The International Journal of Advanced Manufacturing Technology (2022) 123:2017–2029
https://doi.org/10.1007/s00170-022-10259-3

ORIGINAL ARTICLE

A novel fully convolutional neural network approach for detection and classification of attacks on industrial IoT devices in smart manufacturing systems

Mohammad Shahin[1] · F. Frank Chen[1] · Hamed Bouzary[1] · Ali Hosseinzadeh[1] · Rasoul Rashidifar[1]

Received: 14 June 2022 / Accepted: 6 October 2022 / Published online: 26 October 2022

Abstract
Recently, Internet of things (IoT) devices have been widely implemented and technologically advanced in manufacturing settings to monitor, collect, exchange, analyze, and deliver data. However, this transition has increased the risk of cyber-attacks, exponentially. Subsequently, developing effective intrusion detection systems based on deep learning algorithms has proven to become a reliable intelligence tool to protect Industrial IoT devices against cyber threats. This paper presents the implementation of two different classifications and detection utilizing the long short-term memory (LSTM) architecture to address cybersecurity concerns on three benchmark industrial IoT datasets (BoT-IoT, UNSW-NB15, and TON-IoT) which take advantage of various deep learning algorithms. An overall analysis of the performance of the proposed models is provided. Augmenting the LSTM with convolutional neural network (CNN) and fully convolutional neural network (FCN) achieves state-of-the-art performance in detecting cybersecurity threats.

Keywords Malicious attacks · Industrial IoT · Machine learning · Classification and detection · Cybersecurity · Big data

1 Introduction

The continuous integration of cyber-physical systems (CPS) into the Internet has led to a boom in smart IoT devices and the emergence of various applications of Industry 4.0 [1, 2] such as smart manufacturing. A smart manufacturing system is heavily made up of complex networks of large-scale CPS that are safety–critical and rely on networked and distributed control architectures [3]. The decreasing cost of sensors and advanced single board computers combined with better access to high bandwidth wireless networks (currently in its fifth generation—5G) have encouraged the proliferation of the Internet of Things (IoT) systems into manufacturing systems [4]. However, those who choose to reap the benefits of IoT systems have to also face the ever-growing threat of exposure to attacks. Thus, the security of IoT systems has become a very critical issue for individuals and businesses. IoT systems have been targeted by malicious third parties and the trend has been increasing exponentially in numbers and growing in complexity and diversity after the emergence of Mirai in 2016 [5]. It has been reported that during the years 2013–2017, not a single month has gone by without news about sensitive user information data being exposed on the web due to an online breach to a certain enterprise [6]. According to the Industrial Control Systems Monito Newsletter issued by the U.S. Department of Homeland Security, it is estimated that one-third of these cyber-attacks target the manufacturing sector making manufacturing systems at the heart of such attacks [7, 8]. Moreover, based to the National Institute of Standards and Technology (NIST)—part of the U.S. Department of Commerce—, these attacks via cyberspace, target an enterprise's use of cyberspace to disrupt, disable, destroy, or maliciously controlling a computing environment/infrastructure; or destroy the integrity of the data or steal controlled information [9]. To address the increased risks and challenges of the growing number and potential of cyber-attacks, realistic protection and investigation countermeasures such as network intrusion detection and network forensic systems need to be developed effectively [10, 11]. Although, several research have been done to solve and decrease the risk of cyber-attacks with different machine learning models and algorithems [10, 11], it

✉ F. Frank Chen
FF.Chen@utsa.edu

[1] Mechanical Engineering Department, The University of Texas at San Antonio, San Antonio, USA

is necessary to implement novel and efficient methods to keep protections updated. In this paper, for the first time, we propose and compare the use of two novel models, reliable, and effective data analytics algorithms for time series classification on three different and unique datasets. The first approach is long short-term memory fully convolutional network (LSTM-FCN) and the second approach is convolutional neural network with long short-term memory (CNN-LSTM). The results of the current study show how such approaches can be utilized to enhance the deterrence level of malicious attacks in industrial IoT devices.

2 Background

The last three decades have been marked by a significant increase in available data and computing power. Nowadays, data analytics is at the forefront of the war against cyber-attacks. Cybersecurity experts have been utilizing data analytics not only to improve the cybersecurity monitoring levels over their network streams but also to increase real-time detection of threat patterns and to conduct surveillance of real-time network streams [12–14]. Both supervised learning and unsupervised learning techniques in data analytics have been used in the detection process of malicious attacks [12, 15]. One of the special features of neural networks (NN) is that they can be used in both supervised and unsupervised learning processes. NN were inspired by the way the human brain works. NN is composed of different data layers which makes them the best-suited algorithms to be used in different artificial intelligence (AI) and machine learning (ML) applications.

Recurrent neural networks (RNNs) propagate data forward and also backward from later processing stages to earlier stages (networks with cyclic data flows that can be used for applications in natural language processing and speech recognition) [16]. RNN was used to achieve a true positive rate of 98.3% at a false positive rate of 0.1% in detecting malware [17]. In another recently published paper, Shibahara et al. [18] used RNN to detect malware based on network behavior with high precision. Also, Loukas et al. [19] have used RNN on a vehicle's real-time data [19] to develop a mathematical model to detect cyber-physical intrusion for vehicles using a deep learning (DL) approach. Despite many advantages, one problem with RNN is that it can only memorize part of the time series which results in lower accuracy when dealing with long sequences (vanishing information problem). To solve this problem, the RNN architecture is combined with long short-term memory (LSTM) [20]. An RNN-LSTM approach has been used in intrusion detection systems to detect botnet activity within consumer IoT devices and networks [21, 22].

LSTM [20] refers to neural networks that are capable of learning order dependence in sequence prediction and able to remember a lot of previous information using back propagation (BP) or previous neuron signals and include it in the current processing. LSTM can be leveraged with various other architectures of NN. The most noticeable application for such network builds is seen in text prediction, machine translation, speech recognition, and more [16, 23]. LSTM suggests an improvement to the RNN model by replacing the hidden layer nodes with three gates structure (forgetting, input, output) that acts on memory cells through the Sigmoid function. These memory cells are responsible for trading-off information by storing, recording, and updating past data [24].

Convolutional neural network (CNN) uses a feed-forward topology to propagate signals, CNN is more often used in classification and computer vision recognition tasks [16, 25]. Kim et al. [26] used KDD CUP 1999 and CSE-CIC-IDS2018 data sets to develop a CNN model to detect denial-of-service category intrusion attacks, early results showed a high accuracy detection that ranged between 89–99%. CNN was also used by Wang et al. [27], McLaughlin et al. [28], and Gibert et al. [29] to detect malware. The latter evaluated their technique using a Microsoft Malware Classification Challenge dataset and managed to outperform other methods in terms of accuracy and classification time. Wang et al. [27] proposed a malware traffic classification method using a CNN by taking traffic data like images and then presented his method as a new taxonomy of traffic classification from an artificial intelligence perspective. In a unique study, Yu et al. [30] suggested a neural network architecture that combines CNN with autoencoders to evaluate network intrusion detection models. Also, Kolosnjaji et al. [31] proposed neural network architecture that consisted of CNN combined with RNN to better detect malware from a VirusShare dataset showing that this newly developed architecture was able to achieve an average precision of 85.6%. The same approach was also utilized by Mac et al. [32] and Yu et al. [33], to detect domain generating algorithms codes that provide malware with new demands on the fly to prevent their servers from being detected and flagged. In conclusion, CNN is a DL network architecture that learns directly from data without the necessity of manual feature extraction. It is worth noting that CNN can also be very effective for classifying time series, and signal data.

A fully convolutional neural network (FCN) is a CNN without fully connected layers [34]. A major advantage of using FCN models is that it does not require heavy preprocessing or feature engineering since their' neuron layers are not dense (fully connected) [35]. FCN has been used [36] to detect fake fingerprints and it was shown that FCN provides high detection accuracy in addition to less processing times and fewer memory requirements compared to other NN. In

this paper, LSTM will be combined with FCN and CNN to show how these two models can be used to accurately detect cybersecurity threats with three different datasets. LSTM gives any NN model the ability to almost seamlessly model problems with multiple input features.

3 Preprocessing of datasets

Network intrusion detection systems (NIDS) based on DL algorithms have proven to be a reliable network protection tool against cyber-attacks [37]. In this paper, we applied state-of-the-art DL algorithms on three benchmark NIDS datasets known as UNSW-NB15, BoT-IoT, and ToN-IoT. These three different datasets were released by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) in the years 2015, 2018, and 2020, respectively.

3.1 Preprocessing of the Bot-IoT dataset

The Bot-IoT dataset [11] contains roughly 73 million records (instances). The BoT-IoT dataset was created by the Cyber Range Lab of UNSW Canberra. The process involved designing a realistic network environment that incorporated a combination of normal and botnet traffic. For better handling of the dataset, only 5% of the original set was extracted using MySQL queries. The extracted 5%, is comprised of 4 files of approximately 1.07 GB total size, and about 3 million records [38–42]. The dataset includes a range of attack categories as shown in Table 1.

This data set contains 45 explanatory features and one binary response feature (Attack or Benign), only 16 of the 45 features were used as input to our models. In all conducted deep learning models and for all used datasets, feature selection was employed when the algorithm itself extracts the important features.

Furthermore, an upsampling technique [46, 47] was used to overcome the heavily imbalanced binary response feature. The feature contained only 13,859 minority counts of benign compared to a whapping 586,241 majority counts of attack. Upsampling procedure prevents the model from being biased toward the majority label. The existing data points corresponding to the outvoted labels were randomly selected and duplicated into the training dataset.

Since input numerical features have different units which means that they have different scales, the SKlearn Standard Scaler was utilized to standardize numerical features by subtracting the mean and then scaling to unit variance by dividing all the values by the standard deviation [48].

DL models require all features to be numeric. For categorical features where no ordinal relationship is in existence, the integer encoding (assigning an integer to each category) can be misleading to the model and results in poor performance or unexpected results (predictions halfway between categories) as it allows the model to assume a natural ordering between categories. In this case, a one-hot encoding can be applied to the categorical representation [49].

3.2 Preprocessing of the UNSW-NB15 dataset

The UNSW-NB15 dataset was created by capturing 100 GB of the raw traffic data packets using the IXIA PerfectStorm tool at the Cyber Range Lab of UNSW Canberra. It was created by generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors [37, 50–53]. Table 2 shows a list of all the attacks that were generated.

This data set contains 48 explanatory features and one binary response feature (Attack or Benign), only 42 of the

Table 1 List of attacks in the Bot-IoT dataset [37]

Attack type	Definition	Counts
Denial-of-service (DoS)	A type of attack is a malicious attempt to overflow the internet traffic of an IoT device or its' surrounding infrastructure (sensors). The attack leaves the IoT device unavailable making it inaccessible to its intended users	56,833 counts of recorded attacks
Distributed denial-of-service (DDoS)	A type of attack similar to a DoS attack but uses multiple attack resources (computers) to flood a targeted IoT device	56,844 counts of recorded attacks
Operating system scan (OS scan) also known as reconnaissance or prope	A type of infiltration attack that uses Nmap tool to scan the operating system of the targeted IoT device to capture network vulnerabilities [43]	470,655 count of recorded attacks
Keylogging (theft)	A type of information theft in which an attacker compromises a remote host for an IoT device to record the administrator's keystrokes, potentially stealing sensitive information [44, 45]	Both Keylogging and Data Exfiltration (theft attacks) had a count of 1909 recorded attacks
Data exfiltration (theft)	A type of information theft in which an attacker compromises an IoT device to gain unauthorized access to data to transfer it on a remote attacking machine [44, 45]	Both keylogging and data exfiltration (theft attacks) had a count of 1909 recorded attacks
Benign (no attack)	Just normal unmalicious flow of data traffic	13,859 counts of no attacks

Table 2 List of attacks in the UNSW-NB15 dataset [37]

Attack type	Definition	Counts
DoS	An attack that aims to prevent access or availability to data by overloading a computer system's resources with traffic	5051 counts of recorded attacks
Fuzzers	An attack that aims to discover security vulnerabilities in a system. Then cause it to crash by sending large amounts of random data	19,463 counts of recorded attacks
Cross-site script (XSS) or analysis	An attack that targets web applications or end-users through ports, emails, and scripts	1995 counts of recorded attacks
Backdoor	An attack that bypasses security mechanisms by replying to specific constructed client applications	1782 counts of recorded attacks
Exploits	An attack that executes a sequence of commands to control the behavior of a host to exploit a vulnerability	24,736 counts of recorded attacks
Generic	An attack that targets cryptography	5570 counts of recorded attacks
Reconnaissance	A type of infiltration attack that uses Nmap tool to scan the operating system of the targeted IoT device to capture network vulnerabilities [43]	12,291 counts of recorded attacks
Shellcode	A malware attack	1365 counts of recorded attacks
Worms	An attack that replicates itself to spread to other computers	153 counts of recorded attacks
Benign (no attack)	Just normal unmalicious flow of data traffic	550,712 counts of no attacks

48 features were used as input to our models. Similar to the previous dataset, One-Hot Encoder was used to encode categorical features [49].

Since our input features contained a significant amount of outliers, the SKlearn Robust Scaler was used to scale the features and make them robust to outliers. This can be achieved by calculating the median and the interquartile range (IQR). The values of each feature then have their median subtracted and are divided by their IQR [54].

3.3 Preprocessing of the TON-IoT network dataset

This is one of the newly generated datasets in an Industry 4.0 environment. The dataset can be used to evaluate the fidelity and efficiency of different cybersecurity applications based on various DL algorithms [55–57]. The datasets were collected from a realistic and large-scale network designed at the Cyber Range and IoT Labs of UNSW Canberra [58–62]. Table 3 shows a list of all the attacks that were generated.

Table 3 List of attacks in the TON_IoT network dataset [37]

Attack type	Definition	Counts
DoS	An attack that aims to prevent access or availability to data by overloading a computer system's resources with traffic	17,717 counts of recorded attacks
Backdoor	An attack that bypasses security mechanisms by replying to specific constructed client applications	17,247 counts of recorded attacks
DDoS	A type of attack similar to a DoS attack but uses multiple attack resources (computers) to flood a targeted IoT device	326,345 counts of recorded attacks
Reconnaissance	A type of infiltration attack that uses Nmap tool to scan the operating system of the targeted IoT device to capture network vulnerabilities [43]	21,467 counts of recorded attacks
Injection	An attack that injects untrusted SQL and codes to alter the course of execution	468,539 counts of recorded attacks
Man in the middle (MITM)	An attack that intercepts traffic and communications between the victim and the host with which the victim is trying to communicate	1295 counts of recorded attacks
Password	An attack that aims at recovering or retrieving passwords	156,299 counts of recorded attacks
Ransomware	An attack that takes control over the victim's files or devices then asks for compensation in exchange for bringing it back to normal	142 counts of recorded attacks
XSS	An attack that targets web applications or end-users through ports, emails, and scripts	99,944 counts of recorded attacks
Benign (no attack)	Just normal unmalicious flow of data traffic	270,279 counts of no attacks

Table 4 Summary of the characteristics of the datasets and the deployed preprocessing techniques

	BoT-IoT	UNSW-NB15	TON-IoT
Balanced data		✓	✓
Upsampling used	✓		
Outliers being overrepresented		✓	✓
Feature scaling	✓	✓	✓
Robust scalar used		✓	✓
Standard scaler used	✓		
Use of one-hot encoding	✓	✓	✓
Released year	2018	2015	2020
Total #of features	46	49	46
Benign to attack ratio	0.2 to 10	7.61 to 1	2.41 to 10

This data set contains 45 explanatory features and one binary response feature (attack or benign), only 20 of the 45 features were used as input to our models. Similar to the UNSW-NB15 dataset, One-Hot Encoder was used to encode categorical features [49] and SKlearn Robust Scaler was used to scale the features and make them robust to outliers [54].

Table 4 shows a summary of the datasets' characteristics and the preprocessing techniques that were used on them. Note that the benign to attack ratio for the BoT-IoT dataset is imbalanced and thus upsampling was used on it.

4 Results and analysis

Two main architectures were proposed to generate our four different models on the three different datasets, CNN-LSTM (Fig. 1) and LSTM-FCN (Fig. 2). The proposed CNN-LSTM architecture uses a one-dimensional convolutional hidden layer that operates over a 1D sequence with 3 filters (collection of kernels that are utilized to store values learned during the training process) and a kernel size of 32. The convolutional hidden layer is accompanied by

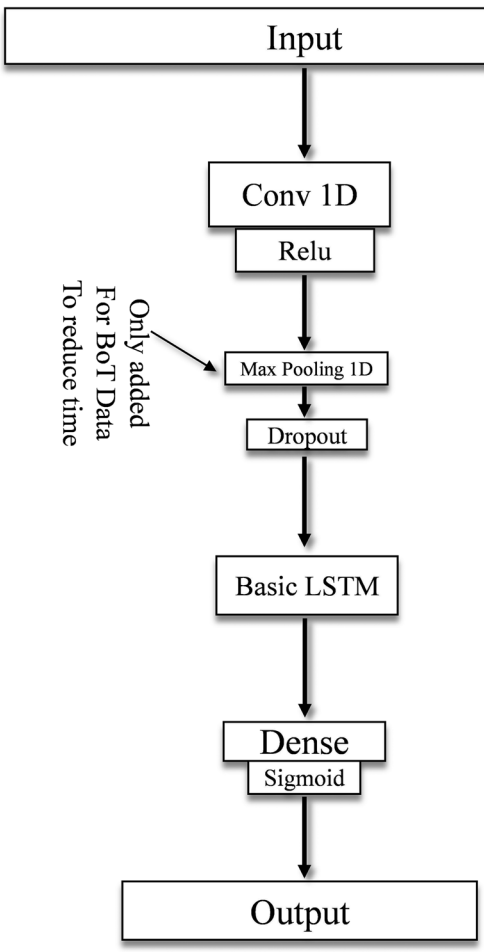

Fig. 1 Proposed CNN-LSTM architecture

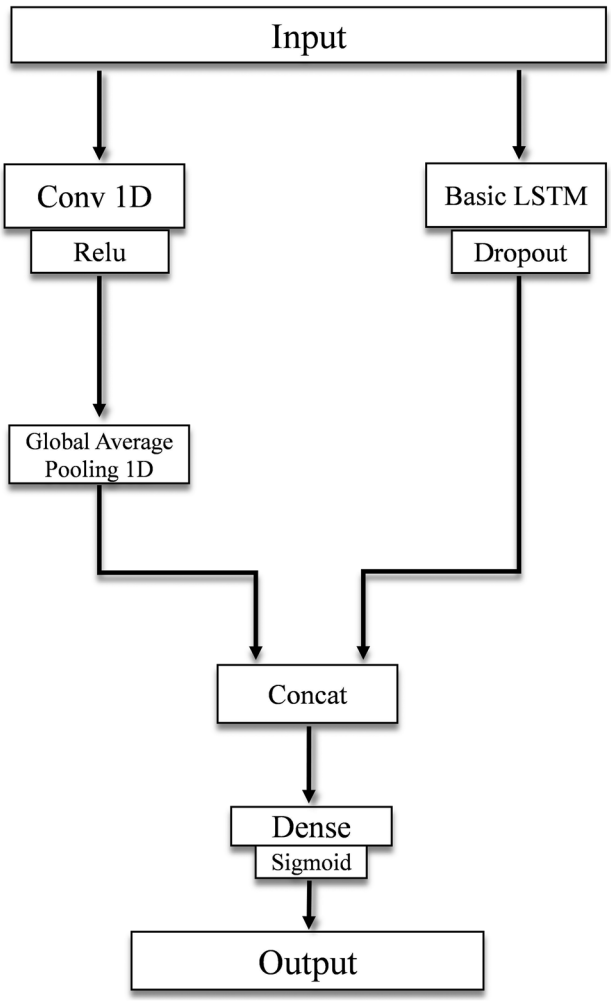

Fig. 2 Proposed LSTM-FCN architecture

batch normalization to normalize its input by applying a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. The hidden layer is used for feature extraction. An activation function is used in the hidden layers of a neural network to allow the model to learn more complex functions. In our architecture, we used rectified linear activation (ReLU) to enhance the results of the training. The ReLU is followed by then a MaxPooling1D layer whose job is to reduce the learning time by filtering the input (output of the previous layer) to the most salient new output. A dropout layer was introduced to avoid overfitting, a common issue in LSTM models. The introduced dropout layer had a probability value of 0.2 at which outputs of the layer are dropped out. The output of the dropout layer is then passed into the LSTM block. The LSTM block comprises a single hidden layer made up of 8 LSTM units, and an output layer used to make a prediction. The LSTM block is followed by a dense layer (a dense layer receives input from all neurons of the previous LSTM output layer) to produce one output value for the sigmoid activation function. The input values for the sigmoid function belong to the set of all real numbers, and its output values have a range of (0, 1) a binary outcome that represents (benign, attack). As part of the optimization of the algorithm, a binary cross-entropy loss function was used to estimate the loss of the proposed architecture on each iteration so that the weights can be updated to reduce the loss on the next iteration [63–68].

LSTM-FCN augments the fast classification performance of temporal convolutional layers with the precise classification of LSTM neural networks [69]. Temporal convolutions are an effective learning model for time series classification problems [35]. The proposed LSTM-FCN has a similar architecture to the proposed CNN-LSTM architecture, but instead, it utilizes a GlobalAveragePooling1D layer to retain much information about the "less important" outputs [65]. The layers then concatenate to one dense final layer with a Sigmoid activation function.

Both models have used Adam optimization algorithm [70] with a steady learning rate of 0.03 (the proportion of weights being updated throughout the 3 epochs of the proposed architecture). The 0.03 is a mid-range value that allows for steady learning. There was no need to optimize the hyperparameters (finding the optimal number of LSTM cells) due to the almost 0% misclassification rate of the proposed models. The default weight initializer that was used in the proposed architecture is GlorotUniform or Xavier Uniform. Since *k*-fold cross-validation (CV) is not commonly used in DL, here it is introduced on each model to investigate if it produces different results by preventing overfitting.

Table 5 Accuracy and loss values for all datasets

Dataset / Method	BoT-IoT		UNSW-NB15		TON-IoT	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
CNN-LSTM	99.99%	0.0016	99.99%	0.0001	98%	0.05
LSTM-FCN	100%	0.0068	100%	0.0054	90%	0.36
CNN-LSTM 5-folds CV	99.99%	0.0020	99.99%	0.0001	95%	0.25
LSTM-FCN 5-folds CV	100%	0.0015	100%	0.0002	85%	0.59

Table 6 Summary of AUROC values from different models

Model / Dataset	CNN-LSTM	LSTM-FCN	CNN-LSTM 5-folds CV					LSTM-FCN 5-folds CV				
			1	2	3	4	5	1	2	3	4	5
BoT-IoT	1.00	1.00	0.500	0.500	0.500	0.500	0.500	0.998	0.976	0.987	0.993	0.998
UNSW-NB15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TON-IoT	0.993	0.868	0.887	0.997	0.999	0.583	0.999	0.543	0.892	0.891	0.807	0.660

Moreover, the k value is chosen as 5, which is very common in ML [71, 72]. Finally, the models have utilized the StratifiedKFold [73] to ensure that each fold of the dataset has the same proportion of observations (balanced) with the response feature. When k-fold CV was not introduced, the train_test_split function from Scikit-learn [74] was utilized to split data into 80% for training and 20% for testing. A summary of the accuracy and loss results for all applied models for all datasets are listed in Table 5.

Accuracy describes just what percentage of test data are classified correctly. In any of these models, there is a binary classification of Attack or Benign. The accuracy of 99.99% means that out of 10,000 rows of data, the model can correctly classify 9999 rows. Table 5 shows that very high accuracy levels (≈ 99.99%) were achieved for the BoT-IoT and UNSW-NB15 datasets. However, this was not the case for the TON-IoT dataset, where accuracy levels ranged from 85–98%. It also reveals that using 5-folds CV has decreased the accuracy of the models used on the TON-IoT dataset. The proposed LSTM-FCN models have shown slightly better performance than the proposed CNN-LSTM models in detecting attacks using the BoT-IoT and the UNSW-NB15 datasets (100% vs. 99.99%) while the CNN-LSTM significantly performed better in detecting attacks compared to the LSTM-FCN using the TON-IoT dataset (98% vs. 90% and 95% vs. 85%).

The models use probabilities to predict binary class Attacks or Benign between 1 and 0. So if the probability of Attack is 0.6, then the probability of Benign is 0.4. In this case, the outcome is classified as an Attack. The loss will be the sum of the difference between the predicted probability of the real class of the test outcome and 1. Table 5 shows that very low loss values were achieved for the BoT-IoT and UNSW-NB15 datasets. At the same time, using 5-folds CV

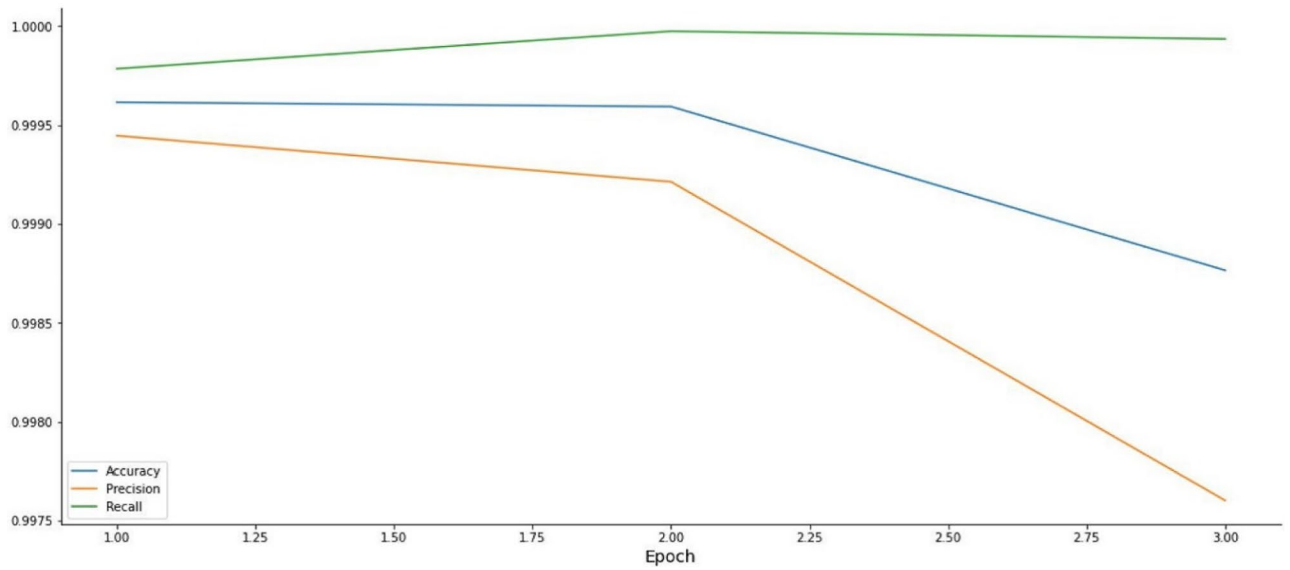

Fig. 3 CNN-LSTM evaluation metrics results for BoT-IoT dataset

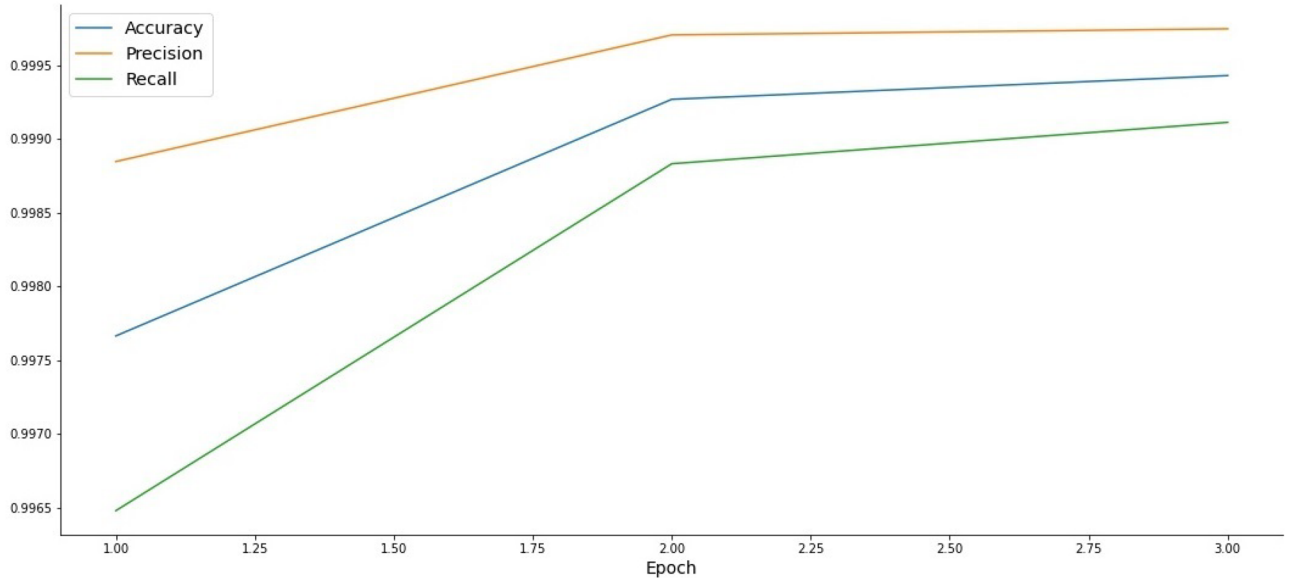

Fig. 4 LSTM-FCN evaluation metrics results for BoT-IoT dataset

reduced the loss values for the FCN-LSTM from 0.0068 to 0.0015 and 0.0054 to 0.0002 for the BoT-IoT and UNSW-NB15 datasets, respectively. However, this was not the case for the TON-IoT dataset where loss values increased when 5-folds CV was implemented.

The area under the receiver operating characteristics (AUROC) is a performance measurement for classification models. The AUROC tells us what the model probability of separating between different classes, Attack or Benign in this case, is. The AUROC is a probability that measures the performance of a binary classifier averaged across all possible decision thresholds. When the AUROC value is 1, it indicates that the model has an ideal capacity to distinguish between Attack or Benign. When the AUROC value is 0, it indicates that the model is reciprocating the classes. In other words, predicting a Benign class as an Attack class and vice versa. Moreover, when the AUROC value is 0.5, it indicates that the model cannot distinguish between Attack or Benign. Table 6 summarizes AUROC values for all proposed models on the three datasets.

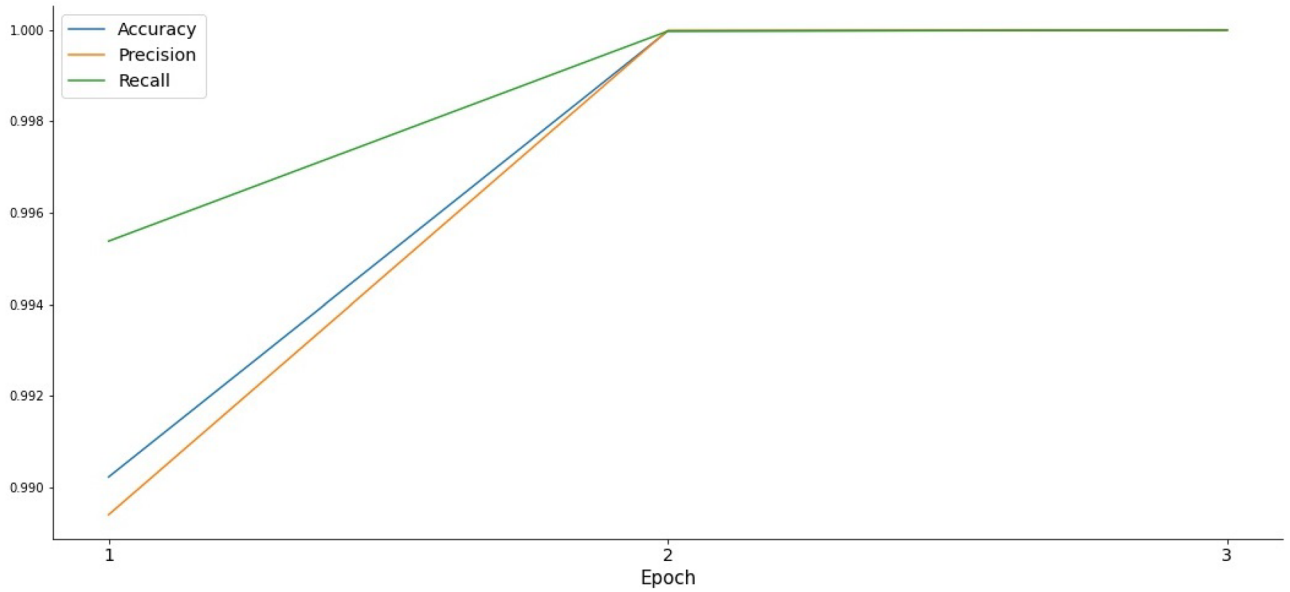

Fig. 5 CNN-LSTM evaluation metrics results for UNSW-NB15 dataset

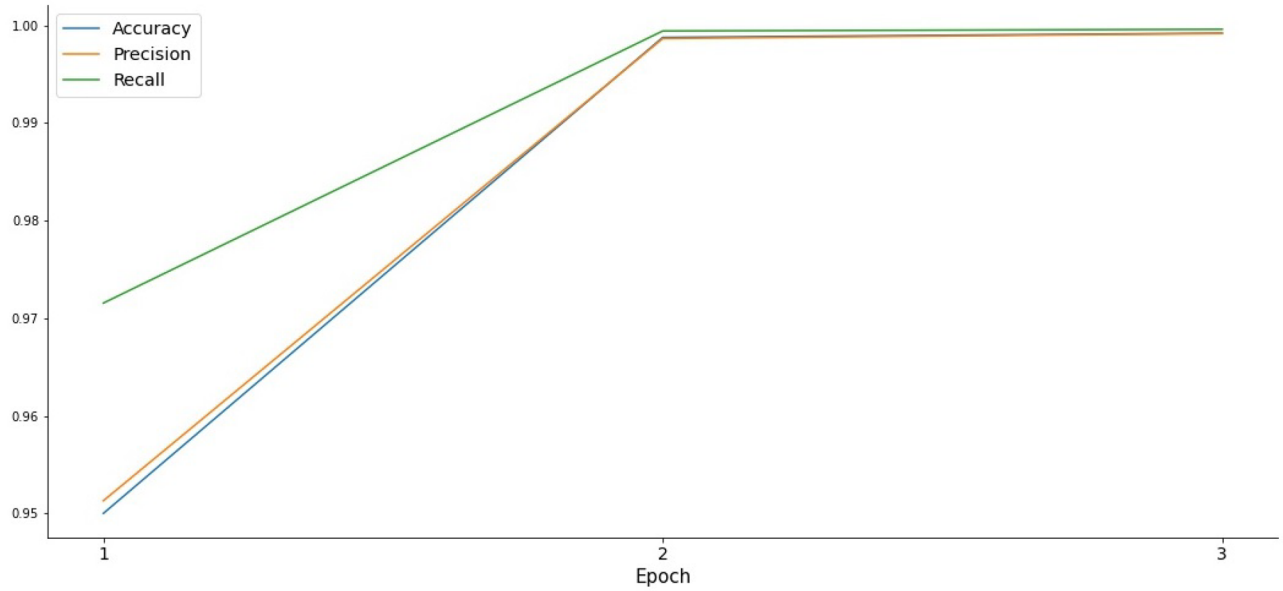

Fig. 6 LSTM-FCN evaluation metrics results for UNSW-NB15 dataset

All models showed ideal capacity (AUROC = 1.00) for predicting Attack or Benign classes for the UNSW-NB15 dataset. As for the BoT-IoT dataset, the CNN-LSTM and LSTM-FCN models showed high capacity (AUROC = 1.00) for predicting Attack or Benign classes. The CNN-LSTM 5-folds CV had an AUROC = 0.5, indicating that this model can be incapable of distinguishing between Attack or Benign. On the other hand, the LSTM-FCN 5-folds CV had an AUROC value larger than 0.992, which means that this model can almost predict Attack or Benign classes.

The TON-IoT dataset showed a slightly different case than the first two datasets, with AUROC values that were 0.993 and 0.868 for the CNN-LSTM and LSTM-FCN models, respectively, indicating that they are near capable of predicting Attack or Benign classes. The 5-folds CV for both CNN-LSTM and LSTM-FCN models showed AUROC values that ranged between 0.543 and 0.999. Figures 3, 4, 5, 6, 7, and 8 demonstrate accuracy, precision, and recall results by CNN-LSTM and LSTM-FCN models for all datasets.

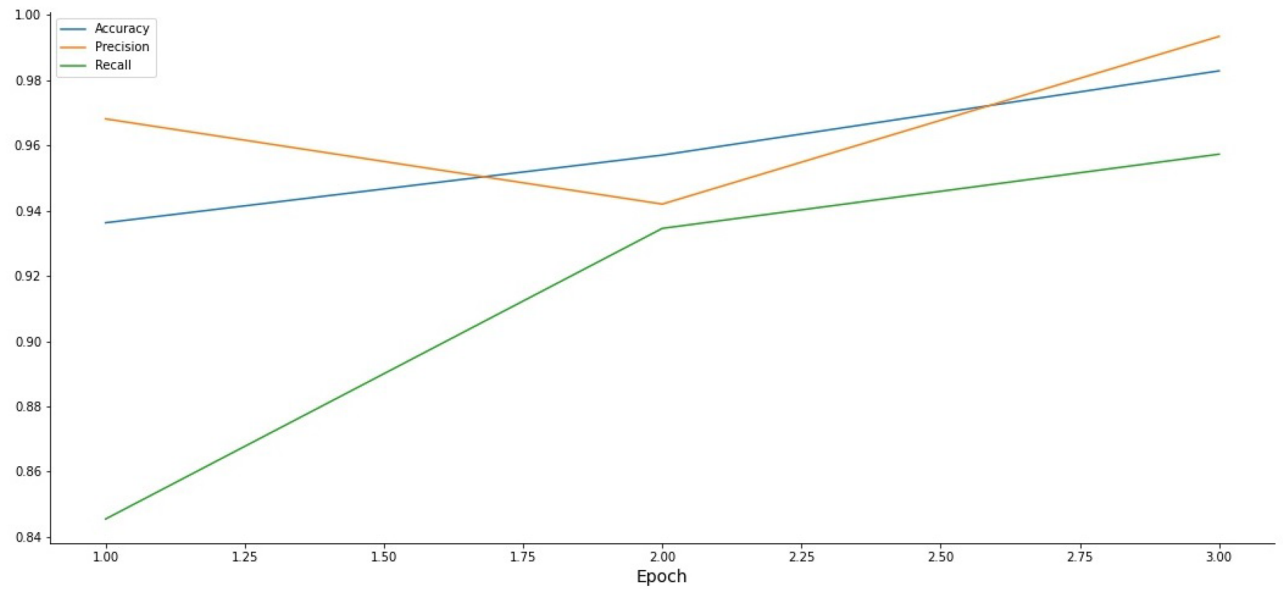

Fig. 7 CNN-LSTM evaluation metrics results for TON-IoT dataset

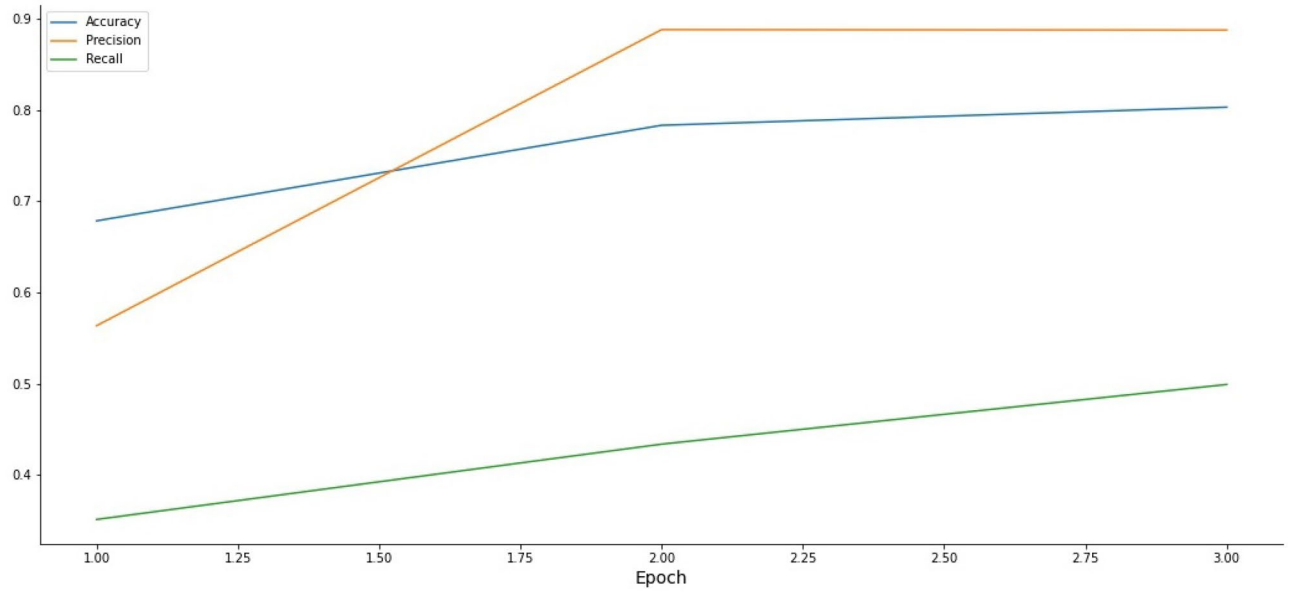

Fig. 8 LSTM-FCN evaluation metrics results for TON-IoT dataset

In addition to accuracy and as seen in Figs. 3, 4, 5, 6, 7, and 8, the values of precision and recall were used to choose the optimal number of epochs used to train our models. A hyperparameter defines the number of opportunities that the algorithm will work through the entire training dataset to learn and update the internal model parameters. This allows for choosing the optimal settings for the model parameters to be used later in the testing dataset to achieve the highest accuracy, precision, and recall values.

Precision is defined as the accuracy of positive predictions (mathematically defined as true positives/true positives + false positives), which is the ability of our model not to label an instance positive (for example, an attach) that is negative (for example, benign). In other words, our detection model can identify only the relevant data points. On the other hand, recall is a measure of the model's ability to detect all positive instances correctly. High recall (mathematically defined as true positives/true positives + false negatives) values indicate high model's sensitivity on reducing the number of false negatives. Table 7 summarizes the optimal number of epochs chosen for each model to yield the highest accuracy, precision, and recall values.

Table 7 Number of epochs for CNN-LSTM vs. LSTM-FCN models

CNN-LSTM			LSTM-FCN		
BoT-IoT	UNSW-NB15	TON-IoT	BoT-IoT	UNSW-NB15	TON-IoT
2	3	3	3	3	3

5 Conclusions

In the current paper, novel deep learning models for attack classification and detection were proposed utilizing Industrial IoT datasets (BoT-IoT, UNSW-NB15, and TON-IoT). The results have shown a state-of-the-art performance in identifying, classifying, and detecting cybersecurity threats. The evaluation process has employed accuracy and AUROC values as performance metrics to show the effectiveness of the proposed models on the three benchmark datasets. The results have shown that deep learning algorithms are capable of accurately detecting and classifying the attacks in more than 99.9% percent of the instances in two of the three datasets employed. Future researchers can explore the usage of attention mechanisms to improve time series classification with the Attention LSTM block. The future endeavor can also focus on studying whether having a similar or different set of features across various datasets can affect the performance of the NIDS via DL algorithms.

Author contribution All authors contributed to this paper's conception and design. Material preparation, data collection, and analysis were performed by Mohammad Shahin, Hamed Bouzarya, and Ali Hosseinzadeha. The first draft of the manuscript was written by Mohammad Shahin and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Declarations

Competing interests The authors declare no competing interests.

References

1. Zheng Y, Pal A, Abuadbba S, Pokhrel SR, Nepal S, Janicke H (2020) Towards IoT security automation and orchestration, 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), TPS-ISA 55–63. https://doi.org/10.1109/TPS-ISA50397.2020.00018
2. Shahin M, Chen FF, Bouzary H, Krishnaiyer K (2020) Integration of Lean practices and Industry 4.0 technologies: smart manufacturing for next-generation enterprises. Int J Adv Manuf Technol 107(5):2927–2936. https://doi.org/10.1007/s00170-020-05124-0
3. Baumann D, Mager F, Wetzker U, Thiele L, Zimmerling M, Trimpe S (2021) Wireless control for smart manufacturing: recent approaches and open challenges. Proc IEEE 109(4):441–467. https://doi.org/10.1109/JPROC.2020.3032633
4. Donnal J, McDowell R, Kutzer M (2020) Decentralized IoT with Wattsworth. 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), Internet of Things (WF-IoT), 2020 IEEE 6th World Forum on 1–6. https://doi.org/10.1109/WF-IoT48130.2020.9221350
5. Sungwon LEE, Hyeonkyu JEON, Gihyun PARK, Jonghee YOUN (2021) Design of automation environment for analyzing various IoT malware. Tehnicki vjesnik / Technical Gazette 28(4):827–835. https://doi.org/10.17559/TV-20210202131602
6. Elhabashy AE, Wells LJ, Camelio JA (2019) Cyber-physical security research efforts in manufacturing - a literature review. in Procedia Manuf 01 34:921–931 https://doi.org/10.1016/j.promfg.2019.06.115
7. Elhabashy AE, Wells LJ, Camelio JA, Woodall WH (2019) A cyber-physical attack taxonomy for production systems: a quality control perspective. J Intell Manuf 30(6):2489–2504. https://doi.org/10.1007/s10845-018-1408-9
8. ICS Monitor Newsletters | CISA. https://www.us-cert.gov/ics/monitors Accessed 20 Oct 2019
9. O'Reilly P, Rigopoulos K, Feldman L, Witte G (2021) 2020 Cybersecurity and Privacy Annual Report. Natl Inst Stand Technol. https://doi.org/10.6028/NIST.SP.800-214
10. Shahin M, Chen FF, Bouzary H, Zarreh A (2020) Frameworks proposed to address the threat of cyber-physical attacks to Lean 4.0 systems. Procedia Manuf 51:1184–1191. https://doi.org/10.1016/j.promfg.2020.10.166
11. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B (2019) Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. Futur Gener Comput Syst 100:779–796. https://doi.org/10.1016/j.future.2019.05.041
12. Mahmood T, Afzal U (2013) Security analytics: big data analytics for cybersecurity: a review of trends, techniques and tools. in 2013 2nd National Conference on Information Assurance (NCIA) 129–134. https://doi.org/10.1109/NCIA.2013.6725337
13. Terzi DS, Terzi R, Sagiroglu S (2017) Big data analytics for network anomaly detection from netflow data. in 2017 International Conference on Computer Science and Engineering (UBMK) 592–597. https://doi.org/10.1109/UBMK.2017.8093473
14. Gaggero GB, Rossi M, Girdinio P, Marchese M (2019) Neural network architecture to detect system faults / cyberattacks anomalies within a photovoltaic system connected to the grid. in 2019 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)1–4. https://doi.org/10.1109/ISAECT47714.2019.9069683
15. Bruce PC, Shmueli G, Patel NR (2016) Data mining for business analytics: concepts, techniques, and applications in Microsoft Office Excel with XLMiner. Wiley-Blackwell
16. Ciaburro G (2017) Neural networks with R. Packt Publishing. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b5582708&site=eds-live&scope=site Accessed 18 Oct 2021
17. Pascanu R, Stokes JW, Sanossian H, Marinescu M, Thomas A (2015) Malware classification with recurrent networks. in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 1916–1920. https://doi.org/10.1109/ICASSP.2015.7178304
18. Shibahara T, Yagi T, Akiyama M, Chiba D, Yada T (2016) Efficient dynamic malware analysis based on network behavior using deep learning. in 2016 IEEE Global Communications Conference (GLOBECOM) 1–7. https://doi.org/10.1109/GLOCOM.2016.7841778
19. Loukas G, Vuong T, Heartfield R, Sakellari G, Yoon Y, Gan D (2018) Cloud-based cyber-physical intrusion detection for vehicles using deep learning. IEEE Access 6:3491–3508. https://doi.org/10.1109/ACCESS.2017.2782159
20. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
21. Kim J, Kim J, Thu HLT, Kim H (2016) Long short term memory recurrent neural network classifier for intrusion detection. in 2016 Int Conf on Platform Technol and Service (PlatCon) 1–5. https://doi.org/10.1109/PlatCon.2016.7456805
22. McDermott CD, Majdani F, Petrovski AV (2018) Botnet detection in the Internet of Things using deep learning approaches. in 2018 Int Jt Conf Neural Netw (IJCNN) 1–8. https://doi.org/10.1109/IJCNN.2018.8489489
23. Chatterjee CC (2019) Implementation of RNN, LSTM, and GRU. Medium. https://towardsdatascience.com/implementation-of-rnn-lstm-and-gru-a4250bf6c090 Accessed 10 Dec 2021
24. Zhao Q, Zhu Y, Wan D, Yu Y, Cheng X (2018) Research on the data-driven quality control method of hydrological time series data. Water (Switzerland) 10(12):23 https://doi.org/10.3390/w10121712
25. Yasrab R, Pound M (2020) PhenomNet: bridging phenotype-genotype gap: a CNN-LSTM based automatic plant root anatomizationsystem. https://doi.org/10.1101/2020.05.03.075184
26. Kim J, Kim J, Kim H, Shim M, Choi E (2020) CNN-based network intrusion detection against denial-of-service attacks. Electronics 9(916):916. https://doi.org/10.3390/electronics9060916
27. Wang Wei, Zhu Ming, Zeng Xuewen, Ye Xiaozhou, Sheng Yiqiang (2017) Malware traffic classification using convolutional neural network for representation learning 712–717. https://doi.org/10.1109/ICOIN.2017.7899588
28. McLaughlin N (2017) Deep android malware detection. in Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, Scottsdale, Arizona, USA. 301–308. https://doi.org/10.1145/3029806.3029823
29. Gibert D, Mateu C, Planes J, Vicens R (2019) Using convolutional neural networks for classification of malware represented as images. J Comput Virol Hacking Tech 15(1):15–28. https://doi.org/10.1007/s11416-018-0323-0
30. Yu Y, Long J, Cai Z (2017) Network intrusion detection through stacking dilated convolutional autoencoders. Secur Commun Netw. https://www.hindawi.com/journals/scn/2017/4184196/ Accessed 20 Jun 2020
31. Kolosnjaji B, Zarras A, Webster G, Eckert C (2016) Deep learning for classification of malware system call sequences, in AI 2016: Advances in Artificial Intelligence. Cham 2016:137–149. https://doi.org/10.1007/978-3-319-50127-7_11
32. Mac H, Tran D, Tong V, Nguyen G, Tran HA (2017) DGA Botnet detection using supervised learning methods. 211–218. https://doi.org/10.1145/3155133.3155166
33. Yu B, Gray DL, Pan J, Cock MD, Nascimento ACA (2017) Inline DGA detection with deep networks. in 2017 IEEE International Conference on Data Mining Workshops (ICDMW) 683–692. https://doi.org/10.1109/ICDMW.2017.96
34. Karim F, Majumdar S, Darabi H (2019) Insights into LSTM fully convolutional networks for time series classification. IEEE Access 7:67718–67725. https://doi.org/10.1109/ACCESS.2019.2916828

35. Wang Zhiguang, Yan Weizhong, Oates T (2017) Time series classification from scratch with deep neural networks: a strong baseline. 2017 International Joint Conference on Neural Networks (IJCNN). Neural Networks (IJCNN) 1578–1585. https://doi.org/10.1109/IJCNN.2017.7966039
36. Park E, Cui X, Nguyen THB, Kim H (2019) Presentation attack detection using a tiny fully convolutional network, IEEE transactions on information forensics and security, information forensics and security, IEEE transactions on. IEEE Trans Inform Forensic Secur 14(11):3016–3025. https://doi.org/10.1109/TIFS.2019.2907184
37. Sarhan M, Layeghy S, Moustafa N, Portmann M (2021) NetFlow datasets for machine learning-based network intrusion detection systems. arXiv:2011.09144 [cs]. 371:117–135. https://doi.org/10.1007/978-3-030-72802-1_9
38. Peterson JM, Leevy JL, Khoshgoftaar TM (2021) A review and analysis of the Bot-IoT dataset. 2021 IEEE International Conference on Service-Oriented System Engineering (SOSE). Service-Oriented System Engineering (SOSE) SOSE 20–27. https://doi.org/10.1109/SOSE52839.2021.00007
39. Koroniotis N, Moustafa N, Sitnikova E, Slay J (2018) Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques. in Mobile Networks and Management, Cham 30–44. https://doi.org/10.1007/978-3-319-90775-8_3
40. Koroniotis N, Moustafa N, Sitnikova E (2020) A new network forensic framework based on deep learning for Internet of Things networks: a particle deep framework. Futur Gener Comput Syst 110:91–106. https://doi.org/10.1016/j.future.2020.03.042
41. Koroniotis N, Moustafa N (2020) Enhancing network forensics with particle swarm and deep learning: the particle deep framework 60. https://doi.org/10.5121/csit.2020.100304
42. Koroniotis N, Moustafa N, Schiliro F, Gauravaram P, Janicke H (2020) A holistic review of cybersecurity and reliability perspectives in smart airports. IEEE Access 8:209802–209834. https://doi.org/10.1109/ACCESS.2020.3036728
43. Cox J, Singh A (2018) Practical network scanning : capture network vulnerabilities using standard tools such as Nmap and Nessus. Packt Publishing. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b5447291&site=eds-live&scope=site Accessed 21 Oct 2021
44. Tankard C (2011) Advanced persistent threats and how to monitor and deter them. Network Security8:16–19. https://doi.org/10.1016/S1353-4858(11)70086-1
45. A survey on authentication attacks and countermeasures in a distributed environment | Semantic Scholar. [Online]. Available: https://www.semanticscholar.org/paper/A-SURVEY-ON-AUTHENTICATION-ATTACKS-AND-IN-A-Jesudoss/4a6383ce27766f892cebb0269d7be20260023cec Accessed 21 Oct 2021
46. Fernández A, García S, Galar M, Prati RC, Krawczyk B, Herrera F (2018) Learning from imbalanced data sets. Springer. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b4768180&site=eds-live&scope=site Accessed 10 Dec 2021
47. Handling imbalanced data- machine learning, computer vision, NLP, Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/11/handling-imbalanced-data-machine-learning-computer-vision-and-nlp/ Accessed 10 Dec 2021
48. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b1535649&site=eds-live&scope=site Accessed 11 Dec 2021
49. Zheng A, Casari A (2018) Feature engineering for machine learning : principles and techniques for data scientists, First edition. O'Reilly Media. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b5167004&site=eds-live&scope=site Accessed 11 Dec 2021.
50. Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), presented at the 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings 07. https://doi.org/10.1109/MilCIS.2015.7348942
51. Moustafa N, Slay J (2016) The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf Syst Secur 25(1–3):18–31
52. Moustafa N, Slay J, Creech G (2019) Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks, IEEE transactions on big data, big data, IEEE transactions on. IEEE Trans Big Data 5(4):481–494. https://doi.org/10.1109/TBDATA.2017.2715166
53. Moustafa N, Creech G, Slay J (2017) Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models, in Data analytics and decision support for cybersecurity: trends, methodologies and applications. Palomares I, Carrascosa, Kalutarage HK, Huang Y, Eds. Cham: Springer International Publishing. 127–156. https://doi.org/10.1007/978-3-319-59439-2_5
54. Witten IH, Frank E, Hall MA, Pal CJ (2017) Data mining : practical machine learning tools and techniques, Fourth edition. Morgan Kaufmann. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b5158398&site=eds-live&scope=site Accessed 11 Dec 2021.
55. Moustafa N (2021) A new distributed architecture for evaluating AI-based security systems at the edge: network TON_IoT datasets. Sustain Cities Soc 72:102994. https://doi.org/10.1016/j.scs.2021.102994
56. Booij TM. Chiscop I, Meeuwissen E, Moustafa N, den Hartog FTH (2021) ToN_IoT: the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion datasets. IEEE Internet of Things Journal 1–1. https://doi.org/10.1109/JIOT.2021.3085194
57. Alsaedi A, Moustafa N, Tari Z, Mahmood A, Anwar A (2020) TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access 8:165130–165150. https://doi.org/10.1109/ACCESS.2020.3022862
58. Moustafa N, Keshky M, Debiez E, Janicke H (2020) Federated TON_IoT windows datasets for evaluating AI-based security applications. in 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) 848–855. https://doi.org/10.1109/TrustCom50675.2020.00114
59. Moustafa N, Ahmed M, Ahmed S (2020) Data analytics-enabled intrusion detection: evaluations of ToN_IoT linux datasets. in 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) 727–735. https://doi.org/10.1109/TrustCom50675.2020.00100
60. Moustafa N (2020) New generations of internet of things datasets for cybersecurity applications based machine learning: TON_IoT datasets. Research Data Australia. https://researchdata.edu.au/new-generations-internet-toniot-datasets/1425941 Accessed 11 Dec 2021
61. Moustafa N (2019) A systemic IoT-fog-cloud architecture for big-data analytics and cyber security systems: a review of fog computing. [cs]. [Online]. Available:https://arxiv.org/abs/1906.01055 Accessed 11 Dec 2021
62. Ashraf J et al (2021) IoTBoT-IDS: a novel statistical learning-enabled botnet detection framework for protecting networks of

smart cities. Sustain Cities Soc 72:103041. https://doi.org/10.1016/j.scs.2021.103041
63. Livieris IE, Pintelas E, Pintelas P (2020) A CNN–LSTM model for gold price time-series forecasting. Neural Comput & Applic 32(23):17351–17360. https://doi.org/10.1007/s00521-020-04867-x
64. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
65. Chollet F (2018) Deep learning with Python. Manning Publications. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b5162307&site=eds-live&scope=site Accessed 12 Dec 2021
66. Mahmoudi MA, Chetouani A, Boufera F, Tabia H (2020) Kernelized dense layers for facial expression recognition. 2020 IEEE International Conference on Image Processing (ICIP), Image Processing (ICIP), 2020 IEEE International Conference on 2226–2230. https://doi.org/10.1109/ICIP40778.2020.9190694
67. Chiluveru SR, Gyanendra, Chunarkar S, Tripathy M, Kaushik BK (2021) Efficient hardware implementation of DNN-based speech enhancement algorithm with precise sigmoid activation function. IEEE transactions on circuits and systems II: express briefs, circuits and systems II: express briefs, IEEE transactions on, IEEE Trans Circuits Syst II 68(11):3461–3465. https://doi.org/10.1109/TCSII.2021.3082941
68. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. in 32nd International Conference on Machine Learning, ICML1:448–456. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-84969584486&site=eds-live&scope=site Accessed 13 Dec 2021
69. Karim F, Majumdar S, Darabi H, Chen S (2018) LSTM fully convolutional networks for time series classification. IEEE Access 6:1662–1669. https://doi.org/10.1109/ACCESS.2017.2779939
70. Kingma DP, Ba J (2017) Adam: a method for stochastic optimization, arXiv:1412.6980[cs]. [Online]. Available: https://arxiv.org/abs/1412.6980 Accessed 13 Dec 2021
71. Kuhn M, Johnson K (2013) Applied predictive modeling. Springer. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b2605857&site=eds-live&scope=site Accessed 13 Dec 2021
72. Ethem Alpaydin (2014) Introduction to machine learning. vol. Third edition. Cambridge, MA: The MIT Press. [Online]. Available: https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=836612&site=eds-live&scope=site Accessed 13 Dec 2021
73. Adagbasa EG, Adelabu SA, Okello TW (2019) Application of deep learning with stratified K-fold for vegetation species discrimination in a protected mountainous region using Sentinel-2 image. Geocarto International 01. https://doi.org/10.1080/10106049.2019.1704070
74. Scikit-learn: machine learning in Python — scikit-learn 1.0.2 documentation. https://scikit-learn.org/stable/index.html Accessed 08 Jan 2022