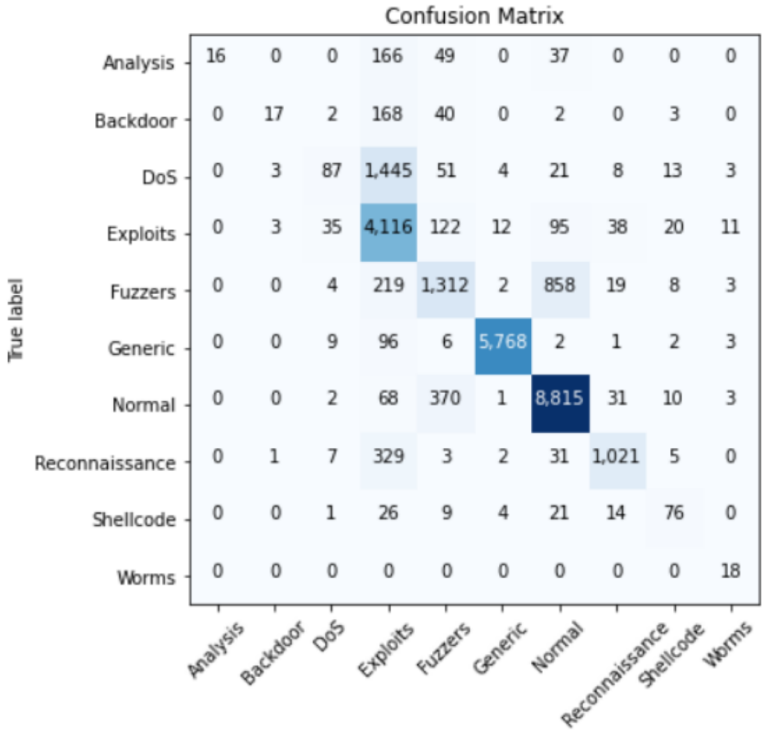

Figure 5. Confusion Matrix.

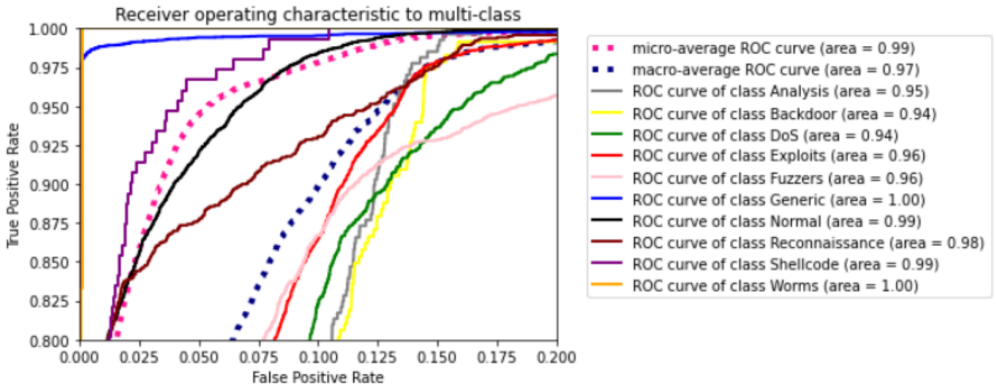

Figure 6. ROC curve.

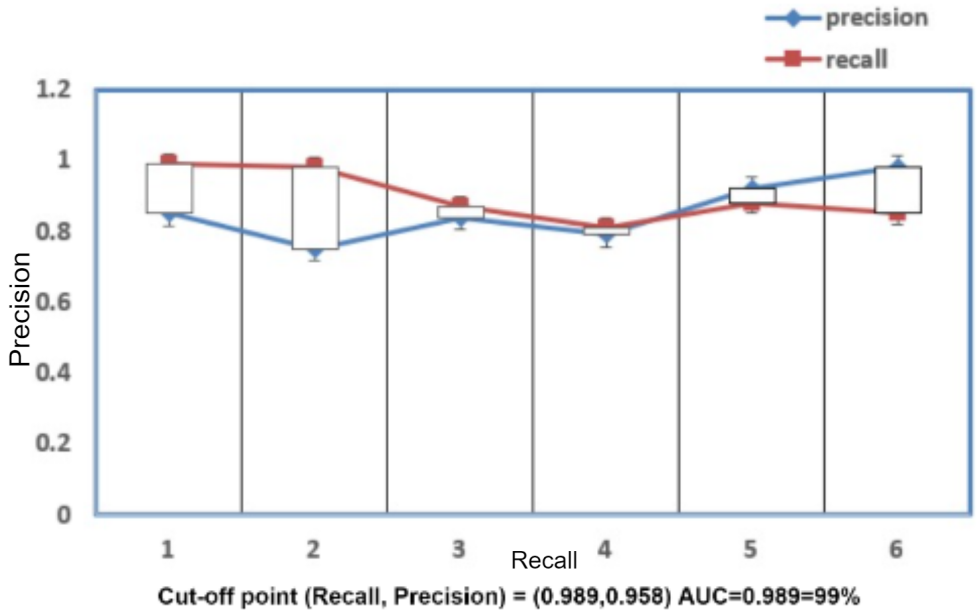

Figure 7. Precision and Recall.

```
Mean Absolute Error -  0.06800262812089355
Mean Squared Error -  0.20532194480946123
Root Mean Squared Error -  0.4531246459965086
```

Figure 8. Statistical Results.

Performance Metrics

- **Accuracy:** It is the percentage of those predictions which are correct to the total True Positive predictions made by the model.

$$\text{Accuracy (Acc)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

where TP = class predictions True positive, TN = True Negative class prediction, FP = False positive class prediction, FN = False negative class prediction.

- **Recall:** It is the ratio of the percentage as true predictions over an actual number of true predictions made by the model.

$$\text{Recall (Rec)} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6}$$

- **Precision:** It is defined as the ratio of the true predictions which are actual over the total true predictions made by the model.

$$\text{Precision (Prec)} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{7}$$

We integrated charting methods using the code, so that the loss functions can be visualized while keeping track of the epoch values. Figure 9 presents the comparison of state-of-the-art techniques.

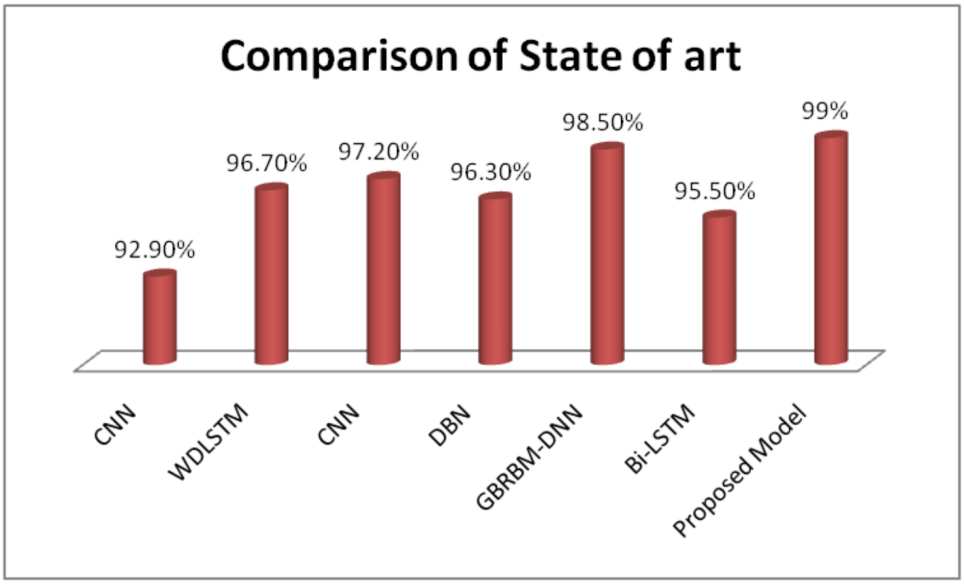

Figure 9. Comparison with the state-of-the-art techniques.

5. Conclusions

The secure smart industrial domain is an important domain, as it holds sensitive data and records which are of major concern. To maintain security and privacy, the deep neural networks CNN with bidirectional LSTM are used in this paper. A predetermined number of epochs, the size of batches sampled, and the number of classes are the requirements or hyperparameters utilized to experiment to develop a model. To classify attacks, the model uses the UNSW NB15 dataset. Using the following hybrid approach in real-time will face the constraint of accuracy as it is not checked on the dynamic dataset and obtaining more consistent outcomes for different types or combinations of deep learning techniques on varied datasets is another area of investigation for future research.

Author Contributions: A. and S.R.; methodology, A., S.R., A.S.; validation, A., D.H.E. and I.D.N.; formal analysis, A. and D.H.E.; investigation, D.H.E. and I.D.N.; resources, S.R.; data curation, D.H.E. and I.D.N.; writing—original draft preparation, A. and S.R. All authors have read and agreed to the published version of the manuscript.

Funding: The Research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R238), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Acknowledgments: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R238), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

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

References

1. Oesterreich, T.D.; Teuteberg, F. Understanding the implications of digitisation and automation in the context of Industry 4.0: A triangulation approach and elements of a research agenda for the construction industry. *Comput. Ind.* **2016**, *83*, 121–139. [CrossRef]
2. Soldani, D.; Fadini, F.; Rasanen, H.; Duran, J.; Niemela, T.; Chan-Dramouli, D.; Nanavaty, N. 5G Mobile Systems for Health-care. In Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, Australia, 4–7 June 2017. [CrossRef]
3. Fera, M.; Macchiaroli, R.; Fruggiero, F.; Lambiase, A. A new perspective for production process analysis using additive manufacturing—Complexity vs. production volume. *Int. J. Adv. Manuf. Technol.* **2018**, *95*, 673–685. [CrossRef]
4. Almon, L.; Riecker, M.; Hollick, M. Lightweight Detection of Denial-of-Service Attacks on Wireless Sensor Networks Revisited. In Proceedings of the Conference on Local Computer Networks, LCN, Singapore, 9–12 October 2017; Volume 2017, pp. 444–452. [CrossRef]
5. Yang, S. Research on network behavior anomaly analysis based on bidirectional lstm. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019; pp. 798–802
6. Longadge, M.; Dongre, M.S.S.; Malik, D.L. Class imbalance problem in data mining: Review. *Int. J. Comput. Netw. (IJCSN)* **2013**, *2*.
7. Sultana, N.; Chilamkurti, N.; Peng, W.; Alhadad, R. Survey on sdn based network intrusion detection system using machine learning approaches. *Netw. Appl.* **2019**, *12*, 493–501. [CrossRef]
8. Bera, A. 80 Mind-Blowing IoT Statisitics (Infographic). February 2019. Available online: https://safeatlast.co/blog/iot-statistics/ (accessed on 11 July 2019).
9. Chernyshev, M.; Baig, Z.A.; Bello, O.; Zeadally, S. Internet of Things (IoT): Research, Simulators, and Testbeds. *IEEE Internet Of Things J.* **2018**, *5*, 1637–1647. [CrossRef]
10. Monika, R.; Tian, G.Y.; Chambers, Y. Deep learning models for cyber security in IoT networks. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019.
11. Available online: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ (accessed on 19 May 2022)
12. Arjoune, Y.; Salahdine, F.; Islam, M.S.; Ghribi, E.; Kaabouch, N. A Novel Jamming Attacks Detection Approach Based on Machine Learning for Wireless Communication. In Proceedings of the International Conference on Information Networking (ICOIN), Bangkok, Thailand, 13–16 January 2020; pp. 459–464.
13. Raj, J.S. Machine Learning Based Resourceful Clustering With Load Optimization for Wireless Sensor Networks. *J. Ubiquitous Comput. Commun. Technol. (UCCT)* **2020**, *2*, 29–38.
14. Zhang, Y.; Li, P.; Wang, X. Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network. *IEEE Access* **2019**, *7*, 31.711–31.722. [CrossRef]
15. Butun, I.; Österberg, P.; Song, H. Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. *IEEE Commun. Surv. Tutorials* **2020**, *22*, 616–644. [CrossRef]
16. Amanullah, M.A.; Habeeb, R.A.A.; Nasaruddin, F.H.; Gani, A.; Ahmed, E.; Nainar, A.S.M.; Akim, N.M.; Imran, M. Deep learning and big data technologies for IoT security. *Comput. Commun.* **2020**, *151*, 495–517. [CrossRef]
17. Adat, V.; Gupta, B.B. Security in Internet of Things: Issues, challenges, taxonomy, and architecture. *Telecommun. Syst.* **2018**, *67*, 423–441. [CrossRef]
18. Maghrebi, H.; Portigliatti, T.; Prouff, E. Breaking cryptographic implementations using deep learning techniques. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 3–26.
19. Papaioannou, M.; Karageorgou, M.; Mantas, G.; Sucasas, V.; Essop, I.; Rodriguez, J.; Lymberopoulos, D.A. Survey on Security Threats and Countermeasures in Internet of Medical Things (IoMT). *Trans. Emerg. Telecommun. Technol.* **2020**, *33*, e4049.
20. Kim, J.; Shin, Y.; Choi, E. An intrusion detection model based on a convolutional neural network. *J. Multimedia Inf. System* **2019**, *6*, 165–172. [CrossRef]
21. Shurman, M.; Khrais, M.R.; Yateem, A.A. IoT Denial-of Service Attack Detection and Prevention Using Hybrid IDS. In *Book IoT Denial-of-Service Attack Detection and Prevention Using Hybrid IDS*; IEEE: Piscataway, NJ, USA, 2019; pp. 252–254.
22. Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener. Comput. Syst.* **2018**, *82*, 761–768. [CrossRef]
23. Ponomarev, S.; Atkison, T. Industrial control system network intrusion detection by telemetry analysis. *IEEE Trans. Dependable Secur. Comput.* **2020**, *13*, 252–260. [CrossRef]
24. Goyal, P.; Sahoo, A.K.; Sharma, T.K.; Singh, P.K. Internet of Things: Applications, security and privacy: A survey. *Mater. Today Proc.* **2021**, *34*, 752–759. [CrossRef]