Future Generation Computer Systems 82 (2018) 761–768

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Distributed attack detection scheme using deep learning approach for Internet of Things

Abebe Abeshu Diro, Naveen Chilamkurti *

Department of Computer Science and IT, La Trobe University, Melbourne, Australia

HIGHLIGHTS

- Deep learning has been proposed for cyber-attack detection in IoT using fog ecosystem.
- We demonstrated that distributed attack detection at fog level is more scalable than centralized cloud for IoT applications.
- It has also been shown that deep models have excelled shallow machine learning models in cyber-attack detection in accuracy.
- In the future, other datasets and algorithms as well as network payload data will be investigated for comparisons and further enhancements.

ARTICLE INFO

Article history:
Received 1 May 2017
Received in revised form 12 July 2017
Accepted 23 August 2017
Available online 1 September 2017

Keywords:
Cybersecurity
Deep learning
Internet of Things
Fog networks
Smart cities

ABSTRACT

Cybersecurity continues to be a serious issue for any sector in the cyberspace as the number of security breaches is increasing from time to time. It is known that thousands of zero-day attacks are continuously emerging because of the addition of various protocols mainly from Internet of Things (IoT). Most of these attacks are small variants of previously known cyber-attacks. This indicates that even advanced mechanisms such as traditional machine learning systems face difficulty of detecting these small mutants of attacks over time. On the other hand, the success of deep learning (DL) in various big data fields has drawn several interests in cybersecurity fields. The application of DL has been practical because of the improvement in CPU and neural network algorithms aspects. The use of DL for attack detection in the cyberspace could be a resilient mechanism to small mutations or novel attacks because of its high-level feature extraction capability. The self-taught and compression capabilities of deep learning architectures are key mechanisms for hidden pattern discovery from the training data so that attacks are discriminated from benign traffic. This research is aimed at adopting a new approach, deep learning, to cybersecurity to enable the detection of attacks in social internet of things. The performance of the deep model is compared against traditional machine learning approach, and distributed attack detection is evaluated against the centralized detection system. The experiments have shown that our distributed attack detection system is superior to centralized detection systems using deep learning model. It has also been demonstrated that the deep model is more effective in attack detection than its shallow counter parts.

1. Introduction

As an emerging technology breakthroughs, IoT has enabled the collection, processing and communication of data in smart applications [1]. These novel features have attracted city designers and health professionals as IoT is gaining a massive application in the edge of networks for real time applications such as eHealth and smart cities [2]. However, the growth in the number, and sophistication of unknown cyber-attacks have cast a shadow on the adoption of these smart services. This emanates from the fact that the distribution and heterogeneity of IoT applications/services make the security of IoT complex and challenging [1,3]. In addition, attack detections in IoT is radically different from the existing mechanisms because of the special service requirements of IoT which cannot be satisfied by the centralized cloud: low latency, resource limitations, distribution, scalability and mobility, to mention a few [4]. This means that neither cloud nor standalone attack detection solutions solve the security problems of IoT. Because of this, a currently emerged novel distributed intelligence, known as fog computing, should be investigated for bridging the gap. Fog computing is the extension of cloud computing towards the network edge to enable cloud-things service continuum. It is based on the principle that data processing and communication should be served closer to the data sources [5]. The principle helps in

* Corresponding author.
E-mail addresses: a.diro@latrobe.edu.au (A.A. Diro), n.chilamkurti@latrobe.edu.au (N. Chilamkurti).

http://dx.doi.org/10.1016/j.future.2017.08.043
0167-739X/

alleviating the problem of resource scarcity in IoT as costly storage, computation and control, and networking might be offloaded to nearby fog nodes. This in turn increases the effectiveness and efficiency of smart applications. Like any services, security mechanisms in IoT could be implemented and deployed at fog layer level, having fog nodes as a proxy, to offload expensive storage and computations from IoT devices. Thus, fog nodes provide a unique opportunity for IoT in deploying distributed and collaborative security mechanisms.

Though fog computing architecture can offer the necessary service requirements and distributed resources, robust security mechanisms are also needed resources to protect IoT devices. As preventive security schemes are always with the shortcomings design and implementation flaws, detective mechanisms such as attack detection are inevitable [6]. Attack detections can be either signature based or anomaly based schemes. The signature based solution matches the incoming traffic against the already known attack types in the database while anomaly based scheme caters for attack detection as a behavioral deviation from normal traffic. The former approach has been used widely because of its high accuracy of detection and low false alarm rate, but criticized for its incapability to capture novel attacks. Anomaly detection, on the other hand, detects new attacks though it lacks high accuracy. In both approaches, classical machine learning has been used extensively [7]. With the ever increasing in the attacker's power and resources, traditional machine learning algorithms are incapable of detecting complex cyber breaches. Most of these attacks are the small variants of previously known cyber-attacks (around 99% mutations). It is evident that even the so called novel attacks (1%) depend on the previous logics and concepts [8]. This means that traditional machine learning systems fail to recognize this small mutation as it cannot extract abstract features to distinguish novel attacks or mutants from benign. The success of deep learning in big data areas can be adopted to combat cyber threats because mutations of attacks are like small changes in, for instance, image pixels. It means that deep learning in security learns the true face (attack or legitimate) of cyber data on even small variations or changes, indicating the resiliency of deep learning to small changes in network data by creating high level invariant representations of the training data. Though the application of DL has been mainly confined to big data areas, the recent results obtained on traffic classification, and intrusion detection systems in [9–11] indicate that it could have a novel application in identification of cybersecurity attacks.

Deep learning (DL) has been the breakthroughs of artificial intelligence tasks in the fields of image processing, pattern recognition and computer vision. Deep networks have obtained a momentum of unprecedented improvement in accuracy of classification and predictions in these complex tasks. Deep learning is inspired by the human brain's ability to learn from experience instinctively. Like our brain's capability of processing raw data derived from our neuron inputs and learning the high-level features on its own, deep learning enables raw data to be fed into deep neural network, which learns to classify the instances on which it has been trained [12,13]. DL has been improved over classical machine learning usually due to the current development in both hardware resources such as GPU, and powerful algorithms like deep neural networks. The massive generation of training data has also a tremendous contribution for the current success of deep learning as it has been witnessed in giant companies such as Google and Facebook [14,15]. The main benefit of deep learning is the absence of manual feature engineering, unsupervised pre-training and compression capabilities which enable the application of deep learning feasible even in resource constraint networks [16]. It means that the capability of DL to self-learning results in higher accuracy and faster processing. This research is aimed at adopting a novel distributed attack detection using deep learning to enable the detection of existing or novel attacks in IoT.

The contributions of our research area:

- To design and implement deep learning based distributed attack detection mechanism, which reflects the underlying distribution features of IoT
- To demonstrate the effectiveness of deep learning in attack detection systems in comparison to traditional machine learning in distributed IoT applications
- To compare the performance of parallel and distributed network attack detection scheme using parameters sharing with a centralized approach without parameters sharing in IoT.

2. Related work

Though research works in the application of deep learning have currently flourished in domains like pattern recognition, image processing and text processing, there are a few promising researches works around cybersecurity using deep learning approach.

One of the applications of deep learning in cybersecurity is the work of [9] on NSL-KDD dataset. This work has used self-taught deep learning scheme in which unsupervised feature learning has been employed on training data using sparse-auto encoder. The learnt features were applied to the labeled test dataset for classification into attack and normal. The authors used n-fold cross-validation technique for performance evaluation, and the obtained result seems reasonable. This research work is like ours in terms of feature learning though it considers centralized system while our approach is distributed and parallel detection system used for fog-to-things computing. The other relevant work is the application of AutoEncoders for anomaly detection in [17], in which normal network profile has been learned by autoencoders through nonlinear feature reduction. In their study, the authors demonstrated that normal records in the test dataset have small reconstruction error while it produced a large reconstruction error for anomalous records in the same dataset. Intrusion detection using a deep learning approach has also been applied in vehicular security in [10]. The research has demonstrated that deep belief networks (DBN) based unsupervised pre-training could enhance the intrusion detection accuracy. Although the work is novel in its approach, the artificial data used, and its centralized approach might limit its practicality in fog networks. Another research of this category has been conducted by [18]. The authors have proposed IDS which adaptively detect anomalies using AutoEncoders on artificial data. Both anomaly detection papers considered artificial data cases which do not reflect the malicious and normal behaviors of real time networks. Apart from that, they adopted a centralized approach which is impractical for distributed applications such as social internet of things in smart city networks.

Deep learning approach has also been applied by [11] for malicious code detection by using AutoEncoders for feature extraction and Deep Belief Networks (DBN) as a classifier for detection. The article has shown that the hybrid mechanism is more accurate and efficient in time than using a single DBN. It strengthens that deep networks are better than shallow ones in cyber-attack detection. The main limitation of this research is that the dataset should have been more recent to come up with the conclusion. Nevertheless, the research is significantly different from ours since it does not handle the distributed training and sharing of updated parameters. The other paper which has investigated deep learning scheme for malicious code detection is [19]. It has applied denoising AutoEncoder for deeper features learning to identify malicious JavaScript

Table 1
Attack categories of Fog ecosystem.

Category	Attack type and description
Probe	• Satan—probing of a network for some well-known weaknesses • Ipsweep—pinging of multiple hosts to reveal the target's IP • Portsweep—scanning of ports to discover services on a host • Nmap—various means of network mapping
R2L	• Warezclient—downloading illegal software uploaded previously by the warezmaster • guess_passwd—guessing password over telnet • warezmaster—uploading illegal software (warez) on FTP server exploiting wrong write permissions • imap—illegal access of local user account using vulnerabilities • ftp_write—creating.rhost file in anonymous FTP to obtain local login. • Multihop–multi-day scenario where a user breaks into a system • phf—CGI script enabling to execute arbitrary commands on a machine with a misconfigured web server. • spy—breaking into system via vulnerabilities to discover important information
U2R	• buffer_overflow—the ffbconfig UNIX system command causes buffer flow leads to root shell • rootkit—enables to access admin level • loadmodule—gaining root shell by resetting IFS • perl—creating root shell by perl attack which sets the user id to root
DoS	• smurf—flooding of ICMP echo reply • Neptune—flooding of SYN on port(s) • Back—requesting of a URL having many backslashes from a webserver • Teardrop—causing system reboot or crash using mis-fragmented UDP packets • Pod—pinging with malformed packets causing reboot or crash • Land—ending UDP packet having the same source and destination address to remote host

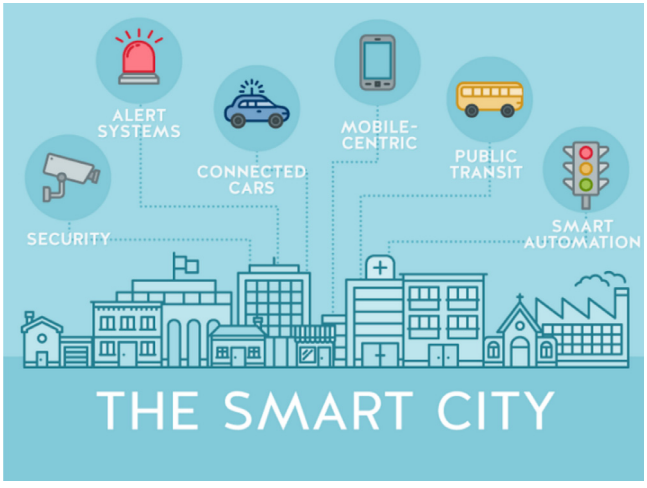

Fig. 1. Social internet of things: Components of smart city.

code from normal code. The result has produced promising accuracy in the best-case scenario. Although the approach is effective in web applications, it can be hardly applied to distributed IoT/Fog systems. Our model is novel as it enables parallel training and parameters sharing by local fog nodes, and detects network attacks in distributed fog-to-things networks using deep learning approach.

3. Cybersecurity in social IoT

The advancements in technologies of hardware have enabled a massive number of IoT devices to be connected to the Internet. Smart city applications are by far the quickest and deeply affected areas of public services by social internet of things as this technological breakthrough is helping cities to manage effectively infrastructures such as water, power, transport, and so on. Typically, the integration of social IoTs and ICT for innovative, smart city design is to create a data-driven approach to public service delivery, infrastructure and public planning. In general, the social IoT applications in components in the smart city are depicted in Fig. 1. However, this massive connection of IoT devices as a data collection and distribution platform in the emergence of smart cities could bring about novel or variant attacks which can cause the loss of multi-million dollars and human life [20].

Though the attacks in IoT seems the same as in traditional Internet, the scale and simplicity of attack targets are larger for IoT with limited protection. As the recent survey [7] shows, DoS attacks are the most frequently associated attack types with IoT/Fog networks in social internet of things such as smart cities. The IoT ecosystem consists of a massive number of smart things distributed across a given geographical area, such as smart city. Millions of users are connected to social services via IoT, taking advantage of it for private and public services. The interconnectivity of these numbers of things, however, makes a fertile target for malicious adversaries who can exhaust their resources and launch DoS attacks. A DoS attack causes the denial of service for legitimate users or nodes by a single host (DoS attack) or coordinated attackers (DDoS attack) [21–23].

However, remote access attacks using backdoors could be major threats as they can escalate to root access attacks. These attacks leave a certain patterns and characteristics in network traffic, which might affect the way learning algorithms will be able to distinguish between an attack and benign. For instance, DoS attacks are usually known by overwhelming a single node from multiple sources. Table 1 shows the common attack categories [24].

4. Overview of deep learning

Deep Learning has been the state of the art for training stability and generalization, and achieved significant scalability on big data. It extracts complex and nonlinear hierarchical features of training data of high dimension to build a model which transforms inputs to outputs (e.g. classification). Multi-layer deep networks are the most prevalent forms of deep learning algorithms. The output of each previous layer and a bias are computed by a nonlinear activation function f to form weighted inputs W_n for the next layer n of a neural network, i.e $a_n = f(W_n a_{k-1})$) [25]. Activation functions are listed in Table 2. Given a set of unlabeled training data $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots\}$, deep learning algorithms usually set output values to be either equal or less than inputs. The cost functions to be optimized in deep models during deep feature extraction are generally loss functions. In Eq. (1), loss function is shown in which the first term is the reconstruction error specified using the mean of sum-of-square error terms for k instances of training data, and

Table 2
Activation functions.

Function	Formula	Range
Tanh	$f(\alpha) = \frac{e^{\alpha}-e^{-\alpha}}{e^{\alpha}+e^{-\alpha}}$	$f(\cdot) \in [-1, 1]$
Rectified linear	$f(\alpha) = \max(0, \alpha)$	$f(\cdot) \in \mathbb{R}_+$
Maxout	$f(\alpha_1, f\alpha_2) = \max(\alpha_1, \alpha_2)$	$f(\cdot) \in \mathbb{R}$

the second term is a regularization term which is used for avoiding over-fitting problem in training.

$$J(W, b) = \frac{1}{2k}\sum_{k=0}^{k}\left(\left|\left|x^{(i)} - \hat{x}^{(i)}\right|\right|\right)^2 + \frac{\lambda}{2}\sum_{l=1}^{nl-1}\sum_{i=1}^{sl}\sum_{j=1}^{sl+1}\left(W_{ji}^{(l)}\right)^2 \quad (1)$$

where nl represents the number of layers and sl is the number of nodes in each layer.

The mechanism of minimizing the loss function $L(W, B|j)$ is a stochastic gradient descent (SGD, where the gradient $\nabla L(W, b|J)$ is a standard gradient computed via backpropagation using constant α as a learning rate. The final parameters W, B are obtained by averaging. Eq. (2) shows the iteration of standard gradient descent on updates of weight W and bias b using sample i until the convergence is obtained. As the gradient descent of parameters over the whole available data (batch) is not efficient, computing gradient over mini-batch (sampling subset) simplifies the learning process [15].

$$W_{ji} := W_{ji} - \alpha\frac{\partial L(W, b|j)}{\partial W_{ji}}$$
$$b_{ji} := W_{ji} - \alpha\frac{\partial L(W, b|j)}{\partial b_{ji}}. \quad (2)$$

In training process, as the layer increases, abstraction of features increases towards the answers of the model. The activation function and weight matrices determine the abstraction nature at every layer of the network, but it is challenging to enable the deep learning model to automatically learn training parameters that meet the accuracy objective of the deep network. The parameters of training are usually learnt through gradient descent, which is a nonlinear optimization problem. Gradient descent is initiated randomly by setting a set of deep network parameters, but it is updated at each step to decrease the gradient by computing gradient descent of nonlinear function being optimized. The output of this repetition yields the optimization of the algorithm to a local optimum.

For a known number of classes, softmax is employed at the end of neural networks as activation function. Having p classes in the dataset and deep network inputs x, softmax assumes that the probability that $P(y = p|x)$ for each value of $p = 1, \ldots, P$ could be estimated as it outputs a K-dimensional vector summed to 1. In other words, our estimate $h_\theta(x)$ take the following form [26]:

$$\mathrm{h}_\theta(x) = \begin{bmatrix} P(y=1|x;\theta) \\ P(y=2|x;\theta) \\ \vdots \\ P(y=P|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{p}\exp\left(\theta^{(j)T}x\right)}\begin{bmatrix} \exp\left(\theta^{(1)T}x\right) \\ \exp\left(\theta^{(2)T}x\right) \\ \vdots \\ \exp\left(\theta^{(p)T}x\right) \end{bmatrix}$$

where $\theta(1), \theta(2), \ldots, \theta(K) \in R^n$ are the parameters of our model, and the term $\frac{1}{\sum_{j=1}^{p}\exp\left(\theta^{(j)T}x\right)}$ is normalization of distribution. The cost function of softmax will be given by:

$$J(\theta) = -\left[\sum_{i=1}^{m}\sum_{p=1}^{p} 1\left\{y^{(i)} = p\right\}\log\frac{\exp\left(\theta^{(p)T}x^{(i)}\right)}{\sum_{j=1}^{p}\exp\left(\theta^{(j)T}x^{(i)}\right)}\right].$$

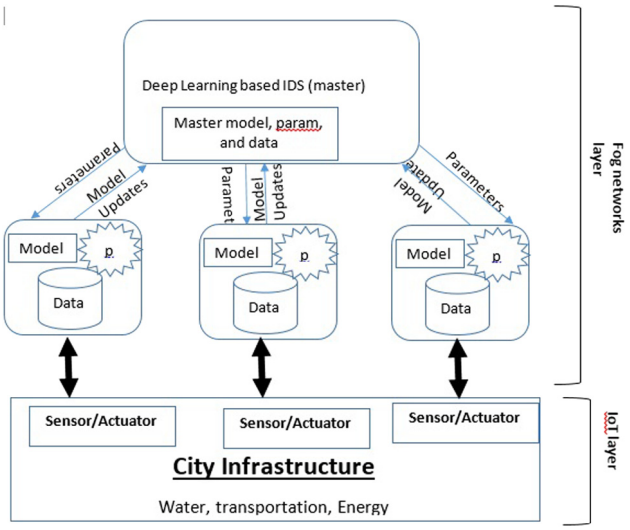

Fig. 2. Distributed attack detection architecture for Fog-to-things networks.

5. Our approach

The fog nodes are responsible for training models and hosting attack detection systems at the edge of the distributed fog network since they are closer to the smart infrastructures supported by social internet of things. The coordinating master node should be in place for collaborative parameter sharing and optimization. In addition to giving the autonomy of local attack detection using local training and parameter optimization, the benefits of this approach are the acceleration of data training near to the source and the gain of updated parameters from neighbors. The master node updates the parameters of each cooperative node, and propagates the resulting update back to the worker nodes. This plays a significant role in offloading storage and computational overheads of models, data and parameters from IoTs while it provides fast response time. The centralized training and optimization approach could be extended to distributed networks by distributing SGD. Fig. 2 shows a general architecture of our distributed and parallel attack detection system in fog-to-things computing.

The outputs of model training on distributed fog nodes are attack detection models and their associated local learning parameters. These local parameters are sent to coordinating fog node for global update and re-propagation. This sharing scheme results in better learning as it enables to share best parameters and avoids local overfitting.

6. Evaluation

6.1. Dataset, algorithm and metrics

KDDCUP99 [27], ISCX [28] and NSL-KDD [24] are the most commonly used datasets in the intrusion detection research. We used NSL-KDD intrusion dataset which is available in csv format for model validation and evaluations. The NSL-KDD intrusion dataset not only reflects the traffic compositions and intrusions, but are also it is modifiable, extensible, and reproducible. The dataset composes of the attacks shown in Table 1, and identified as a key attack in IoT/Fog computing [1–5]. Table 3 shows sample records of NSL-KDD dataset.

The original dataset consists of 125,973 records of train and 22,544 records of test, each with 41 features such as duration,

Table 3
Snapshot of records in NSL-KDD dataset.

0	tcp	ftp_data	SF	491	0	…
0	udp	other	SF	146	0	…
0	icmp	ecr_i	SF	1480	0	…
0	tcp	http	SF	232	8153	…
0	tcp	http	SF	199	420	…
15 159	tcp	ftp	SF	350	1185	…
0	tcp	private	S0	0	0	…
315	udp	other	SF	146	105	…
240	tcp	http	SF	328	275	…
0	tcp	private	S0	0	0	…
0	tcp	private	REJ	0	0	…
5607	udp	other	SF	147	105	…

Table 4(a)
Traffic distribution of NSL-KDD in 2-class.

Traffic	Training	Test
Normal	67 343	9711
Attack	58 630	12 833
Total	**125 973**	**22 544**

Table 4(b)
Traffic distribution of NSL-KDD in multi-class.

Traffic	Training	Test
Normal	67 343	9711
DoS	45 927	7458
Probe	11 656	2754
R2L	995	2421
U2R	52	200
Total	**125 973**	**22 544**

Table 5
The encoded form of our dataset.

features	category
[0.0,491.0,0.0,0....	1
[0.0,146.0,0.0,0....	1
[0.0,0.0,0.0,0.0,...	0
[0.0,232.0,8153.0...	1

protocol, service, flag, source bytes, destination bytes, etc. The traffic distribution of NSL-KDD dataset is shown as in Tables 4(a) and (b).

Before training the network, categorical features have been encoded into discrete features using 1-to-*n* encoding technique. Because of encoding, we obtained 123 input features and 1 label, as shown in Table 5. For our experiment, we have taken the dataset in 2-class (normal vs attack) and 4-class (normal, DoS, Probe, R2L.U2R). The minority class U2R is merged to R2L to form a class of R2L.U2R.

The system uses the same technique of preprocessing for both training and test. At this step, the data is ready for training and testing. Suppose D_n are data across n nodes in the fog ecosystem, and W_n, b_n are parameters their local parameters , each node having local data D_{na} subset of D_n as samples per iteration. Each node runs data training on deep networks in parallel way, but asynchronously exchange learned parameters with the coordinating node. The coordinating node broadcasts the update regularly. The following algorithm shows distributed training on each local fog node while exchanging updated parameters with neighbor nodes via coordinating fog node.

Algorithm 1: local training and parameter exchange

1. Recieve initial or update of W_{ji} *and* b_{ji} from master to workers
2. For node n in the network, do in parallel:
   - Get local training traffic sample i ∈ D_{na} from local D_n
   - Execute SGD on local traffic, and update $w_{ji} \in W_n$, biases $b_{ji} \in b_n$

$$W_{ji} := W_{ji} - \alpha \frac{\partial L(W, b|j)}{\partial W_{ji}} \quad (1)$$

$$b_{ji} := W_{ji} - \alpha \frac{\partial L(W, b|j)}{\partial b_{ji}} \quad (2)$$

   - Compute ΔW_{ji} and Δb_{ji} and send to master node
3. Compute $W_{ji}, b_{ji} = W_{ji} + \Delta W_{ji}, b_{ji} + \Delta b_{ji}$
4. Repeat (1)

In the training and testing phase, Apache Spark [29] has been used for distributed and parallel processing of SGD asynchronously, while Keras on Theano package [30] was employed for deep learning. The most important evaluation metrics for attack detection such as accuracy, the detection rate (DR) and false alarm rate (FAR) were chosen for comparison between deep and shallow models. However, performance metrics such as precision, recall and F1 Measure has been added a comparison between individual classes in the deep learning model DR denotes ratio of intrusion instances detected by the model, while FAR represents the ratio of misclassified normal instances. Accuracy is the percentage of true detection over total data instances. Recall indicates how many of the attacks does the model return, while precision represents how many of the returned attacks are correct. F1 Measure provides the harmonic average of precision and recall [31]. The mathematical representation of these metrics can be derived from confusion matrix as:

	Predicted:NORMAL	Predicted:ATTACK
Actual:NORMAL	TN	FP
Actual:ATTACK	FN	TP

$$ACC = \frac{TP + TN}{(TP + TN + FP + FN)}, \; DR = \frac{TP}{(TP + FN)}, \; FAR = \frac{FP}{(TN + FP)} \quad (3)$$

$$Precision = \frac{TP}{(TP + FP)}, \; Recall = \frac{TP}{(TP + FN)}, \; F1Measure = \frac{2TP}{(2TP + FP + FN)} \quad (4)$$

where, TP: true positive, TN: true negative, FP: false positive, FN: false negative

7. Experimental environment

As a first attempt towards exploring the performance of our model, we used the 2-class (normal and attack) and 4-class (normal, DoS, Probe, R2L.U2R) categories. In performance measure, unseen test data are chosen to represent zero-day attack detections. Our experiment has two objectives. The first one is to compare the result of our distributed attack detection with a centralized system. This experiment has been conducted by deploying the deep learning model on a single node for centralized system, and multiple coordinated nodes for distributed attack detection. To test the performance of parallelism and distribution, as a benchmark for our detection scheme, we varied the number of machines used for training the network as a function of training accuracy. The

Table 6
Accuracy of deep model (DM) and shallow model (SM).

Model type	2-class			4-class		
	Accuracy (%)	DR (%)	FAR (%)	Accuracy (%)	DR (%)	FAR (%)
DM	99.20	99.27	0.85	98.27	96.5	2.57
SM	95.22	97.50	6.57	96.75	93.66	4.97

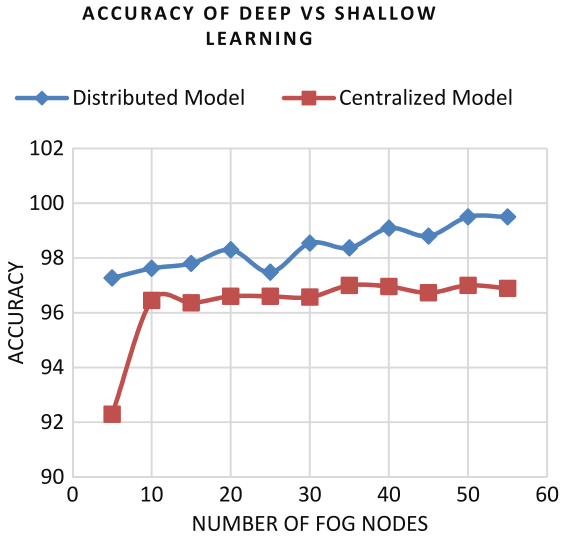

Fig. 3. Accuracy comparison of distributed and centralized models.

Table 7(a)
Performance of 2-class.

Model type	Class	Precision (%)	Recall (%)	F1 Measure (%)
Deep model	Normal	99.36	99.15	99.26
	Attack	99.02	99.27	99.14
Shallow model	Normal	97.95	93.43	95.65
	Attack	92.1	97.50	94.72

Table 7(b)
Performances of 4-class.

Model type	Class	Precision (%)	Recall (%)	F1 Measure (%)
Deep model	Normal	99.52	97.43	98.47
	DoS	97	99.5	98.22
	Probe	98.56	99	98.78
	R2L.U2R	71	91	80
Shallow model	Normal	99.35	95	97
	DoS	96.55	99	97.77
	Probe	87.44	99.48	93
	R2L.U2R	42	82.49	55.55

second one is to evaluate the effectiveness of deep learning against shallow learning algorithms for attack detection in IoT. The deep learning system, after hyper-parameter optimizations, has used 123 input features, 150 first layer neurons, 120 s layer neurons, 50 third layer neurons and the last softmax layer with neurons equal to the number of classes. The model has various batch sizes in 50 epochs, and trained with dropout to avoid the overheating problem.

8. Results and discussions

In the evaluation process, classification accuracy and other metrics were used to show the effectiveness of our scheme compared to shallow models in distributed IoT at fog level. The comparison of distributed training to centralized approach in accuracy is also one of our evaluation criteria. Table 5 compares the accuracy of the deep and shallow models, while Fig. 3 shows the accuracy difference between centralization and distribution.

The experiment result has demonstrated double standards. The first one is that the distributed model has a better performance than the centralized model. As it can be seen from Fig. 3, with the number of increased nodes in the distributed network of Fog systems, the overall accuracy of detection increased from around 96% to over 99%. The detection rate in Table 6 also exhibits that deep learning is better than classic machine learning for both binary and multi-classes. This shows that distributing attack detection functions across worker fog nodes is a key mechanism for attack detection in social IoT systems such as a smart city which needs real time detection. The increase in accuracy on distributed scheme could be because of collaborative sharing of learning parameters which avoids overfitting of local parameters, and hence, contributes to the accuracies of each other. On the other hand, the accuracy of the deep model is greater than that of shallow model, as shown in Table 6. In addition, Table 6 shows the false alarm rate of the deep model, 0.85% is much less than that of machine learning model (6.57%). As shown in Tables 7(a) and (b), the performance of deep learning is better than the normal machine learning model for each class of attack. For instance, the recall of deep model is 99.27%, while the traditional model has a recall of 97.50% for a binary classification. Similarly, the average recall of DM is 96.5% whereas SM has scored average recall of 93.66% in multi-classification. However, Fig. 4 shows that deep learning takes longer learning time than traditional machine learning algorithms while the detection rates (Fig. 5) of the both algorithms are significantly the same. It is expected that deep networks consume larger time in training because of the size of parameters used in learning. The main issue for attack detection systems focuses more on the detection speed than the learning speed. Thus, this indicates that deep learning has a huge potential to transform the direction of cybersecurity as attack detection in distributed environments such as IoT/Fog systems has indicated a promising result.

9. Conclusion and future work

We proposed a distributed deep learning based IoT/Fog network attack detection system. The experiment has shown the successful adoption of artificial intelligence to cybersecurity, and designed and implemented the system for attack detection in distributed architecture of IoT applications such as smart cities. The evaluation process has employed accuracy, the detection rate, false alarm rate, etc. as performance metrics to show the effectiveness of deep models over shallow models. The experiment has demonstrated that distributed attack detection can better detect cyber-attacks than centralized algorithms because of the sharing of parameters which can avoid local minima in training. It has also been demonstrated that our deep model has excelled the traditional machine learning systems such as softmax for the network data classification into normal/attack when evaluated on already unseen test data. In the future, we will compare distributed deep learning IDS for on another dataset and different traditional machine learning algorithms such as SVM, decision trees and other neural networks.

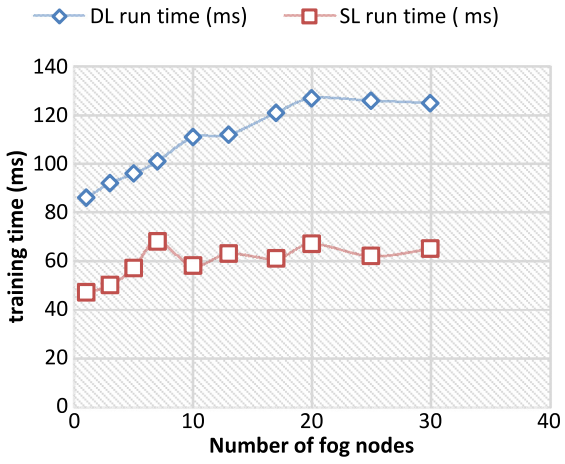

Fig. 4. Comparison between DL and SL in training time.

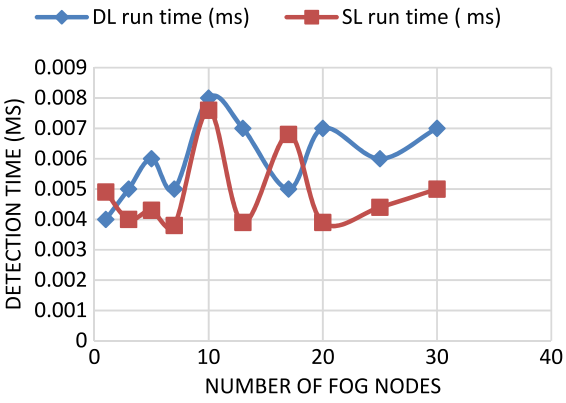

Fig. 5. Comparison between DL and SL in detection time.

Additionally, network payload data, will be investigated to detect intrusion as it might provide a crucial pattern for differentiation.

Acknowledgment

This work was supported by La Trobe University's research enhancement scheme fund.

References

[1] Securing the Internet of Things: A Proposed Framework, 2016, http://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html.
[2] M. Ibrahim, Octopus: An edge-fog mutual authentication scheme, J. Netw. Secur. 18 (6) (2016).
[3] I. Stojemovic, S. Wen, The fog computing paradigm: Scenarios and security issues, in: IEEE Federated Conference on Computer Science and Information Systems, 2014.
[4] A. Alrawais, A. Alhothaily, C. Hu, X. Cheng, Fog computing for the internet of things: Security and privacy issues, IEEE Internet Comput. 21 (2) (2017) 34–42.
[5] S. Yi, Z. Qin, Q. Li, Security and privacy issues of fog computing: A survey, in: International Conference on Wireless Algorithms, Systems and Applications, WASA, 2015.
[6] V.L.L. Thing, IEEE 802.11 network anomaly detection and attack classification: A deep learning approach, in: 2017 IEEE Wireless Communications and Networking Conference, WCNC, San Francisco, CA, 2017, pp. 1–6.
[7] C. Kolias, G. Kambourakis, A. Stavrou, S. Gritzalis, Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset, IEEE Commun. Surv. Tutor. 18 (1) (2016) 184–208.
[8] Guy Caspi, Introducing Deep Learning: Boosting Cybersecurity With An Artificial Brain, http://www.darkreading.com/analytics/introducing-deep-learning-boosting-cybersecurity-with-an-artificial-brain/a/d-id/1326824 (last accessed on 1.07.17).
[9] Quamar Niyaz, Weiqing Sun, Ahmad, Y. Javaid, Mansoor. Alam, Deep learning approach for network intrusion detection system, in: ACM 9th EAI International Conference on Bio-inspired Information and Communications Technologies, New York, 2016.
[10] M.-J. Kang, J.-W. Kang, Intrusion detection system using deep neural network for in-vehicle network security, PLoS One 11 (6) (2016) e0155781. http://dx.doi.org/10.1371/journal.pone.0155781.
[11] Y. Li, R. Ma, R. Jiao, A hybrid malicious code detection method based on deep learning, Int. J. Secur. Appl. 9 (2015) 205–216.
[12] Yoshua Bengio, Pascal Lamblin, Greedy layer-wise training of deep networks, in: Advances in neural …Nr. 1, S. 2007, pp. 153–160 — ISBN: 0262195682.
[13] Li Deng, A tutorial survey of architectures, algorithms, and applications for deep learning, in: APSIPA Transactions on Signal and Information Processing Bd. 3, 2014, Nr. January, S. e2 — ISBN: 2048-7703.
[14] Yann Lecun, Bottou Leon, Bengio Yoshua, Haffner Patrick, Gradient based learning applied to document recognition, in: Proceedings of the IEEE Bd. 86 Nr. 11, S. 1998, pp. 2278–2324 — ISBN: 0018-9219.
[15] J. Dean, G.S. Corrado, R. Monga, K. Chen, M. Devin, Q.V. Le, M.Z. Mao, M'A. Ranzato, A. Senior, P. Tucker, K. Yang, A.Y. Ng, Large Scale Distributed Deep Networks.
[16] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, in: Journal of Machine Learning Research Bd. 11 Nr. 3, S. 2010, pp. 3371–3408 — ISBN 1532-4435.
[17] Mayu Sakurada, Takehisa Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: Ashfaqur Rahman, Jeremiah Deng, Jiuyong Li (Eds.), Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, (MLSDA'14), ACM, New York, NY, USA, 2014, p. 4. http://dx.doi.org/10.1145/2689746.2689747. 8 pages.
[18] Chao Wu, Yike Guo, Yajie Ma, AdaptiveAnomaliesDetectionwithDeepNetwork, on COGNITIVE 2015 : The Seventh International Conference on Advanced Cognitive Technologies and Applications, IARIA, 2015.
[19] Y. Wang, W. Cai, P. Wei, A deep learning approach for detecting malicious JavaScript code, Secur. Commun. Netw. 9 (2016) 1520–1534. http://dx.doi.org/10.1002/sec.1441.
[20] Abebe Abeshu Diro, Naveen Chilamkurti, Neeraj Kumar, Lightweight cybersecurity schemes using elliptic curve cryptography in publish-subscribe fog computing, Mob. Netw. Appl. (2017) 1–11.
[21] http://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-30/dos-attacks.html (last access on 14.11.16).
[22] Charalampos Patrikakis, Michalis Masikos, Olga Zouraraki, Distrib. Denial Serv. Attacks Internet Protoc. J. 7 (4) (2004).
[23] Costa Gondim, João José, et al., "A Methodological Approach for Assessing Amplified Reflection Distributed Denial of Service on the Internet of Things" Ed. Muhammad Imran et al., Sensors 16 (11) (2016) p1855.
[24] M. Tavallaee, E. Bagheri, W. Lu, A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: Second IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA, 2009.
[25] Ruslan Salakhutdinov, Geoffrey E. Hinton, Deep boltzmann machines, in: Proceedings of The 12th International Conference on Artificial Intelligence and Statistics, PMLR 5:448-455, 2009.
[26] http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression, last accessed on 31.05.17.
[27] Knowledge discovery in databases DARPA archive, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, last accessed on 31.05.17.
[28] Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, Ali A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Comput. Secur. 31 (3) (2012) 357–374.
[29] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica, Spark: cluster computing with working sets, in: Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, (HotCloud'10), USENIX Association, Berkeley, CA, USA, 2010, p. 10-10.
[30] Keras deep learning P.W.D. Charles Project Title 2013,https://github.com/charlespwd/project-title (last accessed on 30.11.16).
[31] T.A. Tang, L. Mhamdi, D. McLernon, et al., Deep learning approach for network intrusion detection in software defined networking, in: UNSPECIFIED The International Conference on Wireless Networks and Mobile Communications, (WINCOM'16), IEEE, Fez, Morocco, Oct 26–29, 2016.

Abebe Abeshu Diro is currently a Ph.D. candidate in the Department of IT Computer Science and IT, La Trobe University, Australia. He received his M.Sc. degree in Computer Science from Addis Ababa University, Ethiopia in 2010. He worked at Wollega University from 2007 to 2013 as a Director of ICT Development, and Lecturer in Computer Science. His research interests include Software Defined Networking, Internet of Things, Cybersecurity, Advanced Networking, Machine Learning, and Big Data.

Naveen Chilamkurti is currently the Cybersecurity Program Coordinator, Computer Science and Information Technology, La Trobe University, Melbourne, VIC, Australia. 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, Cybersecurity, wireless multimedia, wireless sensor networks, and Mobile security.