Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications

Abebe Diro and Naveen Chilamkurti

The authors propose an LSTM network for distributed cyber-attack detection in fog-to-things communication. They identify and analyze critical attacks and threats targeting IoT devices, especially attacks exploiting vulnerabilities of wireless communications. The conducted experiments on two scenarios demonstrate the effectiveness and efficiency of deeper models over traditional machine learning models.

Abstract

The evolution and sophistication of cyber-attacks need resilient and evolving cybersecurity schemes. As an emerging technology, the Internet of Things (IoT) inherits cyber-attacks and threats from the IT environment despite the existence of a layered defensive security mechanism. The extension of the digital world to the physical environment of IoT brings unseen attacks that require a novel lightweight and distributed attack detection mechanism due to their architecture and resource constraints. Architecturally, fog nodes can be leveraged to offload security functions from IoT and the cloud to mitigate the resource limitation issues of IoT and scalability bottlenecks of the cloud. Classical machine learning algorithms have been extensively used for intrusion detection, although scalability, feature engineering efforts, and accuracy have hindered their penetration into the security market. These shortcomings could be mitigated using the deep learning approach as it has been successful in big data fields. Apart from eliminating the need to craft features manually, deep learning is resilient against morphing attacks with high detection accuracy. This article proposes an LSTM network for distributed cyber-attack detection in fog-to-things communication. We identify and analyze critical attacks and threats targeting IoT devices, especially attacks exploiting vulnerabilities of wireless communications. The conducted experiments on two scenarios demonstrate the effectiveness and efficiency of deeper models over traditional machine learning models.

Introduction

Machine learning (ML) has been proposed to detect intrusions for over a decade, although its real-world applications were impeded due to significant human effort requirements for feature engineering [1]. Despite feature extraction constituting the majority (about 80 percent [2]) of attack detection efforts, the obtained features fail to represent the underlying accurate patterns of the network data. For instance, features such as duration of the connection, number of bytes sent and received, and error counts are some of the features manually engineered from the packet dump of network traffic. It is probable that these manually crafted features might not be the essential features of intrusion detection systems, and the automatic feature engineering could extract more robust patterns that reflect the real nature of the network traffic. ML also struggles to detect the evolving nature of cyber-attacks accurately due to human errors introduced into the process of feature extraction as well as its shallow algorithms. These problems have hindered the adoption of ML to penetrate the markets of security companies, and signature-based solutions are still widely used despite their high false alarms and low accuracy.

Deep learning (DL) has recently been extensively applied in big data areas such as image classification, object recognition, and natural language processing. It is an advanced ML scheme, consisting of deeply layered neural networks through which features are hierarchically and automatically learned. The recent wide applications for DL is credited with improvements in hardware such as graphical processing untis (GPUs) and optimization in software such as neural networks [3]. One of the advancements in deep learning algorithms is the evolution of long short-term memory networks (LSTMs) from recurrent neural networks (RNNs) as a model capable of learning patterns in long sequences [4]. LSTM networks can be applied to learn features and patterns in network data for classifying them as benign or attack. As a DL algorithm, LSTM reduces the burden of feature engineering over classical ML since it operates on raw data. It is also evident that the LSTM network is resilient against adversaries because adversaries cannot adapt to feature learning algorithms to advance their techniques of breaching. While most DL algorithms work on numeric datasets, LSTM is effective in training on unstructured datasets like those of the Internet of Things (IoT). Historical data could be used to detect attacks over time as the investigation of a single deep packet might not be sufficient to effectively detect patterns. The sequence of network packets can be fed to DL algorithms that can memorize the sequences over a long time. Unlike traditional ML, LSTM networks can be employed to recognize repetitions of attack patterns in the long sequence of packets independent of window size.

The success of DL algorithms in various fields in the presence of emerging field fog/IoT computing brings tremendous security challenges. Despite the availability of big data circulating near IoT, the challenge of designing and implementing robust attack detection systems for IoT devices has resource constraints, delay sensitivity, and distribution issues [5]. These small, smart objects lack the power of processing to host cryptographic

Digital Object Identifier:
10.1109/MCOM.2018.1701270

The authors are with La Trobe University.

functions and ML models, while the remote cloud suffers from the scalability of handling massive IoT communication and analysis of network data for security, and hence fails to respond quickly to attack detection in IoT devices. For instance, the deployment of attack detection in oil pipeline IoT devices in the cloud has shortcomings such as expensive data transport to/from the cloud for analysis, increased reaction time to detect intrusions, and susceptibility to eavesdropping attack. However, fog computing architectures could provide a unique opportunity in offloading security functions to fog nodes, and support the provision of attack detection in proximity for delay-sensitive applications [6]. In the oil pipeline scenario, fog nodes can act as proxies with computing, control, and storage capabilities for attack detection in the sensors and actuators of the oil infrastructure. On the other hand, the generation of big data from smart devices, or IoT, in fog computing could be locally harnessed to detect suspicious behaviors in IoT devices at fog nodes. This rich data could be employed to model traffic patterns over time to detect suspicious and intrusion behaviors. Thus, leveraging algorithms that learn in a distributed manner on local data captured at each fog node could mitigate the attacks and threats in IoT devices.

In this research, we investigate the self-learning capabilities of deep learning in detecting cyber-attacks in fog-to-things computing environments employing fog nodes as data and control processing spots. The article presents two attack detection case studies: denial of service (DoS) detection and multi-attack detection in fog-to-things computing. Thus, in this article, the main contributions include:

1. Study of current state-of-the-art DL-based cyber-attack detection schemes
2. The adopted and proposed LSTM-network-based attack detection in fog-to-things environments
3. The designed and implemented distributed attack detection scheme that considers the resource constraints of fog-to-things communication
4. A comparison of the performance of shallow algorithms with LSTM networks from discriminating attacks

The remaining parts of the article are organized as follows. The next section discusses the principles and architecture of fog-to-things communications. Then an overview of the LSTM network as a DL algorithm is given. Related works are discussed next. We outline attacks and threats in fog-to-things interaction. Next, we review related works and outline the differences of our work from related research. We proceed to explain our attack detection system's architecture, datasets employed, and evaluation. The final section provides conclusions and indicates future directions.

Fog-to-Things Computing: Principles and Architecture

Fog computing has been envisioned as the extension of remote and centralized cloud computing into distributed nodes closer to sources of data. The architecture provides processing, communications, storage, and control functions at the edge of the network, which solves the high bandwidth consumption, quality of service (QoS) degradation, scalability issues, and high latencies of cloud computing. In bridging the gap between IoT and the cloud, fog computing also enables pooling of idle client resources along the cloud-to-thing continuum, which is a natural way of increasing efficiency. Being closer to clients, fog-to-things communications support real-time applications that embed artificial intelligence such as augmented and virtual reality. It is an ideal paradigm for next-generation networks that support the convergence of IoT, fifth generation (5G), and artificial intelligence (AI) applications. Figure 1 shows the typical fog architecture: interaction between IoT, fog nodes, and the cloud. Thus, smart city applications such as smart grids, smart transportation, and healthcare systems greatly benefit from fog computing as it provides embedded and distributed intelligence for IoT in data collection and resource utilization.

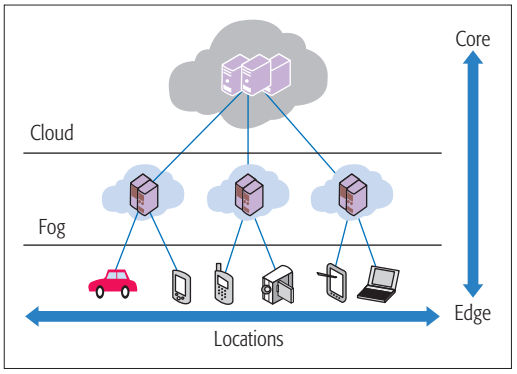

Figure 1. Basic architecture of a fog network [3].

The emerging usage of fog computing does not replace cloud architecture with edge computing, but rather complements the cloud by bringing intelligence into distributed fog nodes. Even though fog computing provides closeness, distribution, and mobility support to IoT devices, cloud computing offers elastic resources for data analytics. Fog computing can be a new distributed platform to efficiently address the architectures and vulnerabilities of IoT devices. Unlike data centers or cellular core networks, it provides computation and control, and storage of security functions closer to things (or IoT). This enables a security function continuum in the communication of cloud-to-things, which is beyond the perimeter of IT security in the premises. Cyber-attacks from the Internet are more quickly detected at fog nodes than in the cloud, which makes fog-computing-based IoT security appealing for critical infrastructures. By containing communications in the proximity of IoT devices, fog enables mitigation of eavesdropping and man-in-the-middle attacks effectively.

LSTM Network Overview

The major shortcoming of traditional ML and many DL algorithms is the lack of memory to recall previous events.

Recurrent neural networks solve this limitation by maintaining loops from current to previous states to enable information persistence. It is suitable for lists and sequences as the overall network

Cyber-attacks from the Internet are more quickly detected at fog nodes than in the cloud, which makes fog computing based IoT security appealing for critical infrastructures. By containing communications in the proximity of IoT devices, fog enables to mitigate eavesdropping and man-in-the-middle of attacks effectively.

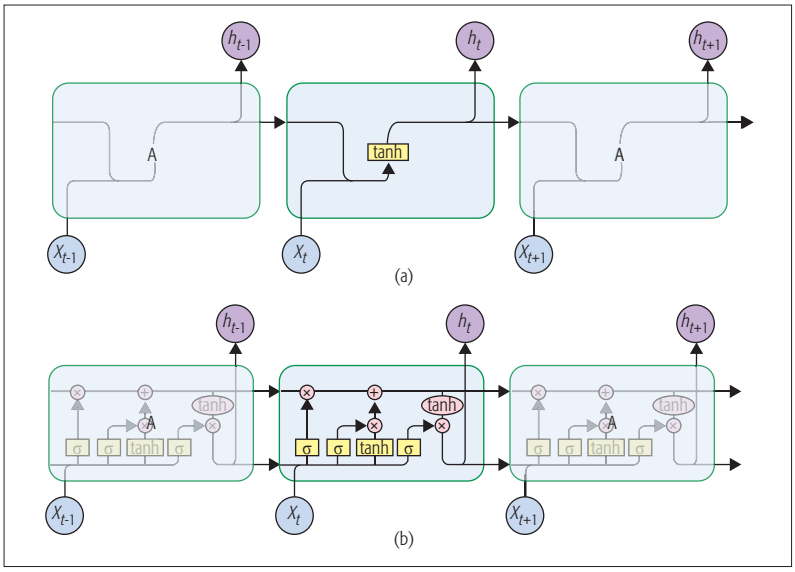

Figure 2. a) RNN with single tanh layer; b) LSTM structure.

consists of a similar block of smaller networks operating in the chain. For instance, the knowledge of previous network traffic state could be used for understanding the current state of the network. RNNs having X input and h output value at time t. As an extension of the artificial neural network (ANN), the traditional RNN has the same problem of feed-forward neural network (i.e., the vanishing problem), which is solved by its recent version, LSTM. LSTM is a special type of RNN that recalls the long-term dependencies over a long period of time. The repeating module of RNN consists of simple structures such as a single tanh layer (Fig. 2a), while LSTM has four interacting layers. The difference between repeating modules of RNN and LSTM is depicted in Fig. 2b.

As shown in Fig. 2b, the repeating module of LSTM has a straight line at the top, which acts as cell state. The three sigmoid layers together with their corresponding pointwise multiplication operation represent gates that control and protect cell states. The outputs of the four neural nets are flown into cell state. The first layer of sigmoid known as forget gate decides the information passed to or discarded from previous cell state based on the output from h_{t-1} and X_t. The output of the forget gate can be either 1, which indicates to keep the output, or 0, which represents discard information. The second sigmoid layer, also known as input gate, enables the decision on which information is to be updated, while a tanh layer creates a list of new candidate values for the cell state. The new update is created by the combination of the second sigmoid and tanh layers to replace the old cell state by the new cell state. Then the old cell state is multiplied with the output of the forget gate, which is later added to the combined output of the input and tanh gates to produce new cell state. Finally, the filtered output is decided by the last sigmoid layer and the final tanh layer based on cell state.

Related Works

As fog computing is an emerging field, few research works have been conducted on cybersecurity issues.

LSTM-RNN has been applied to the KDDCUP 99 dataset to detect intrusions [7]. It has also been employed in [8] by modeling network traffic as time series events. The accuracy obtained using LSTM networks for intrusion detection in these works indicated the capability of DL in extracting patterns from raw data. It is also evident that the long memory of LSTM on a long sequence of data enabled the model to perform better than other classical ML algorithms in intrusion detection. Other research [9] conducted on anomaly detection demonstrated the LSTM network's effectiveness to model normal and anomalous patterns. The performance of the model is another indicator of deep learning efficiency for intrusion detection.

In [10], LSTM was applied to discriminate algorithmically generated domains from normal URLs. The authors applied LSTM on the raw sequence of characters in the URL of domain names. The obtained area under the curve (AUC) demonstrated that the LSTM model outperformed the n-gram approach. This technique is enormously important for detecting malware injection using URLs. A related work [11] investigated detecting phishing sites using the LSTM model. The application of LSTM for feature extraction eliminated the need to engineer lexical features and statistical analysis from URLs. The experimental result was compared with random forest, and it was found that the LSTM model is more accurate. The same study also showed that detection speed and memory requirements of DL are better than those of classical ML. The authors in [12] proposed an intrusion detection system (IDS) named eXpose using the concept of character embedding for feature extraction. Although the study used a convolutional neural network (CNN), the overall process is similar to LSTM training and test. By comparing with classical ML models, the study asserted the superiority of the DL approach over traditional ML algorithms.

The result of the research on IDS conducted using LSTM is an indication of DL's potential to succeed in cybersecurity. It is consistent with other big data areas such as image recognition. Our approach is different from the studies in that our approach employs:

- Fog nodes to offload security functions in detecting attacks in IoT devices
- A local detection scheme on local data and parameters using distributed DL (the best parameters and models from each fog node are updated and exchanged via coordinator node)
- Distributed and scalable attack detection architecture that offloads the cloud, and decreases latency of detection for real-time IoT applications

Attacks in Fog-to-Things Communications

Fog-to-things computing requires end-to-end protection against attackers in the cloud-to-things continuum. The trustworthiness of the communication in the overall fog infrastructure begins with securing individual nodes. Once trusted nodes are formed, the attack detection system plays a pivotal role in the fog environment as defending

	Authentication	Availability	Integrity	Confidentiality
Network attacks	Impersonation/M-in-M	DoS/network flooding	Replay	Eavesdropping
Software attacks	Injection of malware	DoS/resource flooding	Injection of malware	Injection of malware
Hardware attacks	Injection of Trojan	Jamming/bandwidth flooding	Injection of Trojan	Injection of Trojan

Table 1. Attack categories and the violated security goals.

mechanisms alone cannot guarantee the security of networks and systems. This means that the attack detection system provides an additional layer of security for the communications of fog-to-fog, IoT-to-fog, and fog-to-cloud. Although the security requirements of fog-to-things communication remain the same with IT infrastructure, the threat landscape might extend to the physical environment due to the presence of IoT devices. The threats can be seen as network-oriented, software, hardware, and insider attacks. This study is meant to handle the first three attacks. Table 1 summarizes threat categories in IoT and security goals.

As shown in Table 1, fog-to-things communication has several threats and attacks, which could be categorized as DoS/flooding, injection, and impersonation [14]. One of the threats for IoT devices is a DoS attack. DoS attack refers to a cyber-attack that denies legitimate users or entities from accessing services or resources such as communication bandwidth, processors, and memories. In large- scale attacks, multiple coordinated sources can be used in launching attacks against a target, which is known as distributed DoS (DDoS). According to [13], DoS attacks will reach 17 million by 2020, which means an increase by 300 percent from the incidents in 2015. The main protocols used for DoS attack include TCP, UDP, HTTP, ICMP, DNS, and so on. Attacks employing protocols such UDP and ICMP are known to deplete network bandwidth, while TCP SYN flood exhausts processor and memory resources. Recent DoS attacks have been witnessed to have evolved from single flooding to multi- vector attacks. The primary target of DoS attacks in IoT is the unavailability of critical infrastructure and businesses supported by these smart objects. For instance, deauthentication flooding or the surge in the number of other management packets is a potential DoS attack in fog-to-things computing as the communication environment consists of mostly wireless systems. Thus, real-time detection of DoS attacks has become the main concern for large organizations and governments.

Injection attacks are the transmission of undersized packets in massive amounts. In Wi-Fi communications, it could be Active Reservation Protocol (ARP) injections in which two or more frames have repeated initial vectors (IVs). The duration of these frames is also different from normal traffic. It can also be fragmentation attacks during which short and fragmented frames are injected instantly or repeated over time if it is not successful at once. This attack is known to produce small frames that have static and invalid value in the destination address of a frame. A closely related attack in this category is the impersonation attack. This is carried out by fake access points (APs) or modes of communication around legitimate nodes by spoofing addresses. The fake AP broadcasts a massive number of beacon frames to appear stronger in signaling than the real communication node. This type of attack is combined with deauthentication attacks in which the attacking AP or node forces clients to connect to it.

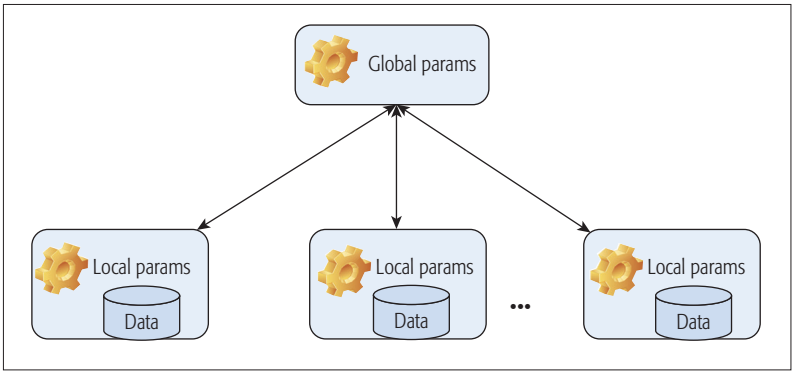

Figure 3. LSTM-network-based attack detection architecture for IoT/fog networks.

The Proposed Attack Detection Scheme

The resource limited nature of IoT devices necessitates lightweight cyber-attack detection closer to applications. This could be achieved by distributing detection sensors on fog nodes in the communication of fog-to-things. However, the distributed fog nodes should be coordinated for experience and parameter exchange. This section discusses the architecture, implementation, evaluation, and results of the LSTM-based cyber-attack detection system in fog-to-things communication.

System Architecture and Algorithm, and Dataset

The enormous resources of cloud computing cannot be harnessed by IoT devices due to centralization and remote location. The real-time applications supported by IoT devices need the security and other functions to be scalable and fast in response. Figure 3 shows the distributed architecture of attack detection in which fog nodes host detection functions.

The architecture reveals that the training and detection function are locally performed on each node, while the coordinating node computes and distributes the update of models and parameters to every node. This arrangement gives the mechanism of partitioning data that would have been stored, trained, and tested at a single spot, the cloud. It distributes computations, controls, and data stored on local nodes so that each node detects the intrusion of nearby IoT devices while it exchanges learned experiences with neighbor nodes via a coordinator node. This is a mechanism of building a lightweight and scalable attack detection scheme for resource-limited nodes.

Parameters updated by optimizers and SGD on each node are sent to the coordinator node. The coordinating node calculates the aggregated update of parameters from the updates of each node, and exchanges the parameters by sending them back to the distributed nodes.

1. If (first_time)
   a. Receive initialize parameters W_{ji} and b_{ji} from coordinator node
2. Else
   a. Receive update parameters ΔW_{ji} and Δb_{ji} from coordinator node3. Obtain $DATA_{ln}$ local data input
4. Get sample data $DATA_{na}$ and divide into $DATA_{nac}$
5. For threads n_c on node n, done in parallel:
   a. Train instance $i \in DATA_{nac}$
   b. Update $w_{ji} \in W_n$, biases $b_{ji} \in b_n$

$$W_{ji} \coloneqq W_{ji} - \alpha \frac{\partial L(W, b \mid j)}{\partial W_{ji}}$$

$$b_{ji} \coloneqq W_{ji} - \alpha \frac{\partial L(W, b \mid j)}{\partial b_{ji}}$$

   c. Return updates of ΔW_{ji} and Δb_{ji}
6. Send ΔW_{ji} and Δb_{ji} to coordinator node
7. Apply step (2)

Algorithm 1.

The detection mechanism starts by collecting data per node. The parallel training and parameter update algorithm (Algorithm 1) has been adopted from sequential stochastic gradient descent (SGD) because fog-to-things computing is distributed in nature. Having initial training weights W_n, a learning rate α, and bias parameters b_n, $DATA_n$ can be the data collected at all distributed n nodes. Each local data $DATA_{ln}$ on the given fog node could be distributed into $DATA_{na}$ samples. The $DATA_{na}$ samples on each node could further be divided into $DATA_{nac}$ on processor threads n_c.

As indicated in the above algorithm , the coordinator node sends initial random parameters to the rest of the nodes, and training of the models is performed with distributed nodes. Parameters updated by optimizers and SGD on each node are sent to the coordinator node. The coordinating node calculates the aggregated update of parameters from the updates of each node, and exchanges the parameters by sending back to the distributed nodes.

The lack of recent network dataset for intrusion detection is a problem unsolved by cybersecurity researchers. For traditional Internet, the Defense Advanced Research Project Agency's (DARPA's) KDDCUP99 intrusion dataset is frequently used, although it has the problems of redundant, unrealistic, and old records. The nsl-KDD dataset has improved the KDDCUP99 dataset by eliminating its shortcomings, but still, it is not an up-to-date dataset. We consider two datasets used for binary and multi-class classifications to show the performance of our model on different levels of complex data.

The first dataset is the ISCX intrusion dataset [15], which exists in pcap format. The dataset consists of network traces collected over seven days of a week, including full packet payload in pcap format, and hence is modifiable, extensible, and reproducible. The ISCX2012 data set collected on the first and fifth day has been used for training and testing LSTM models in the first case study. The pcap files of the dataset have been transformed into fields or features using Bro IDS, which are fed to the embedding layer of the LSTM network. The dataset has 440,991 normal and 71,617 DoS attack traffic distributions.

The second case study uses a reduced version of the public dataset from [14]. The data were collected from various devices, mobile phones, laptops, and smart TV using Wi-Fi APs. The devices were chosen to show mobility in connections. The data collected was carried out by launching 15 common attacks using various tools in a wireless environment categorized under normal (1,633,190 train and 530,785 test instances), flooding (4848 train and 8097 test instances), injection (65,379 train and 16,682 test instances), and impersonation attacks (48,522 train and 20,079 test instances). As fog-to-things communication is mostly wireless in nature, and around 70 percent [14] of the wireless communications use Wi-Fi connection, it is evident that this dataset and its attack categories could appropriately describe the fog-to-things environment.

Experimentation, Evaluation, and Results

The experiments have been conducted on 64 GB RAM and 16 core processors of nectar cloud. A Keras on Theano package has been used for implementing DL functions of LSTM networks, and Apache Spark has provided a distributed computing platform. We have used two learning approaches: unsupervised and supervised. The training phase has used an unsupervised mechanism of feature learning, while the detection phase employs a supervised scheme. The features extracted during training are mapped against test features during detection. The raw inputs of our datasets were used for the embedding layer of LSTM. As shown in Fig. 4, the training phase employs self-taught, unsupervised learning to extract features. The test phase provides features and attack categories for tuning the learning process. The experimentation has employed 30 embedding layers, 10 middle layers, and a sigmoid layer for the ISCX dataset. The model employed 128 batch sizes in 15 epochs and trained with dropout to avoid the overheating problem.

The mode was evaluated against classical ML-based logistic regression/softmax in both datasets. Evaluation metrics such as accuracy, precision, recall, and others have been used for performance measurement. Accuracy is the ratio of true detection over the total instances in the data. Precision is a measure of relevancy of obtaining results, while recall, on the other hand, is a measure of how many relevant results are returned. In an attack detection system, low false positives and false negatives are desirable evaluation metrics as they indicate high relevancy. Thus, the area under the precision-recall curve is an important measure of model performance.

In the binary classification on the ISCX dataset, the training and validation losses have converged to the same point after 10 of 15 epochs (Fig. 4a). As shown in Table 2a, the overall accuracy (99.91 percent) of the deep model is over 9 percent higher than the shallow model (90 percent). Moreover, the precision-recall curve (Fig. 4b) of the LSTM model is higher and greater than that of logistic regression, which indicates that the system perfectly classifies attack and normal instances into their respective categories. It is evident that the obtained high values of precision and recall are related to low false positives and false negatives, respectively. Finally, training and evaluation

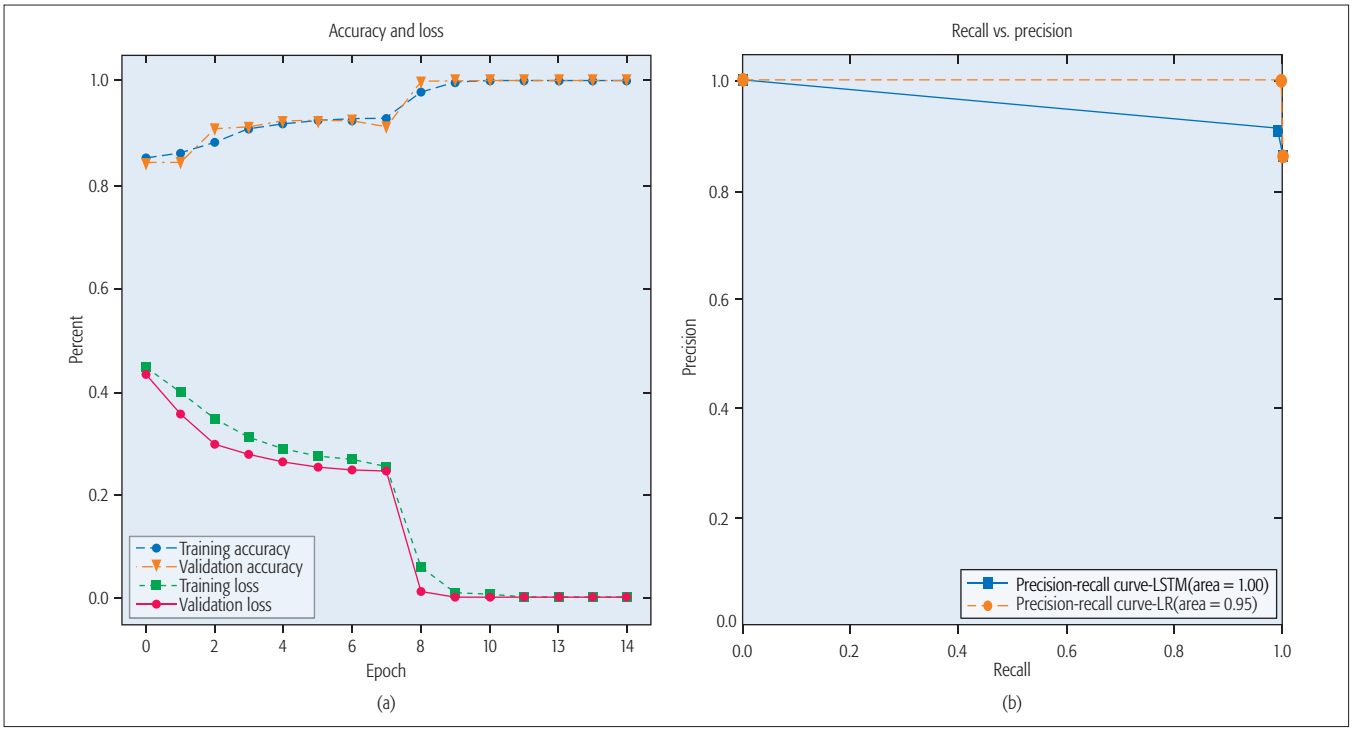

Figure 4. Binary classification on ISCX dataset: a) model performance; b) precision-recall curve.

(a)						
Algorithm	ISCX dataset			AWID dataset		
	Accuracy	Recall	Precision	Accuracy	Recall	Precision
LSTM	99.91	99.96	99.85	98.22	98.9	98.5
LR	90/91	99	89.11	84.87	90	85
(b)						
Algorithm	ISCX dataset			AWID dataset		
	Train instance/s	Test instance/s	Memory (MB)	Train instance/s	Test instance/s	Memory (MB)
LSTM	374.8	80.2	5.6	1220	240	134
LR	54.3	1177.8	286	3023	6230	1356

Table 2. a) Performance results of models; b) running time and memory requirements of the models.

times of the LSTM model have been compared to the LR model, as shown in Table 2b. The average number of instances trained on the LR model is 54.3, while LSTM required 374.6 training instances/s. It was found that the training LSTM network has taken more considerable time than training LR. Moreover, it is also interesting to observe that once the models have been trained, the LR methodology is able to evaluate 1177.8 instances/s, compared to 80.2 instances/s by LSTM. However, the memory requirements of the LR model are almost 50 times those of LSTM due to the complexity of storing the models' parameters. This indicates the compact features and models of DL.

In the multi-classification, we compared the LSTM model with softmax classification. The overall accuracy of the LSTM model (98.85 percent) is superior to the accuracy of softmax (86 percent). It is consistent with the binary classification in performance. The model's accuracy has been challenged by the imbalance of classes and the accuracy of impersonation attacks.

Conclusion and Future Work

We have studied state-of-the-art DL-based cyber-attack detections. Our study has found out that signature-based solutions lack the capability of detecting novel attacks, while classical ML mechanisms lack scalability, robustness, and accuracy. Thus, we have adapted an LSTM network to detect cyber-attacks in distributed fog-to-things communication. The experiments were conducted on ISCX and AWID datasets using keras on theano and Apache Spark. Fog nodes were leveraged for training on local data obtained from IoT devices in the vicinity, and sent the best parameters to a coordinator node for update. The updated parameters are exchanged among fog nodes via coordinator node broadcast. The results have been compared with classical ML algorithms in various evaluation metrics. Apart from eliminating the human loop in the feature extraction process, it has been demonstrated that deep learning is resilient in discriminating attacks from normal traf-

Distributing training data over fog nodes and coordinating those using parameters exchanges through a centralized node increases attack detection accuracy and scalability. Thus, it can effectively address the resource limitations of IoT devices and the scalability problems of the cloud by providing distributed intelligence between the cloud and the things.

fic. The AWID dataset has shown a similar trend to the ISCX dataset for all metrics. It has been observed that distributing training data over fog nodes and coordinating those using parameters exchanges through a centralized node increases attack detection accuracy and scalability. Thus, it can effectively address the resource limitations of IoT devices and the scalability problems of the cloud by providing distributed intelligence between the cloud and the things. It will be worth looking at other deep learning algorithms on the same and more complex datasets in the future.

References

[1] C. Yin *et al.*, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, Oct. 2017, pp. 21954–61.

[2] A.Diro and N.Chilamkurti, "Distributed Attack Detection Scheme Using Deep Learning Approach for the Internet of Things," *Future Generation Computer Systems*, vol. 82, May 2018, pp. 761–68.

[3] N. Shone *et al.*, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Trans. Emerging Topics in Computational Intelligence*, vol. 2, no. 1, Feb. 2018, pp. 41–50.

[4] P. Malhotra *et al.*, "Long Short Term Memory Networks for Anomaly Detection in Time Series," *Proc. Euro. Symp. Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, Apr. 22–24, 2015.

[5] A. Diro *et al.*, "Lightweight Cybersecurity Schemes Using Elliptic Curve Cryptography in Publish-Subscribe Fog Computing," *Mobile Net. Appl.*, vol. 22, no. 112, Oct. 2017, pp. 1–11.

[6] A. Abeshu and N. Chilamkurti, "Deep Learning: The Frontier for Distributed Attack Detection in Fog-to-Things Computing," *IEEE Commun. Mag.*, vol. 56, no. 2, Feb. 2018, pp. 169–75.

[7] J. Kim *et al.*, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," *Int'l. Conf. Platform Tech. and Service*, Jeju, 15- 17 Feb., 2016, pp. 1–5.

[8] R. Staudemeyer, "Applying Long Short-Term Memory Recurrent Neural Networks to Intrusion Detection," *South African Computer J.*, vol. 56, no. 1, 2015, pp.136–54.

[9] L. Bontemps *et al.*, "Collective Anomaly Detection Based on Long Short-Term Memory Recurrent Neural Networks," *Int'l. Conf. Future Data and Security Engineering*, Vietnam, Springer, Nov. 23–35, 2016, pp. 141–52.

[10] J. Woodbridge *et al.*, "Predicting Domain Generation Algorithms with Long Short-Term Memory Networks," 2016, arXiv preprint arXiv:1611.00791, accessed Oct. 15, 2017.

[11]. A. Bahnsen *et al.*, "Classifying Phishing URLs Using Recurrent Neural Networks," *2017 APWG Symp. Electronic Crime Research*, Scottsdale, AZ, 25–27 Apr., 2017, pp. 1–8.

[12] J. Saxe and K. Berlin, " eXpose: A Character-Level Convolutional Neural Network with Embeddings for Detecting Malicious URLs, File Paths and Registry Keys," 2017, eprint arXiv:1702.08568, accessed Mar. 12, 2017.

[13] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS Attack via Deep Learning," *2017 IEEE Int'l. Conf. Smart Computing*, Hong Kong, 29–31 May, 2017, pp. 1–8.

[14] C. Kolias *et al.*, "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset," *IEEE Commun. Surveys & Tutorials*, vol. 18, no. 1, 2016, pp. 184–208.

[15] A. Shiravi *et al.*, "Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection," *Computers & Security J.*, vol. 31, no. 3, May 2012, pp. 357–74.

Biographies

Abebe Abeshu Diro received his M.Sc. degree in computer science from Addis Ababa University, Ethiopia, in 2010. He worked at Wollega University as a director of ICT Development, and lecturer in the Computer Science Department. He is currently a Ph.D. candidate in the Department of IT Computer Science and IT, La Trobe University, Australia. His research interests include software defined networking, the Internet of Things, cybersecuirity, advanced networking, machine learning, and big data.

Naveen Chilamkurti (n.chilamkurti@latrobe.edu.au) is currently cybersecurity program coordinator, Computer Science and Information Technology, La Trobe University. He obtained his Ph.D. degree from La Trobe University. His current research areas include intelligent transport systems (ITS), smart grid computing, vehicular communications, vehicular cloud, cyber security, wireless multimedia, wireless sensor networks, and mobile security.