Implementation of a Novel Fully Convolutional Network Approach to Detect and Classify Cyber-Attacks on IoT Devices in Smart Manufacturing Systems

Mohammad Shahin, FFrank Chen(✉), Hamed Bouzary, Ali Hosseinzadeh, and Rasoul Rashidifar

The University of Texas at San Antonio, San Antonio, TX 78249, USA
FF.Chen@utsa.edu

Abstract. In recent years, Internet of things (IoT) devices have been widely implemented and industrially improved in manufacturing settings to monitor, collect, analyze, and deliver data. Nevertheless, this evolution has increased the risk of cyberattacks, significantly. Consequently, developing effective intrusion detection systems based on deep learning algorithms has proven to become a dependable intelligence tool to protect Industrial IoT devices against cyber-attacks. In the current study, for the first time, two different classifications and detection long short-term memory (LSTM) architectures were fine-tuned and implemented to investigate cyber-security enhancement on a benchmark Industrial IoT dataset (BoT-IoT) which takes advantage of several deep learning algorithms. Furthermore, the combinations of LSTM with FCN and CNN demonstrated how these two models can be used to accurately detect cyber security threats. A detailed analysis of the performance of the proposed models is provided. Augmenting the LSTM with FCN achieves state-of-the-art performance in detecting cybersecurity threats.

Keywords: Smart manufacturing · Industrial IoT · Machine learning · Cybersecurity

1 Introduction

In the last two decades, there has been growing interest in smart Internet of things (IoT) devices in many applications of Industry 4.0 [1] such as smart manufacturing due to increasing the integration of cyber-physical systems (CPS) into the internet [2]. Generally, large-scale CPS networks made smart manufacturing systems that are safety-critical and rely on networked and distributed control architectures [3]. Recently, with decreasing cost of sensors and superior access to high bandwidth wireless networks, the usage of IoT devices in manufacturing systems has increased significantly [4]. Nevertheless, the implementation of IoT devices into manufacturing systems increases the risk of cyber-attacks. Therefore, the security of IoT systems has become a vital concern to businesses.

© The Author(s) 2023

According to the report of Industrial Control Systems Monito Newsletter, approximately one-third of the cyber-attacks target the manufacturing sector [5]. Furthermore, based on the National Institute of Standards and Technology (NIST), these attacks via cyberspace, target an enterprise's use of cyberspace to destroy, or maliciously control a computing infrastructure [6].

Realistic security and investigation countermeasures, such as network intrusion detection and network forensic systems, must be designed effectively to face the rising threats and challenges of cyber-attacks [7]. Today, data analytics is at the forefront of the war against cyber-attacks. Cybersecurity experts have been employing data analytics not only to improve the cybersecurity monitoring levels over their network streams but also to increase real-time detection of threat patterns [8, 9].

Neural Networks (NN) were inspired by the way the human brain works. NN algorithms are well-suited for usage in a variety of Artificial Intelligence (AI) and (Machine Learning) ML applications because they are made up of several data layers. Recurrent Neural Networks (RNNs) transmit data back and forth from later processing stages to earlier stages (networks with cyclic data flows that may be employed in natural language processing and speech recognition) [10]. RNN was used to achieve a true positive rate of 98.3% at a false positive rate of 0.1% in detecting malware [11]. In another paper, Shibahara et al. [12] utilized RNN to detect malware based on network behavior with high precision. Also, 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) [13]. An RNN-LSTM approach has been used in intrusion detection systems to detect botnet activity within consumer IoT devices and networks [14].

LSTM [13] refers to neural networks capable of learning order dependency in sequence prediction and remembering a large amount of prior information via Back Propagation (BP) or previous neuron outputs and incorporating them into present processing. LSTM can be leveraged with various other architectures of NN. The most notable application for such network builds is seen in text prediction, machine translation, speech recognition, and more [10]. By replacing the hidden layer nodes that act on memory cells through the Sigmoid function, LSTM proposes an enhancement to the RNN model. These memory cells are in charge of exchanging information by storing, recording, and updating previous information [15].

Convolutional Neural Network (CNN) uses a feed-forward topology to propagate signals, CNN is more often used in classification and computer vision recognition tasks [10]. In a unique study, Yu et al. [16] suggested a neural network architecture that combines CNN with autoencoders to evaluate network intrusion, detection models. Also, Kolosnjaji et al. [17] 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%. In conclusion, CNN has a Deep Learning (DL) network architecture that learns directly from data without the necessity of manual feature extraction.

Fully Convolutional Neural Network (FCN) is a CNN without fully connected layers [18]. A major advantage of using FCN models is that it does not require heavy preprocessing or feature engineering since the FCN neuron layers are not dense (fully connected) [19]. FCN has been used [20] 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.

Although progress has been made to solve and decrease the risk of cyber-attacks with different machine learning models and algorithms, it is necessary to implement novel and efficient methods to keep protections updated. In this study, 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 a Bot-IoT dataset. 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. This paper shows how DL algorithms can be vital in detecting cybersecurity threats by proposing novel algorithms and evaluating their efficiency and fidelity on a new dataset. The next three sections discuss the preprocessing methodology of the dataset, the results and analysis of this paper, and the conclusion.

2 Preprocessing of Datasets

Network Intrusion Detection Systems (NIDS) based on DL algorithms have proven to be a reliable network protection tool against cyber-attacks [14]. In this paper, we applied state-of-the-art DL algorithms on a benchmark NIDS dataset known as BoT-IoT [11]. This dataset was released by The Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) in 2018.

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 randomly extracted using MySQL queries. The extracted 5%, is comprised of 4 files of approximately 1.07 GB total size, and about 3 million records, [21]. The dataset includes a range of attack categories such as Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), Operating System Scan (OS Scan) also known as Reconnaissance or Prope, Keylogging (Theft), Data Exfiltration (Theft), Benign (No attack).

This dataset 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, feature selection was employed when the algorithm itself extracts the important features. Furthermore, an upsampling technique [46] was used to overcome the heavily imbalanced binary response feature. The feature contained only 13859 minority counts of benign compared to a whapping 586241 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 [22]. 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 [23].

3 Results and Analysis

To create our four main models on the dataset, two basic architectures were proposed: CNN-LSTM. The suggested CNN-LSTM architecture employs a one-dimensional convolutional hidden layer with three filters (collection of kernels used to store values learned during the training phase) and a kernel size of 32 that operates over a 1D sequence. Batch normalization is used in conjunction with the convolutional hidden layer to normalize its input by implementing a transformation that keeps the mean output near 0 and the output standard deviation close to 1. The hidden layer is used to extract features. In the hidden layers of a neural network, an activation function is employed to allow the model to learn increasingly complicated functions. Rectified Linear Activation (ReLU) was utilized in our design to improve the training performance. The ReLU is then followed by a MaxPooling1D layer, to minimize the learning time by filtering the input (prior layer's output) to the most important new output. A dropout layer was included to prevent overfitting, which is a typical problem with LSTM models. The added dropout layer has a probability of 0.2, which means that the layer's outputs are dropped out. The dropout layer's output is subsequently sent into the LSTM block. A single hidden layer made up of 8 LSTM units and an output layer are used to create the LSTM block. After the LSTM block, a Dense layer (which gets input from all neurons in the preceding LSTM output layer) produces one output value for the sigmoid activation function. The sigmoid function's input values are all real integers, and its output values are in the range of (0, 1), a binary result that reflects (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 [24, 25].

LSTM-FCN combines the exact classification of LSTM Neural Networks with the quick classification performance of temporal convolutional layers [26]. For time series classification tasks, temporal convolutions have proven to be an effective learning model [19]. 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 [27]. The layers are then concatenated into a single Dense final layer with Sigmoid activation.

Both models have utilized Adam Optimization Algorithm [28] with a steady learning rate of 0.03 (the proportion that weights are 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 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. Moreover, the k value is chosen as 5 which is very common in the field of ML [29, 30]. The models have utilized the StratifiedKFold [31] to ensure that each fold of the dataset has the same proportion of observations (balanced) with the response feature. In the case where k-fold CV was not introduced, the train_test_split function from Scikit-learn [32] was utilized to split data into 80% for training and 20% for testing. A summary of the accuracy and loss results for the applied models is listed in Table 1.

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. When accuracy is 99.99%, it means that out of 10000 rows of data, the model can correctly classify 9999 rows. Table 2 shows that very high accuracy levels (−99.99%) were achieved for the BoT-IoT datasets. The proposed LSTM-FCN models have shown slightly better performance than the proposed CNN-LSTM models in detecting attacks using the BoT-IoT dataset (100% vs 99.99%).

Table 1. Accuracy and Loss values for different methods.

Methods	Accuracy	Loss
CNN-LSTM	99.99%	0.0016
LSTM-FCN	100%	0.0068
CNN-LSTM 5-folds CV	99.99%	0.0020
LSTM-FCN 5-folds CV	100%	0.0015

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 2 shows that very low loss values were achieved for the BoT-IoT dataset. At the same time, using 5-folds CV reduced the loss values for the FCN-LSTM from 0.0068 to 0.0015.

The Area Under the Receiver Operating Characteristics (AUROC) is a performance measurement for classification models. The AUROC reveals the model probability of separating between various classes, Attack or Benign in this case. The AUROC is a probability that measures the performance of a binary classifier averaged across all possible decision thresholds. When 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. Table 2 shows a summary of AUROC values for all proposed models. The CNN-LSTM and LSTM-FCN models showed high capacity (AUROC = 1.00) of predicting Attack or Benign classes.

Table 2. Summary of AUROC values from different models.

CNN-LSTM	LSTM-FCN	CNN-LSTM 5-folds CV					LSTM-FCN 5-folds CV				
		1	2	3	4	5	1	2	3	4	5
1.00	1.00	0.500	0.500	0.500	0.500	0.500	0.998	0.976	0.987	0.993	0.998

4 Conclusions

In this paper, novel deep learning models for attack classification and detection were proposed utilizing the Industrial IoT dataset (BoT-IoT). The results revealed cutting-edge performance in terms of detecting, classifying, and identifying cybersecurity threats. The evaluation process has utilized accuracy and AUROC values as performance metrics to show the effectiveness of the proposed models on the three benchmark datasets. Deep learning algorithms were shown to be capable of successfully identifying and categorizing assaults in more than 99.9% of cases in two of the three datasets used. With the Attention LSTM block, future researchers may investigate the use of attention processes to enhance time series classification. Future research might look at whether having a similar or distinct collection of characteristics across different datasets affects the NIDS' performance using DL methods.

References

1. Shahin, M., Chen, F.F., Bouzary, H., Krishnaiyer, K.: Integration of lean practices and Industry 4.0 technologies: smart manufacturing for next-generation enterprises. Int. J. Adv. Manufact. Technol. **107**(5–6), 2927–2936 (2020). https://doi.org/10.1007/s00170-020-05124-0
2. Zheng, Y., Pal, A., Abuadbba, S., Pokhrel, S.R., Nepal, S., Janicke, H.: Towards IoT security automation and orchestration. In: 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), 2020 Second IEEE International Conference on, TPS-ISA, pp. 55–63 (2020). https://doi.org/10.1109/TPS-ISA50397.2020.00018
3. Baumann, D., Mager, F., Wetzker, U., Thiele, L., Zimmerling, M., Trimpe, S.: Wire-less control for smart manufacturing: recent approaches and open challenges. Proc. IEEE **109**(4), 441–467 (2021). https://doi.org/10.1109/JPROC.2020.3032633
4. Donnal, J., McDowell, R., Kutzer, M.: Decentralized IoT with wattsworth. In: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), Internet of Things (WF-IoT), 2020 IEEE 6th World Forum on, pp. 1–6, (2020). https://doi.org/10.1109/WF-IoT48130.2020.9221350
5. Elhabashy, A.E., Wells, L.J., Camelio, J.A., Woodall, W.H.: A cyber-physical attack taxonomy for production systems: a quality control perspective. J. Intell. Manuf. **30**(6), 2489–2504 (2018). https://doi.org/10.1007/s10845-018-1408-9
6. O'Reilly, P., Rigopoulos, K., Feldman, L., Witte, G.: 2020 Cybersecurity and privacy annual report. Natl. Inst. Stand. Technol. (2021). https://doi.org/10.6028/NIST.SP.800-214
7. Shahin, M., Chen, F.F., Bouzary, H., Zarreh, A.: Frameworks proposed to address the threat of cyber-physical attacks to lean 4.0 systems. Procedia Manufact. **51**, 1184–1191 (2020). https://doi.org/10.1016/j.promfg.2020.10.166

8. Mahmood, T., Afzal, U.: Security analytics: big data analytics for cybersecurity: a review of trends, techniques and tools. In: 2013 2nd National Conference on Infor-mation Assurance (NCIA), pp. 129–134 (2013). https://doi.org/10.1109/NCIA.2013.6725337
9. Gaggero, G.B., Rossi, M., Girdinio, P., Marchese, M.: 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), pp. 1–4 (2019). https://doi.org/10.1109/ISAECT47714.2019.9069683
10. Ciaburro, G.: Neural Networks with R. Packt Publishing (2017). https://libproxy.txstate.edu/login?, https://search.ebsco-host.com/login.aspx?direct=true&db=cat00022a&AN=txi.b5582708&site=eds-live&scope=site. Accessed 18 Oct 2021
11. Pascanu, R., Stokes, J.W., Sanossian, H., Marinescu, M., Thomas, A.: Malware classification with recurrent networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1916–1920 (2015). https://doi.org/10.1109/ICASSP.2015.7178304
12. Shibahara, T., Yagi, T., Akiyama, M., Chiba, D., Yada, T.: Efficient dynamic mal-ware analysis based on network behavior using deep learning. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–7 (2016). https://doi.org/10.1109/GLOCOM.2016.7841778
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
14. Kim, J., Kim, J., Thu, H.L.T., Kim, H.: Long short term memory recurrent neural network classifier for intrusion detection. In: 2016 International Conference on Plat-form Technology and Service (PlatCon), pp. 1–5 (2016). https://doi.org/10.1109/PlatCon.2016.7456805
15. Zhao, Q., Zhu, Y., Wan, D., Yu, Y., Cheng, X.: Research on the data-driven quality control method of hydrological time series data. Water (Switzer-land), **10**(12), 23 (2018). https://doi.org/10.3390/w10121712
16. Yu, Y., Long, J., Cai, Z.: Network intrusion detection through stacking dilated con-volutional autoencoders. Secur. Commun. Netw. **16** (2017). https://www.hindawi.com/journals/scn/2017/4184196/. Accessed 20 Jun 2020
17. Kolosnjaji, B., Zarras, A., Webster, G., Eckert, C.: Deep learning for classification of malware system call sequences. In: Kang, B.H., Bai, Q. (eds.) AI 2016. LNCS (LNAI), vol. 9992, pp. 137–149. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50127-7_11
18. Karim, F., Majumdar, S., Darabi, H.: Insights into LSTM fully convolutional networks for time series classification. IEEE Access **7**, 67718–67725 (2019). https://doi.org/10.1109/ACCESS.2019.2916828
19. Zhiguang, W., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: 2017 International Joint Conference on Neural Networks (IJCNN), Neural Networks (IJCNN), pp. 1578–1585 (2017). https://doi.org/10.1109/IJCNN.2017.7966039
20. Park, E., Cui, X., Nguyen, T.H.B., Kim, H.: Presentation attack detection using a tiny fully con-volutional network. IEEE Trans. Inform. Forensic Secur. **14**(11), 3016–3025 (2019). https://doi.org/10.1109/TIFS.2019.2907184
21. Peterson, J.M., Leevy, J.L., Khoshgoftaar, T.M.: A review and analysis of the Bot-IoT dataset. In: 2021 IEEE International Conference on Service-Oriented System En-gineering (SOSE), Service-Oriented System Engineering (SOSE), pp. 20–27 (2021). https://doi.org/10.1109/SOSE52839.2021.00007
22. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press (1995). https://lib-proxy.txstate.edu/login?, https://lib-proxy.txstate.edu/login?. Accessed 11 Dec 2021

23. Zheng, A., Casari, A.: Feature engineering for machine learning : principles and techniques for data scientists, First edition. O'Reilly Media (2018). https://libproxy.txstate.edu/login?, https://search.ebsco-host.com/login.aspx?direct=true&db=cat00022a&AN=txi.b5167004&site=eds-live&scope=site. Accessed 11 Dec 2021
24. Livieris, I.E., Pintelas, E., Pintelas, P.: A CNN–LSTM model for gold price time-series forecasting. Neural Comput. Appl. **32**(23), 17351–17360 (2020). https://doi.org/10.1007/s00521-020-04867-x
25. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
26. Karim, F., Majumdar, S., Darabi, H., Chen, S.: LSTM fully convolutional networks for time series classification. IEEE Access **6**, 1662–1669 (2018). https://doi.org/10.1109/ACCESS.2017.2779939
27. Chollet, F.: Deep learning with Python. Manning Publications (2018). https://libproxy.txstate.edu/login?, https://search.ebsco-host.com/login.aspx?direct=true&db=cat00022a&AN=txi.b5162307&site=eds-live&scope=site. Accessed 12 Dec 2021
28. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization (2017). arXiv:1412.6980 [cs]. http://arxiv.org/abs/1412.6980. Accessed 13 Dec 2021
29. Kuhn, M., Johnson, K.: Applied predictive modeling. Springer (2013). https://libproxy.txstate.edu/login?, https://search.eb-scohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b2605857&site=eds-live&scope=site. Accessed 13 Dec 2021
30. Alpaydin, E.: Introduction to Machine Learning, vol. Third edition. Cambridge, MA: The MIT Press (2014). https://lib-proxy.txstate.edu/login?, https://search.ebscohost.com/login.aspx?di-rect=true&db=nlebk&AN=836612&site=eds-live&scope=site. Accessed 13 Dec 2021
31. Adagbasa, E.G., Adelabu, S.A., Okello, T.W.: Application of deep learning with stratified K-fold for vegetation species discrimation in a protected mountainous region using Sentinel-2 image. Geocarto Int. **37**(01), 142-162 (2019). https://doi.org/10.1080/10106049.2019.1704070
32. scikit-learn: machine learning in Python — scikit-learn 1.0.2 documentation. https://scikit-learn.org/stable/index.html. Accessed 08 Jan 2022

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.