ISSN 2347 - 3983
Volume 9. No. 4, April 2021
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter23942021.pdf
https://doi.org/10.30534/ijeter/2021/23942021

Proposed Back Propagation Deep Neural Network for Intrusion Detection in Internet of Things Fog Computing

Hassan Adegbola Afolabi[1], Abdurazzag Aburas[2]
[1]School of Electrical, Electronic and Computer Engineering
University of Kwazulu-Natal South Africa, 219048801@stu.ukzn.ac.za
[2]School of Electrical, Electronic and Computer Engineering
University of Kwazulu-Natal South Africa,aburasa@ukzn.ac.za

ABSTRACT

Internet of things (IoT) is an emerging concept which aims to connect billions of devices with each other anytime regardless of their location. Sadly, these IoT devices do not have enough computing resources to process huge amount of data. Therefore, Cloud computing is relied on to provide these resources. However, cloud computing based architecture fails in applications that demand very low and predictable latency, therefore the need for fog computing which is a new paradigm that is regarded as an extension of cloud computing to provide services between end users and the cloud user. Unfortunately, Fog-IoT is confronted with various security and privacy risks and prone to several cyberattacks which is a serious challenge. The purpose of this work is to present security and privacy threats towards Fog-IoT platform and discuss the security and privacy requirements in fog computing. We then proceed to propose an Intrusion Detection System (IDS) model using Standard Deep Neural Network's Back Propagation algorithm (BP-DNN) to mitigate intrusions that attack Fog-IoT platform. The experimental Dataset for the proposed model is obtained from the Canadian Institute for Cybersecurity 2017 Dataset. Each instance of the attack in the dataset is separated into separate files, which are DoS (Denial of Service), DDoS (Distributed Denial of Service), Web Attack, Brute Force FTP, Brute Force SSH, Heartbleed, Infiltration and Botnet (Bot Network) Attack. The proposed model is trained using a 3-layer BP-DNN

Key words: Back propagation, CIC 2017 datasets, Deep neural network, Fog computing, Internet of things, Intrusion detection systems

1. INTRODUCTION

Internet of Things (IoT) is an emerging concept which aims to connect billions of devices with each other anytime regardless of their location. As the use of IoT rapidly evolves and increases, our daily activities and lifestyle has been revolutionized and improved by this technology and its application has been adopted in several domains such as *Smart Homes* which is one of the most important application of IoT. Equipping our homes and offices with IoT technologies to enable home automation. [1], *Smart Supply Chains* [2], *Smart Cities*, Smart Security [3]. etc. IoT provide users with several technological facilities all under one umbrella and puts a lot of challenges in front of researchers. Other domains and applications of IoT includes Smart Grid, Wearables, Smart Health Care (Digital Health and Telemedicine) *etc*. A report published by Gartner, Inc. forecasts that over 20 billion connected things will be in use globally by 2020 [4]. Therefore, lots of data will be produced and handled by these IoT devices [5].Estimation in trillions of GB. Unfortunately, IoT devices do not have enough computing resources to process huge amount of data. Therefore, cloud computing is usually involved and relied on to solve the drawbacks limiting IoT [6]. It has been recognized as a success factor for IoT applications due to its uses for storage, analysis and processing of data. [7]. However, cloud computing based IoT becomes unsuccessful in applications that demand very low and predictable latency which are geographically distributed, large-scale distributed control systems and/or high-speed mobile [8]. To address these drawbacks, fog computing was proposed by Cisco Systems Inc. in 2012 [9]. It is regarded as an extension of cloud computing to provide services between end users and the cloud user [10]. In fog computing, extremely confidential information would be processed in the fog layer which may lead to many new security issues. Thus, it is unreasonable to trust all the fog nodes. In addition to the unique threats it suffers from due to adoption of fog infrastructure, classical security issues which are inherited from IoT cloud computing is still a challenge. Unfortunately, security techniques in cloud computing cannot be directly applied in fog computing. Hence, Internet of Things fog computing cannot be adopted despite its usefulness without proper security mechanisms in place.

1.1 Security Issues in IoT Fog Computing

Cloud computing platform is vulnerable to be attacked by external hackers due to its centralized computing framework and data storage. However, it is difficult for attackers to gain access to users' data in fog computing because the collected data is analyzed and maintained on local fog node closest to

data sources. [11]. However, fog computing still inherits various security risks from cloud computing, it cannot be deemed secure. An attacker may launch attacks like Denial-of-Service, Man-in-the-Middle Sybil, Forgery, impersonation, eavesdropping etc. to disrupt the IoT fog computing platform. [12].An IoT-Fog computing platform is shown in figure 1 below:

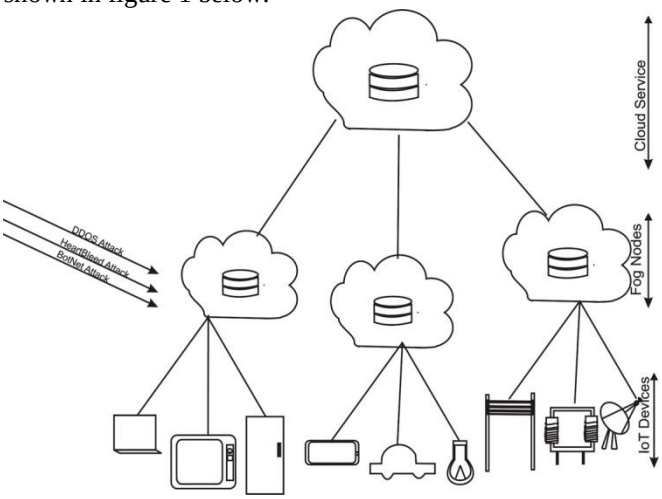

Figure 1: IoT-Fog Architecture

1.2 Intrusion Detection System with IoT Fog as A Case Study

The Intrusion detection system is a well-known security technique deployed to protect sensitive networks like WSN, IOT, 4G etc. against attacks. The task of an IDS is to detect unusual activities that potentially indicate ongoing attacks or malicious activities. Internal and external malicious intruders can attack any entity at any time in the IoT fog computing platform. If there is no efficient intrusion detection system in place to detect malicious intruders, attacks may be successful which will slowly undermine the services provided by the IoT fog computing architecture. [11]. Although, collaboration is possible between adjacent fog nodes and the ones at higher level in the network to detect intrusion and attacks [12].Ensuring that the whole IoT fog computing architecture is protected is a necessity. Therefore, a dependable intrusion detection system is required for each IoT device or fog node. Although there are different classes of IDS which employs different detection techniques, IDS model can either be signature based and/or anomaly-based detection model. In signature based, detection technique is limited to attack behavior pattern that has been pre-defined in the database while Anomaly based model uses a set of rules to detect anomalies in the network behavior based on heuristic techniques. Unfortunately, the later model generates lots of false positive alarms. [13].

This paper is focused on anomaly-based detection technique because of its ability to detect unknown and zero-day attacks. Anomaly based IDS is identified with machine and artificial learning algorithm approaches like support vector machines, naive Bayes, decision trees, artificial neural network etc. Although, several neural network approaches have been adopted to develop various IDS, our work proposes an Intrusion Detection System using an efficient Deep Neural Network's Back Propagation architecture.

2. PREVIOUS RESEARCH WORK

Pandeeswari and Kumar [14]. Proposed VMM (virtual machine monitor) layer intrusion detection system in cloud environment called hypervisor detector. The integration of fuzzy c means (FCM) and artificial neural network (ANN) was used to design the proposed system. The system involves three stages which are the generation of clusters of the same class to improve ANN leaning capacity by the fuzzy-c average algorithm, the generated clusters are used to train different ANN algorithms in the second stage, and a fluid aggregation module is used to reduce the error and integrate results of several ANNs in the third stage. A major setback of this work is the use of the DARPA's KDD cup dataset 1999 for the experiments. Hodo et al [15].Presents a type of supervised ANN namely multi-level perceptron which is trained using internet packet traces and then assessed based on its ability to thwart Distributed Denial of Service (DDoS/DoS) attacks. Their research focused on the normal and threat pattern classification in IoT Network. The IoT-Simulated Network validates the ANN procedure and the results showed a high accuracy in successfully detecting various DDoS/DoS attacks. However, the method is limited to ddos/dos attacks only and more attacks can be introduced to test its reliability. Moreover, other deeper neural networks like convolutionary and back propagation approach can be studied to improve the accuracy of the framework.Hosseinpour et al [16]. Introduced an Artificial Immune System (AIS) distributed and lightweight IDS, with a high detection accuracy. The IDS architecture is spread across a three-layered IoT structure which includes the edge, fog and the cloud layers. In the cloud layer, the IDS train its detectors and clusters primary network traffic. An intelligent data concept is used to analyze intrusion alerts in the fog layer. In the edge layer, detectors are deployed in edge devices. However, a major setback of the work is that it does not detect potential botnet attacks using smart data technology and an old dataset was used to experiment the lightweight IDS. A decision tree (DT) application from an earlier study proposed the use of a fog-based security system for IoT devices [17]. DT was used to examine network traffic in order to detect intrusions and suspicious sources of traffic. An et al [18]. Introduced the Sample Selected Extreme Learning Machine Lightweight IDS (SS-ELM) for FC/MEC to overcome the fog node space limitations. However, SS-ELM was proposed by the authors because fog nodes/MEC hosts cannot store huge amounts of training data sets. Therefore, the selected samples that has been computed, sampled and stored by the cloud servers is sent for training to fog nodes/MEC hosts. Also, according to the authors, KDD Cup 99 data set was used for the experiments because no FC/MEC intrusion detection training sets were available.

3. PROPOSED MODEL

The research objective of this proposed work is to find an improved BP-DNN algorithm that will provide a better accuracy and detection rates for anomaly-based intrusion detection system. The choice of the backpropagation algorithm is due to its computational efficiency andsimplicity.

3.1 BP-DNN Model

Backpropagation is the concept of calculating the error (backpropagating the error) contributed by each node from back all the way down to the front and using it to adjust the weight which has a direct effect on the output. The partial derivate of the error function with respect to the weight is called the gradient descent. This determines thedirection in which we move during backpropagation down the network.
Our major aim is to update the weights at each layer during each step of the backprop. To do that, the partial derivative of the error with respect to weight of each layer is required. This can be achieved using chain rule[19]-[21]. We use the Canadian Institute for Cybersecurity (CIC) 2017 dataset [22] for the experiment to generate the training and testing samples. The dataset was split into two. One half for training and the other for testing phase.
The diagram for the proposed 3-layer BP-DNN architecture and the flowchart for model is shown in figures 2 and3 respectively.
The system will be subdivided into three main phases namely data pre-processing, training and the testing phase.

3.2 The Data Pre-Processing Phase

This phase involves 2 main stages which are:
3.2.1 Dividing the Dataset: This involves detecting the dataset features to be used and separating the training sets from the test sets. Both the training sets and test sets would be separated into features and targets. Features and the targets are labelled columns on the datasets that indicates if a network data traffic is an attack or not. The considered percentage split of the data is 75%-25% for training and test sets respectively.

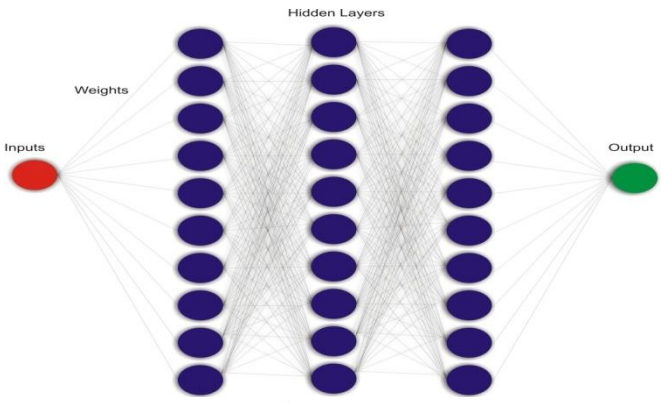

Figure 2: Diagram of the Proposed BP-DNN Model

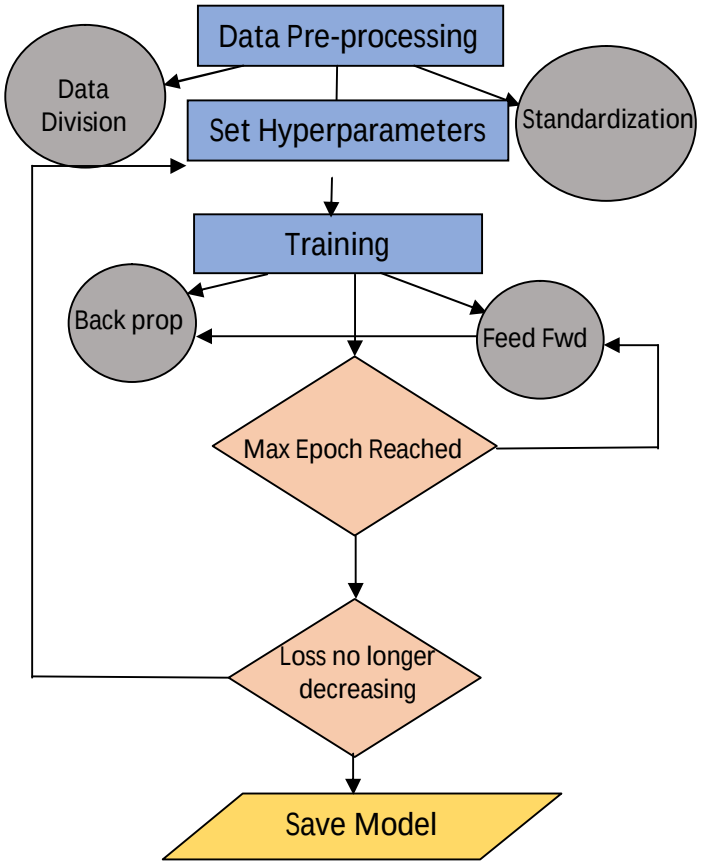

Figure 3: Flowchart for proposed BP-DNN

3.2.2 Data Standardization: This simply means that datasets will be squashed to fall between zeros(0s) and ones(1s) using Batch Normalization which addresses the issues of gradients that explode or vanishes due to high learning rate in a typical deep network. Small changes to the parameters is prevented from amplifying into larger and sub-optimal changes in activation in gradients by normalizing activations throughout the network. i.e. it prevents the training from getting stuck in the saturated regimes of nonlinearities [23]. This can be achieved using the Min-Max Rescaling technique. Min-max scaling will replace every value in a column with a new value using(1) below:

$$m = \frac{x - x_{min}}{x_{max} - x_{min}} (1)$$

Where:m = new value,
x = original cell value,
x_{min}= minimum value of the column
x_{max} = maximum value of the column [24].

The reason for data normalization is because of the target (labelled) in the datasets which will be classified as either an attack or normal traffic, 1 and 0 respectively. 0.5 will be used as a midpoint between the two classes of data types such that prediction below 0.5 signifies a normal traffic and above 0.5 signifies an attack.

3.3 Training Phase

This phase involves the forward pass and the backward pass until the required number of epochs is reached before validation and testing. This is generally represented by (2)below:

$$Z = Wa + b \quad (2)$$

Where Z, W, a and b are output, weights, input and bias respectively. The product of W and a (Wa) together is called the weighted inputs.
The weight is normally updated using (3)

$$W = Wi - \delta * \frac{\partial E}{\partial W} \quad (3)$$

Where W_i is the initial weight,
W is the updated weight,
δ is the learning rate
$\frac{\partial E}{\partial W}$ is the change in the weight that occurred during the forward pass that lead to the error.

In order to increase the convergence speed where the network will arrive at a global minima, the maximum and best fitted value of the learning rate is normally obtainedby adjusting the learning value at every interval during the training process. The Error Function E depends on the output which in turn depends on the weighted inputs (which is actually the output from the previous layer) which finally depends on the weights.

$E(O^l, O^{l-1} \dots O^{l-2})$, $O(Z^l, Z^{l-1} \dots Z^{l-2})$, $Z(W^l, W^{l-1} \dots W^{l-2})$

Therefore, going through the various layers, Z becomes:

$$Z_k^{[l]} = W_{k.1}^{[l]}.a_1^{[l-1]} + W_{k.2}^{[l]}.a_2^{[l-2]} + \dots W_{n^{[l-n]}}^{[l]}.a_{n^{[l-n]}}^{[l-n]} + b_k^{[l]}, \quad \forall k \in [1, \dots, n^{[l]}]$$
$$(W^l, W^{l-1} \dots W^{l-2}) = E(O_1^{[l]}(Z_1^{[l]}\left(W_1^{[l]}, \dots, W_1^{[1]}\right)), \dots, E(O_2^{[l-2]} \left(Z_2^{[l-2]}\left(W_2^{[l-2]}, \dots, W_2^{[l-2]}\right)\right) \quad (4)$$

Applying the chain rule:

$$\frac{\partial E(W^l, W^{l-1} \dots W^{l-2})}{\partial W_i^{[l-1]}} = \sum_{k=1}^{n^{[l]}} \frac{\partial E(O^l, O^{l-1} \dots O^{l-2})}{\partial O_i^{[l-1]}} . \frac{\partial O(Z^l, Z^{l-1} \dots Z^{l-2})}{\partial Z_i^{[l-1]}} . \frac{\partial Z(W^l, W^{l-1} \dots W^{l-2})}{\partial W_i^{[l-1]}} \quad (5)$$

3.3.1 Training the Algorithm:

Weight Initialization: The starting value of the weights can have a significant effect on the training process [25]. Weights should be randomly chosen in such a way that the sigmoid is primarily activated in its linear region. Large weights will saturate the sigmoid resulting in a small gradient and make learning slow, small weights will make gradients very small. The gradients are large enough thatthe learning can proceed and the network will learn the linear part of the mapping before the nonlinear part, if the weights are randomly drawn from a distribution. To achieve this, Gaussian random normal distribution was used.

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-u)^2}{2\sigma^2}} \quad (6)$$

Where u and $\sigma$are mean and standard deviation respectively. The square of the standard deviation is the variance.

3.3.2 Setting the Hyperparameters:

Learning rate: Although choosing a different learning rate for each weight can improve the convergence, a global shared learning rate would be chosen for this work as this is likely to increase the processing speed of the network which is proportional to the square root of the number of connections sharing that weight.

Hidden layers: This is another sensitive parameter to set, as it can cause the network to be overfitted or underfitted. We used 3 hidden layersfor this work. But based on the network testing prediction accuracy and training error rate, number of hidden layers can be fluctuated as an extension of our work to further optimize our model**.**

*Epochs:*Epoch is the number of times the dataset is passed through the standard backpropagation network model for training. The number of epochs will be carefully chosen after multiple trials to prevent the slow training process of the network and also to prevent it from overfitting.

3.4 Testing Phase

The phase involves passing the data through the designed model in a forward pass to see its accuracy by comparing it with the midpoint that was chosen to separate the upper bound (anomaly packets) and the lower bound (normal packets) then finding the average of the results.
Another important process to note in our model is the process of transformation which normally takes place at theend of a particular layer. This serves as input to the next layer which is then multiplied by the weights. Nonlinear activation functions are what give neural networks their nonlinear capabilities. sigmoid function is the activation function that will be deployed in our work, it is the most common forms of activation function that asymptotes at some finite value as is approached [26].

3.5 Results and Analysis

Table 1 below shows the result obtained. The results were relatively uniform across the dataset that was used:

Table 1: Results obtained

Dataset	Size (Mb)	Accuracy
Dos/DDoS	75,317	78.3%
DDoS LOIT	172,782	78.3%
Botnet(ARES)	56,950	84.5%
Pot Scan	75,104	86.5%
WebAttack(BruteForce,XSS,Sql)	81,155	78.3%
Brute Force	50,804	78.3%

Average accuracy obtained is about 80.7%. However, after passing it through an unlabeled dataset, the accuracy of the result was 84.5% which is slightly greater than the average accuracy.

3.6Software and Hardware Platform

An Artificial Neural Network of 3 deep layers was used for our standard BP-DNN model with a hidden neutron of one at each layer. The Fog server is simulated in a MacOs operating system (2.8 GHz CPU intel core i7, 16.00 GB RAM 2133MHz LPDDR3), and the BP-DNN algorithm is implemented using Jupiter Notebook version 6.1.5.

4. CONCLUSION AND FUTURE WORKS

Several algorithms such as decision trees, naive Bayes, support vector machines, artificial neural networks and others have been developed to solve problems of intrusion detection, most of these algorithms were evaluated with old datasets. Most model that used recent datasets were designed for the cloud platform. We used a modern dataset obtained from the canadian institute for cybersecurity (CIC2017)datasetsto implementa standardBP-DNN algorithms with 3 hidden layers and low computational complexity for intrusion prediction accuracy. Extension of our work would be the actual training, testing and evaluation of our model with the CIC datasets and the comparison of the performance with various other IDS algorithms deployed on fog nodes.

REFERENCES

1. S.Li, L.Da Xu, and S. Zhao, “The internet of things: a survey,” Inf. Syst. Front., vol. 17, no. 2, pp. 243–259, 2015. https://doi.org/10.1007/s10796-014-9492-7 , [accessed on Sept. 2020].
2. Z. Pang, Q. Chen, W.Han,andL. Zheng, “Value-centric design of the internet-of-things solution for food supply chain: Value creation, sensor portfolio and information fusion,” Inf. Syst. Front., vol. 17, no. 2, pp. 289–319, 2015 https://doi.org/10.1007/s10796-012-9374-9 ., [accessed on Sept. 2020]
3. Zanella, A., Bui, N., Castellani, A., Vangelista, L., &Zorzi, M. (2014). Internet of things for smart cities. IEEE Internet Things J 1 (1): 22–32.https://doi.org/10.1109/jiot.2014.2306328., [accessed on Sept. 2020].
4. Gartner Says 8.4 Billion Connected “Things” Will Be in Use in 2017, Up 31 Percent from 2016,” http://www.gart- ner.com/newsroom/id/3598917, Feb. 2017; accessed on 08/03/2017.
5. S. Verma et al., “A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues,” IEEE Commun. Surveys Tutorials, vol. 19, no. 3, 2017, pp. 1457–77.
6. Zheng, X., Martin, P., Brohman, K., & Xu, L. D. (2014). Cloud Service Negotiation in Internet of Things Environment: A Mixed Approach. *IEEE Transactions on Industrial Informatics*, *10*(2), 1506–1515. https://doi.org/10.1109/TII.2014.2305641
7. Botta, A., de Donato, W., Persico, V., &Pescapé, A. (2016). Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, *56*, 684–700. https://doi.org/10.1016/j.future.2015.09.01
8. F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog Computing: A Platform for Internet of Things and Analytics,” in Big Data and Internet of Things: A Roadmap for Smart Environments, Studies in Computational Intelligence, vol. 546, N. Bessis and C. Dobre, Eds. Cham: Springer International Publishing, 2014, pp. 169–186.
9. F. Bonomi et al, “Fog Computing and its Role in the Internet of Things,” Proceedings of the first edition of the MCC workshop on Mobile cloud computing, Aug. 2012. [Online]. Available: https://doi.org/10.1145/2342509.2342513, pp. 13–16.
10. “Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are,” CISCO, White Paper, 2015.
11. J. Ni, K. Zhang, X. Lin and X. Shen, "Securing Fog Computing for Internet of Things Applications: Challenges and Solutions," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601-628, Firstquarter 2018, doi: 10.1109/COMST.2017.2762345.
12. R. Roman, J. Lopez, and M. Manbo, “Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security, Threats and Challenges,” *Futur. Gener. Comp. Syst.*, vol. 78, pp. 680–698, 2018.
13. D. A. Effendy, K. Kusrini and S. Sudarmawan, "Classification of intrusion detection system (IDS) based on computer network," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia, 2017, pp. 90-94, doi: 10.1109/ICITISEE.2017.8285566.

14. Pandeeswari, N., Kumar, G. Anomaly Detection System in Cloud Environment Using Fuzzy Clustering Based ANN. *Mobile NetwAppl*21, 494–505 (2016). https://doi.org/10.1007/s11036-015-0644-x
15. E. Hodo, X. Bellekens, A. Hamilton, P. L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in 2016 International Symposium on Networks, Computers and Communications (ISNCC), May 2016, pp. 1–6.
16. Hosseinpour, F.; VahdaniAmoli, P.; Plosila, J.; Hämäläinen, T.; Tenhunen, H. An intrusion detection system for fog computing and IoT based logistic systems using a smart data approach. *Int. J. Digit. Content Technol. Appl.*2016, *10*, 34–46.
17. Alharbi, S.; Rodriguez, P.; Maharaja, R.; Iyer, P.; Subaschandrabose, N.; Ye, Z. Secure the Internet of Things with Challenge Response Authentication in Fog Computing. In Proceedings of the 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC), San Diego, CA, USA, 10–12 December 2017; pp. 1–2.
18. An, X.; Zhou, X.; Lü, X.; Lin, F.; Yang, L. Sample selected extreme learning machine-based intrusion detection in fog computing and MEC. *Wirel. Commun. Mob. Comput.*2018, *2018*, 7472095.
19. Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for machine learning*. Cambridge University Press.
20. Ng, A. (2019). Machine learning yearning: Technical strategy for ai engineers in the era of deep learning. *Retrieved online at https://www. mlyearning. org*.
21. Hefferon, J. (2018). Linear Algebra Third Edition
22. Canadian Institute for Cybersecurity Dataset 2017. https://www.unb.ca/cic/datasets/index.html
23. Ioffe, S., &Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR
24. Ozdemir, S., &Susarla, D. (2018). *Feature Engineering Made Easy: Identify unique features from your dataset in order to build powerful machine learning systems*. Packt Publishing Ltd.
25. LeCun Y.A., Bottou L., Orr G.B., Müller KR. (2012) Efficient BackProp. In: Montavon G., Orr G.B., Müller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 7700. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35289-8_3.
26. Sadowski, P. (2016). *"Notes on Backpropagation"*, Department of Computer Science, University of California Irvine.