An Ensemble of Deep Recurrent Neural Networks for Detecting IoT Cyber Attacks Using Network Traffic

Mahdis Saharkhizan, Amin Azmoodeh, Ali Dehghantanha, *Senior Member, IEEE*, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, Reza M. Parizi, *Senior Member, IEEE*

***Abstract*—Internet of Things (IoT) devices and systems will be increasingly targeted by cybercriminals (including nation state-sponsored or affiliated threat actors) as they become an integral part of our connected society and ecosystem. However, the challenges in securing these devices and systems are compounded by the scale and diversity of deployment, the fast-paced cyber threat landscape, and many other factors. Thus, in this paper, we design an approach using advanced deep learning to detect cyber attacks against IoT systems. Specifically, our approach integrates a set of Long-Short-Term-Memory (LSTM) modules into an ensemble of detectors. These modules are then merged using a decision tree to arrive at an aggregated output at the final stage. We evaluate the effectiveness of our approach using a real-world dataset of Modbus network traffic and obtain an accuracy rate of over 99% in the detection of cyber attacks against IoT devices.**

***Index Terms*—Deep Learning, Recurrent Neural Networks, Internet of Things, IoT Security, Network Traffic.**

I. Introduction

THE Internet of Things (IoT) can be broadly defined as a pervasive network of a (broad) range of connected smart nodes that offer diverse digital services, including the collection of environmental and user data. For example, IoT nodes can sense, process, and communicate (complex) information through IoT infrastructures to improve the quality and quantity of services and user experience in sectors ranging from healthcare to transportation to power management to military, etc. On the flip side, IoT devices and systems can also be an attack vector where an attacker (or an adversary) can seek to obtain information, target other entities (e.g. governments), and/or facilitate nefarious activities.

In existing networks (including those comprising IoT devices), security systems such as intrusion detection systems (IDSs) are typically used to monitor the network traffic and identify suspicious activities within the traffic [1], [2]. IDSs can be either signature-based or anomaly-based, where signature-based IDSs recognize intrusions (or suspicious activities) by finding the relationship between previously learned rules/signatures of known attacks' rules. Anomaly-based IDSs, on the other hand, monitor network traffic and compare the traffic with previously learned patterns to spot malicious activities [3]. It is known that signature-based methods are not very effective in detecting new and unknown attacks. Anomaly-based methods have shown to be able to recognize known and new attacks [4] to some degree, but they often have high false-positive rates.

In recent years, there has been renewed focus in exploring the utility of artificial intelligence (AI) techniques, such as machine learning (ML) and deep learning (DL), in designing cybersecurity solutions, such as malware detection [5], [6], [7], [8], [9], [10], [11] and threat intelligence [12], forensic investigation [13], and privacy-preserving techniques [14]. DL-based approaches typically include a learning model with several layers, and each layer contains a significant number of computational nodes. However, designing efficient and effective AI-based IoT attack detection systems remains an open research challenge.

In this paper, we propose a new approach that monitors the network traffic of IoT networks over Modbus protocol [15], and extracts network packets to train an ensemble of Long-Short-Term-Memory (LSTM) models. From there, it aggregates the output of LSTMs by a decision tree and assigns the right label to each network connection. The proposed approach is characterized by the following capabilities:

- A significantly high level of accuracy in detecting different attacks within IoT networks.
- The capability to detect attacks for different periods, including right from the start of an attack.
- The marginal false positive rate with respect to the ensemble of detection modules.

The rest of this paper is organized as follows. Section II briefly reviews the extant literature. In Section III, we outline our research methodology. In Sections IV and V, we present our proposed approach and describe the evaluation setup and discuss the findings. Finally, Section VI concludes this paper.

II. Related Literature

There has been an increasing focus on ensuring the security of IoT networks, partly due to the popularity of IoT devices in our society [16]. For example, in 2014, Oh et al. [17] introduced a signature-based intrusion detection method for IoT systems,

M. Saharkhizan is with the Shiraz University, Shiraz, Iran, e-mail: mahdis@cybersciencelab.org

A. Azmoodeh and A. Dehghantanha are with the Cyber Science Lab, University of Guelph, Canada, emails: aazmoode@uoguelph.ca, adehghan@uoguelph.ca

K-K. R. Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, Texas, USA, email: raymond.choo@fulbrightmail.org

R. M. Parizi is with the College of Computing and Software Engineering, Kennesaw State University, GA, USA, email: rparizi1@kennesaw.edu

Manuscript received xx, 2020; revised xx, 2020.

using multiple pattern-matching algorithms. Using a dataset of Snort and ClamAV extracted rules, their approach has a detection rate between 81% and 90% in different experimental settings. In another study, Anthi et al.[18] proposed an anomaly-based IDS for smart home IoT devices using a three-layer IDS that leverages a supervised learning mechanism. They achieved F-measure between 90% and 98%. However, signature-based mechanisms can be bypassed by modifying the attack's rules without degrading its harmfulness.

To facilitate malicious activity detection, Azmoodeh et al.[19] proposed a DL-based approach that extracts a graph of executable files' operation codes. They then introduced a feature selection method and used it to generate an adjacency matrix of extracted graphs prior to training a Convolutional Neural Network (CNN) in order to identify malicious and benign applications. This approach needs binary executable as input. In order to enhance the security of wireless IoT networks, Aminanto et al.[20] proposed a deep-feature extraction and selection model using deep Autoencoders. Then, they combined the autoencoder with a supervised classification algorithm and reportedly achieved a detection rate of 99.918% and a false alarm rate of 0.012% for detection impersonation attacks. This method is designed to have maximum performance for impersonation attacks while our approach considered a broader range of IoT's cyber attacks.

Activities within IoT systems rely on sequential data such as the sequence of network packets, operational codes, or environmental sensed variables. Hence, HaddadPajouh et al.[21] presented a deep Recurrent Neural Network-based approach that uses the sequence of IoT executable's operational codes for training (together with a dataset of IoT malware and benign samples). Findings from their evaluation reported an accuracy rate of 98.8% in recognizing malicious payloads. For device-level detection in IoT systems, Azmoodeh et al.[22] collected power consumption signals of infected IoT nodes. Then, they applied a grinding algorithm to these signals and trained different classifiers to identify infected nodes. They reportedly achieved an accuracy rate of 94.27% in detecting IoT nodes infected by crypto-ransomware. However, [21] and [22] requires device-level information to detect malicious activities. For enhanced scalability and robustness, Diro et al. [23] presented an LSTM-based model for distributed cyber attack detection in Fog to Things communication. They reportedly achieved an accuracy rate of 99.91% and 98.22% on the ISCX and AWID datasets, respectively. This method has not been proposed to apply to Modbus network traffic. In addition, the proposed method considers a window of network session to enhance its detection rate as well to reduce false alarms.

Supervisory Control and Data Acquisition (SCADA) and Modbus protocol are two fundamental building blocks in IoT-based systems, particularly those deployed in critical infrastructures and industrial systems [24], [25]. Anton et al. [26] evaluated the performance of ML-based anomaly detection systems on the industrial Modbus dataset of cyber attacks. They investigated the utility of Support Vector Machine (SVM), Random Forest(RF), k-Nearest Neighbor(kNN) and k-means clustering on a synthetic dataset, and the findings suggested that SVM has the highest accuracy rate of 100% in the majority of their experimental settings. Despite the significant performance, this approach was designed and assessed on a synthetic dataset which degrades its reliability to work on real environments. In another study, Goldenberg and Wool [27] modeled the behavior of Modbus protocol to detect intrusions using Deterministic Finite Automaton (DFA). Their algorithm modeled the traffic of the Modbus protocol and was sensitive to anomalies. The authors reportedly obtained accuracy rates between 65% and 99% for different numbers of DFAs. Ullah and Mahmoud [28] proposed a hybrid model for detecting anomalous SCADA data using an ML-based model, which eliminates irrelevant features to increase the accuracy of detection. According to the authors, their approach achieved a precision rate of 100% in the majority of their experiments on the KDD99 dataset. [26], [27], [28] make effort to identify anomalies that are more likely to have higher false alarm while the proposed approach is proposed to learn attack behaviors and detect them.

III. Research Methodology

In this section, we first describe our dataset, its content, and our approach to preparing it for the learning task (Section III-A). Then, we explain our approach to extract data for the learning task (Section III-B), and introduce our evaluation approach to evaluate the competency of the learning task (Section III-C).

A. Dataset

To have a clear view of the used dataset in this research, Section III-A1 provides a description of Modbus protocol, and Section III-A2 gives the information about the characteristics of the dataset.

1) Modbus over TCP/IP: The Modbus[1] protocol is widely deployed in Industrial Control Systems (ICS), and it works in a Master/Slave mode. Although it was initially developed for serial communication, it is now often used over Transmission Control Protocol (TCP). There have been different versions of Modbus proposed over the years, namely *Modbus RTU*, *Modbus ASCII*, and *Modbus over TCP/IP*.

The Modbus/TCP is a recognized and approved protocol by the Internet Assigned Number Authority (IANA) since 1996, with its default port number of 502. Instead of using the device address, Modbus uses an IP address to communicate and interact between master and slave nodes. As shown in Figure 1, Protocol Data Unit (PDU) frames that include function code to run on the device, is the fundamental part of the Modbus/TCP packet [29]. The majority of Modbus messages include commands such as read and write to control industrial nodes.

2) Dataset Description: The dataset [30] used in our study contains Modbus/TCP network traffic data, which have been simulated based on a small-sized process industrial automation scenario. The dataset includes five categories of network traffic namely *Clean* traffic, *Man-In-The-Middle (MITM)* attack, *Ping DDoS Flood* attack, *Modbus Query Flood* attack and *TCP SYN DDoS Flood* attack [31]. Network traffics of dataset were captured into *pcap*[2] files. Table I gives information about the number of pcap files corresponding to each class.

[1]http://www.modbus.org

[2]https://en.wikipedia.org/wiki/Pcap

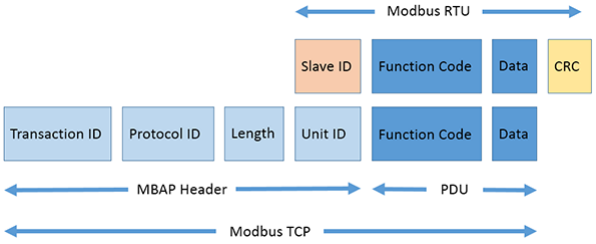

Fig. 1. Modbus PDU over TCP/IP

TABLE I
DATASET INFORMATION(NUMBER OF PCAP FILES FOR EACH CLASS)

Class	Number of pcap files
1- Clean	3
2- MITM	22
3- ModbusQueryFlooding	52
4- PingFloodDDoS	37
5- TcpSYNFloodDDoS	37

B. Extracting Modbus Flows

In order to extract the captured Modbus network traffics, CICFlowmeter[3] toolset [32] was first utilized. As a result, 83 features were extracted for each network packet. Table II provides information about the number of samples (network packet) belong to each class. Then, to eliminate features that were highly correlated to environment setup and were suspected of causing bias in resulting machine learning model, *FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort,* and *Timestamp* were removed. Finally, the column normalization on the prepared dataset was applied.

C. Evaluation Metrics

The following criteria are used to evaluate the utility of machine learning aided techniques in intrusion detection:

- True Positive (TP): indicates that an intrusion is correctly identified.
- True Negative (TN): indicates that a benign activity is detected as a non-malicious activity correctly.
- False Positive (FP): indicates that a benign activity is falsely detected as a malicious activity.
- False Negative (FN): indicates that an intrusion is not detected and labeled as a non-malicious activity.

Based on the criteria described above, the following metrics are introduced to quantify the effectiveness of a given system:

[3]http://www.netflowmeter.ca/netflowmeter.html

TABLE II
EXTRACTED SAMPLES FROM PCAP FILES

Class	Number of Samples
1- Clean	259,635
2- MITM	230,330
3- ModbusQueryFlooding	4,021,403
4- PingFloodDDoS	616,746
5- TcpSYNFloodDDoS	730,971
Total	5,859,085

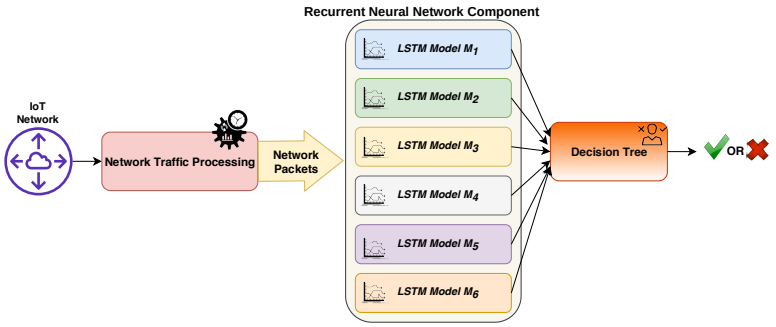

Fig. 2. The Proposed Method Overview

- **Accuracy:** indicates the number of samples that a classifier correctly detects, divided by the number of all samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- **Precision:** is another metric that indicates the ratio of predicted intrusion samples that are correctly predicted:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- **Recall:** indicates the ratio of intrusion samples that are correctly predicted:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- **F-Measure:** is the harmonic mean of precision and recall, and is defined as follows:

$$F - Measure = \frac{2 * TP}{2 * TP + FP + FN} \quad (4)$$

IV. PROPOSED METHOD

The proposed method includes a stack of deep Recurrent Neural Networks (RNN) (Section IV-A) that are trained with the prepared dataset and a decision tree that aggregates the output of RNNs (Section IV-B). Figure 2 illustrates the conceptual view of our proposed method.

A. LSTM Models

Deep Recurrent Neural Networks (RNNs) are a fundamental category of deep learning models that are proposed to apply learning tasks on sequential data. Despite the considerable capability of RNNs to learn from sequential information, they suffer from the problem of missing data dependency during the long-term data patterns [33]. Long-Short-Term-Memory (LSTM) [34] is the fundamental and widely applied architecture of RNNs that is capable of recognizing the pattern of dependency between the sequence of input data and learn the long-term pattern of data. Figure 3 shows the structure of an LSTM cell. An LSTM is formed by a set of cells, and each cell includes three main layers, namely *forget gate, input gate* and *output gate*. *Forget gate* is responsible for removing previous information of each cell and functions as follows:

$$f_t = \sigma(w_f [h_{t-1}, x_t] + b_f) \quad (5)$$

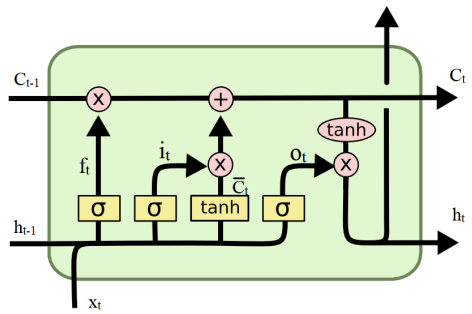

Fig. 3. An LSTM Cell

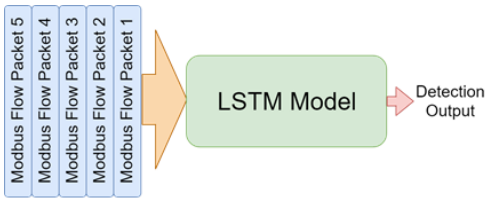

Fig. 4. An LSTM for input window size of 5 packets

where w_f and b_f are weights and bias of cell that are learned during training phase of the LSTM. Then, *input gate* which updates information of cell is calculated as follows:

$$i_t = \sigma(w_i[h_{t-1}, x_t]+b_i) \qquad \bar{C}_t = \tanh(w_c[h_{t-1}, x_t]+b_c) \quad (6)$$

Finally, the *output gate* generate the cell output for next cell and output of network as follows:

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \qquad h_t = \sigma(w_o[h_{t-1}, x_t]+b_o) * \tanh(C_t) \quad (7)$$

Generally, an LSTM network includes one layer of cells. However, increasing the depth of the network elevates its performance and accuracy for learning and recognizing complex sequential patterns [35]. The proposed method includes a stack of LSTMs having various settings to learn the various pattern of network traffic associated with clean and attack scenarios. The number of layers and the capacity of networks are two main settings to design our LSTMs. Table III gives information about the proposed method's LSTMs. Besides, as described in Section III, we train the proposed method for different window sizes of the packet, and therefore, for each window size, the input size of each LSTMs varies. Figure 4 illustrates an LSTM for window size of 5.

TABLE III
THE PROPOSED METHOD'S LSTMS SETTINGS

Mode Name	Number of Layers	Layer Size
LSTM-1	1	(100)
LSTM-2	1	(200)
LSTM-3	2	(100,100)
LSTM-4	2	(200,100)
LSTM-5	3	(100,100,100)
LSTM-6	3	(100,50,20)

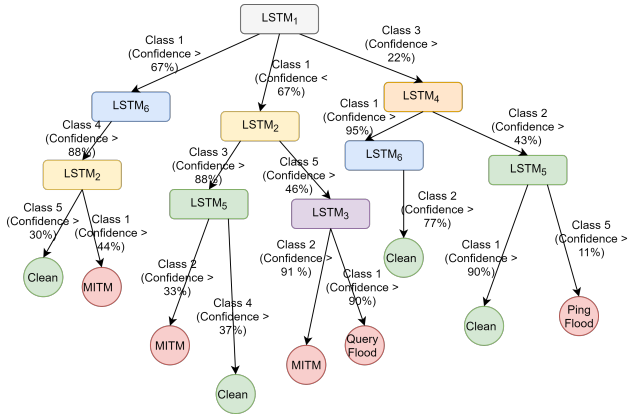

Fig. 5. A Schematic View of Decision Tree in the Proposed Method

B. Ensemble of LSTMs

In order to make an aggregated decision about the output of LSTMs, we integrate a Decision Tree (DT) component into our proposed method. The DT accepts a collection of confidence rates for each class within the dataset and decides about output. The input of the DT module from LSTMs is as follows:

$$Input\ of\ DT = \{LSTM_{i,c}\ where\ i \in \{Number\ of\ LSTMs\} \ AND\ c\ \in Number\ of\ Classes\} \quad (8)$$

$LSTM_{i,c}$ refers to confidence rate of i^{th} LSTM trained model for class c. The DT accepts these confidence rates as inputs and hierarchically learns the correlation between the confidence rate of LSTMs and the true label of network traffic. Figure 5 schematically illustrates how DT component functions in the proposed method. In other words, DT identifies the manifold of the output space of LSTMs and provides us with an explainable model to decide about the final label.

V. EVALUATIONS AND FINDINGS

In this section, we evaluate the potential of state-of-the-art classification algorithms on the prepared dataset (Section V-A). Then, we describe the performance of different LSTMs' in recognizing Modbus cyber attacks (see Section V-B). Findings from Section V-C demonstrate the robustness of the proposed approach in detecting IoT cyber-attack using network traffic. We also discuss the training and inference times of the proposed system.

The experiments are implemented on an Ubuntu 16 system with 128GB of memory and 32 Core i7 CPUs. All scripts for extracting and preprocessing data as well as learning tasks are written in *Python 3.7*. We utilize *Tensorflow*[4] as our deep learning platform. The experiments are performed for seven different window sizes, namely: $\{1, 5, 10, 15, 20, 30, 40\}$, for training the LSTMs. We then apply the 10-fold cross-validation technique [36].

A. State-of-the-art Classifiers

Before evaluating LSTM and the proposed method's outcomes and in order to assess the performance of prevalent clas-

[4]https://www.tensorflow.org/

TABLE IV
PERFORMANCE OF STATE-OF-THE-ART CLASSIFIERS: A COMPARATIVE SUMMARY

Classifier	Accuracy	Precision	Recall	F-Measure
KNN	85.09%	87.50%	83.33%	85.37%
SVM	86.96%	89.89%	86.96%	88.40%
MLP	88.20%	88.17%	91.11%	89.62%
Random Forest	90.68%	92.22%	91.21%	91.71%

sification algorithms, we apply four state-of-the-art classifiers, namely: k-Nearest Neighbor (KNN), Multi-layer Perceptron (MLP), Support Vector Machine (SVM) and Random Forest (RF), on the prepared dataset. We use *Scikit-learn* [5] to implement these classification methods. As for KNN, k=1 and for MLP, the size 0f hidden layer is set to 200.

Table IV summarizes the results of the experiments.

B. LSTMs

In the first stage of our study, we train a set of LSTMs (see also Section IV-A) and evaluate their performance to identify IoT cyber attacks using network traffic. Specifically, we train six different LSTMs having different structures and on seven different window sizes. The *epoch* for training deep learner is 200 and batch size sets to 1024. In addition, we utilize the *Adam* optimizer.

During our experiments, we monitor the performance metrics described in Section III-C, in order to analyze the first stage of our proposed method. Figures 6 to 11 present the performance of LSTMs in classifying Modbus network traffic for different window sizes. For each metric, the figure includes a group of bar charts that report the metric's highest obtained values for each window size setting.

From the findings, we observe that the LSTM classification approach outperforms the other state-of-the-art classifiers (see Table IV). Also, we observe that the *Precision* of the trained model surpasses other evaluation criteria, and the performance of different LSTMs varies over window sizes (i.e. LSTMs have learned different patterns of network traffic). A general trend is an increased window size results in increased performance. We also observe that $LSTM_4$ is the most accurate model, with an accuracy rate of 95.59% (for window size = 40) and the average accuracy of LSTM models is 92.46%. $LSTM_3$ (for window size = 40) is the best model to positively predict samples, with a precision rate of 99.7%. The average precision rate is 95.1%. In terms of recall metric, $LSTM_4$ for window size = 40 achieves 95.54% true positive rate and the average is 91.92%. For the mean of precision and recall, $LSTM_4$ for window size = 40 has an F-Measure of 95.58% and the average is 92.45%.

[5]https://scikit-learn.org/

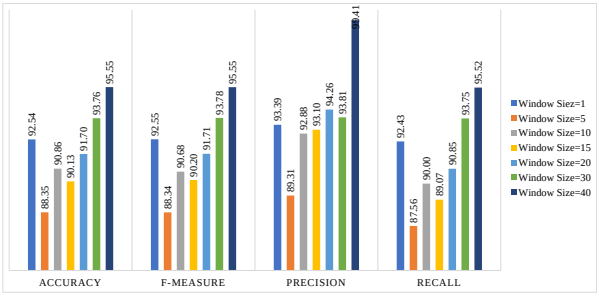

Fig. 6. Performance of LSTM Model#1 Over Modbus Network Traffic

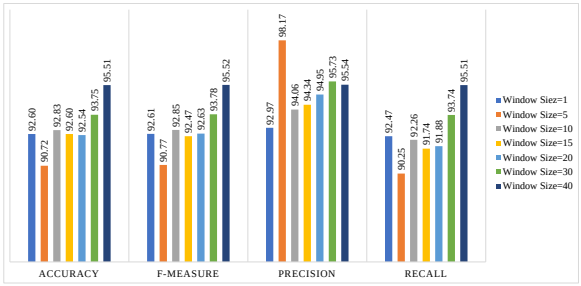

Fig. 7. Performance of LSTM Model#2 Over Modbus Network Traffic

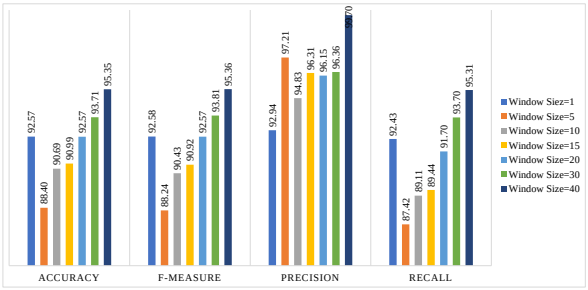

Fig. 8. Performance of LSTM Model#3 Over Modbus Network Traffic

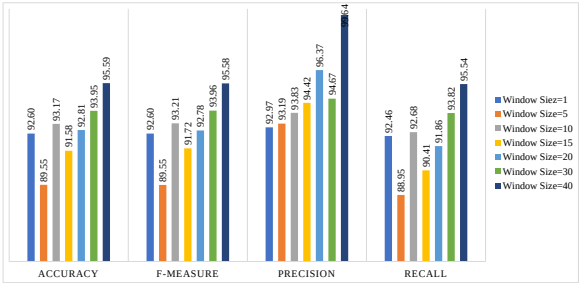

Fig. 9. Performance of LSTM Model#4 Over Modbus Network Traffic

C. Proposed Approach

Similar to the preceding section, we evaluate the performance of the proposed model for different window sizes while a decision tree aggregates the output of LSTMs. As shown in Table V, the best accuracy rate obtained is for window size

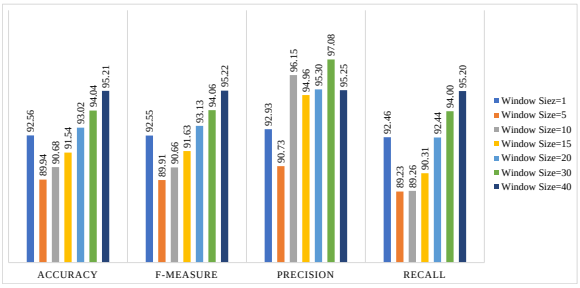

Fig. 10. Performance of LSTM Model#5 Over Modbus Network Traffic

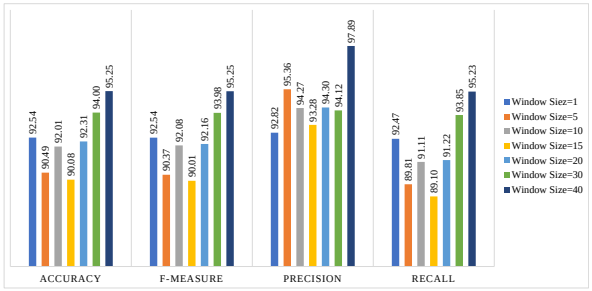

Fig. 11. Performance of LSTM Model#6 Over Modbus Network Traffic

= 40, where our proposed approach obtains an accuracy rate of 99.62% and the average accuracy rate is 98.99%. In terms of precision, we achieve 99.41% precision, and the average decreases from 95.1% to 96.51%. The proposed method is capable of reaching a detection rate of 99.35% for window size = 5, and the average detection rate decreases to 92.32%. In terms of F-Measure, our approach obtains 99.30%, and the average decreases from 92.45% to 95.67%. One can observe that the proposed method outperforms single LSTMs and the other state-of-the-art classification algorithms for both maximum and average performance.

TABLE V
PERFORMANCE OF OUR PROPOSED APPROACH FOR DIFFERENT WINDOW SIZES

Winow size	Accuracy	Recall	Precision	F1-measure
window-size-1	96.459	93.454	94.205	93.812
window-size-5	99.237	99.351	99.265	99.308
window-size-10	99.428	85.211	85.185	85.198
window-size-15	99.370	92.209	99.363	94.594
window-size-20	99.369	98.428	99.185	98.797
window-size-30	99.465	98.579	99.236	98.892
window-size-40	99.620	98.883	99.418	99.142

D. Time Discussion

In terms of time complexity, an ideal cyber-attack detection should have reasonable and short training and inference times while having acceptable detection performance. During the training time, our proposed method (constructed by a set of LSTMs) achieves acceptable performance within about 25 training epochs that last less than 65 seconds. Figure 12 presents the average performance of LSTMs (see also Section IV-A) over training epochs.

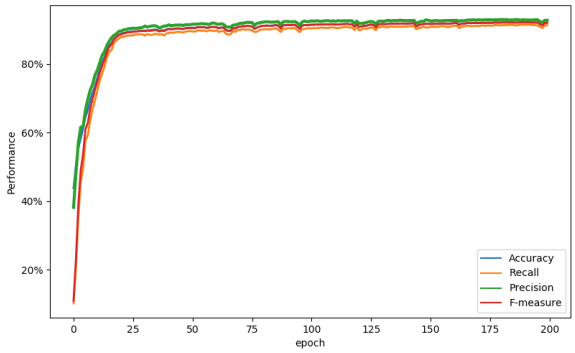

Fig. 12. Average performance of LSTM base models

In terms of inference time, the proposed method requires only 40 packets in the Modbus network session to detect cyber attacks (see Table V). Based on the dataset we used, it requires less than a second for all different attack scenarios to transmit 40 packets over a network session. In addition, the proposed method latency for processing network packets and inferring is approximately $55ms$ on average for the experimental workstation.

VI. CONCLUDING REMARKS

The expanding number of industries utilizing IoT devices partly contributed to an increase in the frequency, size, and severity of cyber attacks against IoT networks; thus, creating an arms race between the cyber defenders and the cyber attackers. In this paper, we presented a novel ensemble method to detect IoT cyber attacks over Modbus network traffic. In our approach, we integrated an ensemble of LSTM deep models and aggregated their outputs to achieve enhanced robustness. Findings from our evaluations demonstrated the potential of our approach in an IoT system, where using the decision tree as an aggregator provides an explainable structure to enhance the transparency of the proposed method [37].

In the future, we will explore the explainability of LSTM models to propose a more transparent deep learning model for detecting IoT cyber attacks. We also plan to deploy the proposed approach to different IoT protocols and transfer the learned Modbus cyber attacks to other domains.

REFERENCES

[1] K. A. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, 2019.

[2] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, 2018.

[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, vol. 36, p. 42–57, 01 2013.

[4] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[5] E. M. Dovom, A. Azmoodeh, A. Dehghantanha, D. E. Newton, R. M. Parizi, and H. Karimipour, "Fuzzy pattern tree for edge malware detection and categorization in iot," *Journal of Systems Architecture*, vol. 97, pp. 1 – 7, 2019.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[7] A. N. Jahromi, S. Hashemi, A. Dehghantanha, K.-K. R. Choo, H. Karimipour, D. E. Newton, and R. M. Parizi, "An improved two-hidden-layer extreme learning machine for malware hunting," *Computers & Security*, p. 101655, 2019.

[8] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, "A survey on malware detection using data mining techniques," *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, p. 41, 2017.

[9] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, 2019.

[10] H. Karimipour, A. Dehghantanha, R. M. Parizi, K. R. Choo, and H. Leung, "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids," *IEEE Access*, vol. 7, pp. 80778–80788, 2019.

[11] H. HaddadPajouh, R. Khayami, A. Dehghantanha, K.-K. R. Choo, and R. M. Parizi, "Ai4safe-iot: an ai-powered secure architecture for edge layer of internet of things," *Neural Computing and Applications*, pp. 1 – 15, 2020.

[12] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Computers & security*, vol. 72, pp. 212–233, 2018.

[13] J. Mena, *Machine learning forensics for law enforcement, security, and intelligence*. Auerbach Publications, 2016.

[14] J. Zhao, R. Mortier, J. Crowcroft, and L. Wang, "Privacy-preserving machine learning based data analytics on edge devices," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, pp. 341–346, 2018.

[15] N. Goldenberg and A. Wool, "Accurate modeling of modbus/tcp for intrusion detection in scada systems," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 63–75, 2013.

[16] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on internet of things security: Requirements, challenges, and solutions," *Internet of Things*, p. 100129, 2019.

[17] D. Oh, D. Kim, and W. Ro, "A malicious pattern detection engine for embedded security systems in the internet of things," *Sensors*, vol. 14, no. 12, pp. 24188–24211, 2014.

[18] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home iot devices," *IEEE Internet of Things Journal*, vol. 6, pp. 9042–9053, Oct 2019.

[19] A. Azmoodeh, A. Dehghantanha, and K. R. Choo, "Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning," *IEEE Transactions on Sustainable Computing*, vol. 4, pp. 88–95, Jan 2019.

[20] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep abstraction and weighted feature selection for wi-fi impersonation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 621–636, March 2018.

[21] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," *Future Generation Computer Systems*, vol. 85, pp. 88–96, 2018.

[22] A. Azmoodeh, A. Dehghantanha, M. Conti, and K.-K. R. Choo, "Detecting crypto-ransomware in iot networks based on energy consumption footprint," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1141–1152, 2018.

[23] A. Diro and N. Chilamkurti, "Leveraging lstm networks for attack detection in fog-to-things communications," *IEEE Communications Magazine*, vol. 56, pp. 124–130, Sep. 2018.

[24] S. Ghosh and S. Sampalli, "A survey of security in scada networks: Current issues and future challenges," *IEEE Access*, vol. 7, pp. 135812–135831, 2019.

[25] B. Zhu and S. Sastry, "Scada-specific intrusion detection/prevention systems: a survey and taxonomy," in *Proceedings of the 1st workshop on secure control systems (SCS)*, vol. 11, p. 7, 2010.

[26] S. D. Anton, S. Kanoor, D. Fraunholz, and H. D. Schotten, "Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ARES 2018, pp. 41:1–41:9, ACM, 2018.

[27] N. Goldenberg and A. Wool, "Accurate modeling of modbus/tcp for intrusion detection in scada systems," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 63 – 75, 2013.

[28] I. Ullah and Q. H. Mahmoud, "A hybrid model for anomaly-based intrusion detection in scada networks," in *2017 IEEE International Conference on Big Data (Big Data)*, pp. 2160–2167, Dec 2017.

[29] D. Peng, H. Zhang, L. Yang, and H. Li, "Design and realization of modbus protocol based on embedded linux system," in *2008 International Conference on Embedded Software and Systems Symposia*, pp. 275–280, July 2008.

[30] I. Frazão, P. H. Abreu, T. Cruz, H. Araújo, and P. Simões, "Denial of service attacks: Detecting the frailties of machine learning algorithms in the classification process," in *International Conference on Critical Information Infrastructures Security*, pp. 230–235, Springer, 2018.

[31] N. R. Rodofile, K. Radke, and E. Foo, "Framework for scada cyber-attack dataset creation," in *Proceedings of the Australasian Computer Science Week Multiconference*, p. 69, ACM, 2017.

[32] A. H. Lashkari., G. D. Gil., M. S. I. Mamun., and A. A. Ghorbani., "Characterization of tor traffic using time based features," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,*, pp. 253–262, INSTICC, SciTePress, 2017.

[33] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies."

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 190–198, Curran Associates, Inc., 2013.

[36] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia of database systems*, pp. 532–538, 2009.

[37] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, 2017.

Mahdis Saharkhizan received her B.Sc. degree in Electrical Engineering majoring in Electronics from Shiraz University in fall of 2017. She is mostly interested in application of machine learning in cyber security, IoT and cloud computing. Since 2015, she is an active member of IEEE Shiraz University Student Branch and has arranged different seminars and courses to help engineering students enhancing their programming and other necessary skills.

Amin Azmoodeh received his B.Ss degree in computer engineering and M.Sc degree in artificial intelligence from Shiraz University. He is pursuing his Ph.D. at the University of Guelph, Canada. His main research interests include the theory of machine learning and artificial intelligence and adversarial machine learning. Besides, Amin is interested in the application o machine learning, especially in cybersecurity and digital forensics. He is a Microsoft Certified Professional and has several years' experience in analyzing and implementing Enterprise Resource Planning software.

Ali Dehghantanha is the director of Cyber Science Lab in the School of Computer Science, University of Guelph (UofG), Ontario, Canada. He has served for more than a decade in a variety of industrial and academic positions with leading players in Cyber-Security and Artificial Intelligence. Prior to joining UofG, he has served as a Sr. Lecturer in the University of Sheffield, UK and as an EU Marie-Curie International Incoming Fellow at the University of Salford, UK. He has PhD in Security in Computing and a number of professional certifications including CISSP and CISM. His main research interests are malware analysis and digital forensics, IoT security and application of AI in the Cyber Security.

Kim-Kwang Raymond Choo (SM'15) received a Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA). In 2016, he was named the Cybersecurity Educator of the Year - APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn), and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence in Scalable Computing (Middle Career Researcher), 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, British Computer Society's 2019 Wilkes Award Runner-up, 2019 EURASIP Journal on Wireless Communications and Networking (JWCN) Best Paper Award, Korea Information Processing Society's Journal of Information Processing Systems (JIPS) Survey Paper Award (Gold) 2019, IEEE Blockchain 2019 Outstanding Paper Award, Inscrypt 2019 Best Student Paper Award, IEEE TrustCom 2018 Best Paper Award, ESORICS 2015 Best Research Paper Award, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. He is also a Fellow of the Australian Computer Society, and Co-Chair of IEEE Multimedia Communications Technical Committee's Digital Rights Management for Multimedia Interest Group.

Reza M. Parizi is the director of Decentralized Science Lab (dSL) at Kennesaw State University, GA, USA. He is a consummate technologist and software security researcher with an entrepreneurial spirit. He is a senior member of IEEE, IEEE Blockchain Community, and ACM. Prior to joining KSU, he was with the New York Institute of Technology. He received a Ph.D. in Software Engineering in 2012 and M.Sc. and B.Sc. degrees in Software Engineering and Computer Science respectively in 2008 and 2005. His research interests are R&D in decentralized AI, blockchain systems, smart contracts, IoT and emerging issues in the practice of secure software-run world applications.