FDeep: A Fog-based Intrusion Detection System for Smart Home using Deep Learning

Tahani Gazdar
Cyber Security Department-CCSE
University of Jeddah, Jeddah, Saudi Arabia

***Abstract*—Smart Home is an application of the Internet of Things (IoT) that connects smart appliances and the Internet. The emergence of Smart Home has caused many security and privacy risks that can lead to fatal damages to the user and his property. Unfortunately, Intrusion detection systems designed for conventional networks have shown their inefficiency when deployed in Smart Home environments for many reasons that rely basically on the resources-constrained devices and their inherent intermittent connectivity. So, an intrusion detection system designed for IoT and particularly Smart Home is mandatory. On the other hand, Deep learning shows its potential in enhancing the performance of Intrusion Detection Systems. According to recent studies, Deep learning-based intrusion detection systems are deployed either on the devices or in the Cloud. However, Deep learning models are greedy in terms of resources which makes it challenging to deploy them on Smart Home devices. Besides, in the IoT architecture, the IoT layer is far from the Cloud layer which may cause additional latency and jitter. To overcome these challenges, a new intrusion detection system for Smart Home deployed in the Fog Layer is proposed, it is called FDeep. FDeep will inspect the traffic using a Deep Learning model. To select the most accurate model, three Deep Learning models are trained using an IoT dataset named TON/IIOT, also the proposed models are compared to an existing one. The obtained results show that the long short-term memory model combined with the convolutional neural networks outperforms the other three models. It has the best detection accuracy compared to other Deep Learning models.**

Keywords*—Fog computing; smart home; deep learning; IDS; classification

I. Introduction

The IoT environment is a system of smart devices that are interconnected and enabled with sensing and data transmitting capabilities [1]. IoT technology is used in Smart Home to enhance safety and convenience. Nevertheless, providing privacy and security in Smart Home environments is the key challenge facing their deployment. Many attacks threaten a Smart Home and may expose the data or properties to risks. One of the popular attacks is Denial of Service (DoS) on client-server applications. Eavesdropping is another common attack where an attacker intercepts data packets flowing in the network.

An intrusion detection system (IDS) analyzes the data packets to detect potential intrusion, and then generates alerts in case of any suspicious behaviour [2]. Particularly, anomaly-based IDSs are characterized by their efficiency in detecting zero-day and new attacks. This kind of IDS is based on Machine learning (ML) models that can automatically learn from experience and analyze patterns based on collected data. Deep Learning (DL) is a subcategory of ML, it can understand patterns using many layers of processing however it needs a large amount of data [3][4]. This represents one of the main challenges facing deploying IDS based on DL in IoT, in addition to the limited computation capabilities of the IoT devices. More important, most IDSs proposed in the literature are deployed on IoT devices which is not practical. DL models cannot run efficiently on resources constrained IoT devices because of the huge amount of data to analyze in addition to the complexity of DL models that needs high computation resources [2]. Further, many approaches implement the DL models in the Cloud, however, a centralized intrusion detection approach is not adequate for IoT because of their large scale and the caused latency. A distributed approach would be more appropriate. Besides, the efficiency of DL models degrades over time, so they need to be re-trained regularly with new data to enhance the detection capabilities mainly for zero-day and unknown attacks [5]. Most approaches re-train the models either on the Cloud or on desktops, then replace the old model with the new one. In these approaches, a huge amount of data packets is required. In the context of IoT and particularly Smart Home, the collected data may contain sensitive data related to the end user which may violate his privacy. Thus, an efficient, distributed, and accurate IDS for Smart Home is required to provide strong intrusion detection capabilities [2].

In this paper, new Fog-based IDS architecture dedicated to Smart Home named FDeep is proposed. The objective is to leverage Fog computing in IoT to provide a distributed, efficient, and high computation capacity IDS for Smart Home. An IoT application particularly a Smart Home consists of three main layers: the Edge layer and the Cloud layer and a Network layer. The Edge layer consists of a set of Smart devices responsible for sensing their environment, collecting data, and transmitting it through a Network layer to the Cloud Layer. This data is processed and analyzed in the Cloud layer to provide a wide range of applications and services to the end user. The two layers communicate which each other through gateways and routers. Generally, the Edge layer is too far from the Cloud layer, which might cause additional latency and jitter and requires high bandwidth consumption. Here, the paradigm of Fog computing comes to overcome this limitation by extending the Cloud and making it closer to the user. Fog computing provides computing, networking, and storage services to the Edge and Cloud layers in a distributed environment [6]. Unlike many approaches where DL models are implemented either in the Cloud or the Edge layers, in

FDeep the proposed DL model will be implemented in the Fog layer which represents the glue between the Cloud and Edge Layers. Besides, to maintain the DL model and avoid its degradation in terms of detection, periodic training of the DL model will be triggered using the data newly collected from the real network in the Fog layer.

The key contributions of the paper are summarized as follows:

- In FDeep, the DL model will be implemented in the Fog layer to guarantee low latency and reduce bandwidth consumption in addition to a fast inspection and detection of the attacks occurring in the IoT layer.
- An accurate DL model aiming to classify the attacks into seven types is proposed.
- An IoT dataset is used to train our models and select the most accurate one to be deployed in FDeep.
- The DL model to be deployed in FDeep will be periodically re-trained and updated to avoid its degradation. To do so, the data packets are collected from the IoT layer of the real network and fed to the model to re-train it.

The remainder of the paper is structured as follows. In Section II, the most recent approaches interested in DL-based IDS for IoT and its applications are discussed. Section III presents the architecture of FDeep in detail. In Section IV, the used dataset is described. In Section V, the experiments set up are presented and the obtained results are analyzed. Section VI concludes the paper and presents future work.

II. Related Work

DL-based Intrusion detection systems are widely adopted to inspect network traffic and detect intrusion and misbehaviour [2][7]. In [8], a DDoS detection system based on Deep Learning DL is proposed. The authors evaluated the performance of several DL approaches with ML techniques for DDoS attack detection. The obtained results point out the potential of Deep Learning in enhancing the accuracy of detection of DDoS attacks. A behavioural model is proposed in [9] to detect malicious traffic generated by compromised IoT devices using Autoencoder (AE). It is a kind of Deep Learning that learns unsupervised data. AEs are widely used in IoT security. In their study, AEs were used to extract features derived from cyber systems. In [10] a Deep Learning based intrusion detection system for IoT has been proposed. The authors used an SMO model to enhance the convergence time and feature extraction. Besides, they used the SDPN classifier to distinguish benign traffic from malicious one. The authors pretend that their model can handle datasets with redundant values and uncertain or missing data. However, the proposed model detects a few attacks: DoS, U2R, probe, and R2L. Unfortunately, all the above-discussed approaches don't mention how to implement the proposed DL models.

The authors proposed in [5] an attack detection system for IoT based on Deep Learning. They have implemented the proposed framework on Fog nodes. The authors have adopted an LSTM model to analyze and inspect the data packets collected from IoT devices to detect potential attacks. The maintenance of the model is performed in the Cloud layer to update the DL model and avoid its degradation in terms of detection capabilities. The limitation of the proposed architecture is the fact that the administrator must regularly check the accuracy of the model and maintain it manually. Additionally, the authors used dataset KDD CUP 99 which is not an IoT-specific dataset. However, according to [2], a model trained for a specific network or application may not be efficient for another network or application. In IoT, it is even recommended to design each kind of device with its own DL/ML model given the diversity of their sensing capabilities [2]. In [11], an LSTM model has been used to detect attacks in Fog-to-Things Communications. They proposed a distributed framework to detect attacks. However, their approach detects only attacks on Fog nodes and disregards the risk from IoT devices.

In [12], the authors proposed an intrusion detection model to detect attacks in a Fog computing environment. They have integrated CNN with LSTM to obtain an integrated and more accurate model. The limitation of this approach is the use of NSL-KDD which is not an IoT dataset. According to [13], many research studies are interested in intrusion detection systems for Fog-based IoT applications, however, these approaches detect only intrusions in Fog nodes. In [14], the authors proposed an IDS based on Ensemble learning for Fog-to-things environments. To enhance anomaly detection accuracy, the authors combine many classifiers to build two levels of classifications. The first layer detects the potential anomalies while the second layer classifies the detected anomalies. Similarly, DL models for attack detection in Fog-assisted IoT are proposed in [15] and [16]. Again, a non-IoT-specific dataset is used. Consequently, the accuracy of the model may be negatively affected once it is deployed in Fog-to-things environments.

To overcome the limitations of the above-discussed models we propose in this paper to implement the DL model in the Fog layer. Additionally, an IoT-specific dataset called TON/IIoT is used, it contains records collected from real IoT devices, and also it contains records about seven attacks.

III. Proposed IDS Architecture

A. Fog Computing and IoT

Recently, Cloud computing is the most popular computing paradigm. However, the spreading of IoT creates several challenges for Cloud computing. Particularly, most IoT applications require low latency which is not easily provided by Cloud computing since it has a centralized architecture that ill-suites the huge scale and the distribution of IoT. Moreover, the huge scale of IoT that relies basically on the high number of Smart devices increases the amount of data generated in the network. Sending this high amount of data to the Cloud will certainly need a high network bandwidth in addition to privacy concerns. Data generated in edge devices should be processed locally as most as possible. However, edge devices are usually resource-constrained devices unable to perform complex tasks and run complex protocols and models. Moreover, the intermittent connectivity due to the geographical distribution of edge devices makes them unable to benefit from

uninterruptible Cloud services. Thus, an intermediate layer between the Cloud and the IoT devices is unavoidable [6].

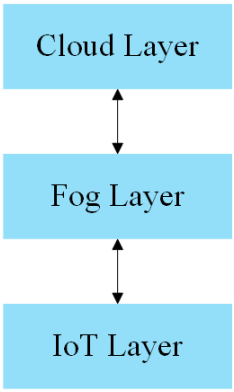


Fig. 1. 3-tier fog architecture.

Fog computing comes to bridge this gap and provide distributed networking, storage, control, and computing, capabilities between the Edge Layer and the Cloud layer [6]. Typically, the Fog layer is connected to the edge layer, which means that it is closer to the user than the Cloud layer. It is characterized by its flexibility to provide computation services to a large scale of end users in its proximity. Hence, it fosters decreasing the latency and accelerating decision-making. As depicted in Fig. 1, Fog computing has three-tier architecture: IoT layer, Fog layer, and Cloud Layer.

The lower layer is the IoT layer which is composed of the IoT devices such as smart vehicles smartphones, sensors, Cameras, smart appliances, etc. These IoT devices are heterogeneous in terms of computing capabilities, vendors, firmware, etc. They are geographically distributed, and their role is to collect data like temperature, fire, images, etc, and send them to the upper layer. The Fog layer consists of many Fog nodes responsible for providing computing, storage, and control services to the Cloud layer and IoT layer. A Fog node may be a router, a virtualized Fog server, or a simple data centre. Typically, the data is routed from the IoT layer to the Fog layer using smart gateways. The Cloud layer is the upper layer that consists of many servers characterized by their high storage and computational capabilities.

B. FDeep Architecture

As depicted in Fig. 2, the architecture of FDeep consists of 3 layers: IoT layer, Fog layer, and Cloud Layer. In the following, the role of each layer and the communications between them are detailed.

1) Data collection: All packets exchanged between the Smart devices in the IoT layer or to/from the Fog Layer pass through gateways. A network sniffer such as sflow or tcpdump, is deployed on these gateways to capture all network traffic, and the output will be saved in .pcap files. Then, the records are preprocessed to extract the useful features (timestamp, protocol, IP addresses, etc) and create CSV files containing pertinent data that will be inspected in the Fog layer to detect potential intrusions in the Smart Home. These data will be also used to re-train the DL model to improve its detection capabilities.

2) Data analysis and intrusion detection: The objective of FDeep is to detect potential intrusions in a Smart Home environment. To this end, a DL model is used to catch attacks and classify them. Three DL algorithms will be trained later and the best one in terms of detection accuracy will be deployed in FDeep.

In the current paper, the aim is to leverage Fog computing to build an efficient IDS by deploying the DL model on the Fog layer, doing so has many benefits. First, Fog computing is usually close to the user which reduces the latency and makes it best suited the real-time applications in IoT contexts. This feature is interesting in our context because rapid intrusion detection is required. The data will be processed close to the smart devices from which it is originating, no need to transmit it to the Cloud which reduces bandwidth consumption. Secondly, the OpenFog Security Group[1] has defined two main security goals of Fog computing. The first goal is the intrinsic security of Fog computing in terms of responsiveness, availability, fault tolerance, and trust. Fog computing can provide security services for the IoT layer since most of them are resources constrained such as identity verification, and end-point protection.

In the current study, the aim is to use a DL classifier to detect and classify intrusion in a Smart Home based on data received from the IoT layer, and since most IoT devices have limited resources, it seems interesting to implement the DL model in the Fog layer instead. It will provide a scalable, distributed and high computation environment for FDeep.

3) Updating the DL model: Generally speaking, DL models degrade over time because new applications may be deployed, and attacks may emanate. That is why maintaining the DL model is mandatory [5]. Hence, periodic training of the DL model will be triggered in the Fog Layer using the traffic newly captured from the IoT layer. Doing so has two main benefits, first, it will reduce the load on the Cloud layer in terms of computation, and also reduce the latency of transmitting the collected data from the IoT Layer to the Cloud. Secondly, it will make the model customized to the set of edge devices under the control of the Fog nodes. The edge devices deployed in the IoT layer may have different sensing capabilities and are vulnerable to different attacks, so it is interesting to have a DL model trained on the data received from the target devices [2].

The different models obtained in the Fog layer may be later shared with the Cloud layer to create a more generic model in a way similar to Federated Learning [17].

[1] https://opcfoundation.org/markets-collaboration/openfog/

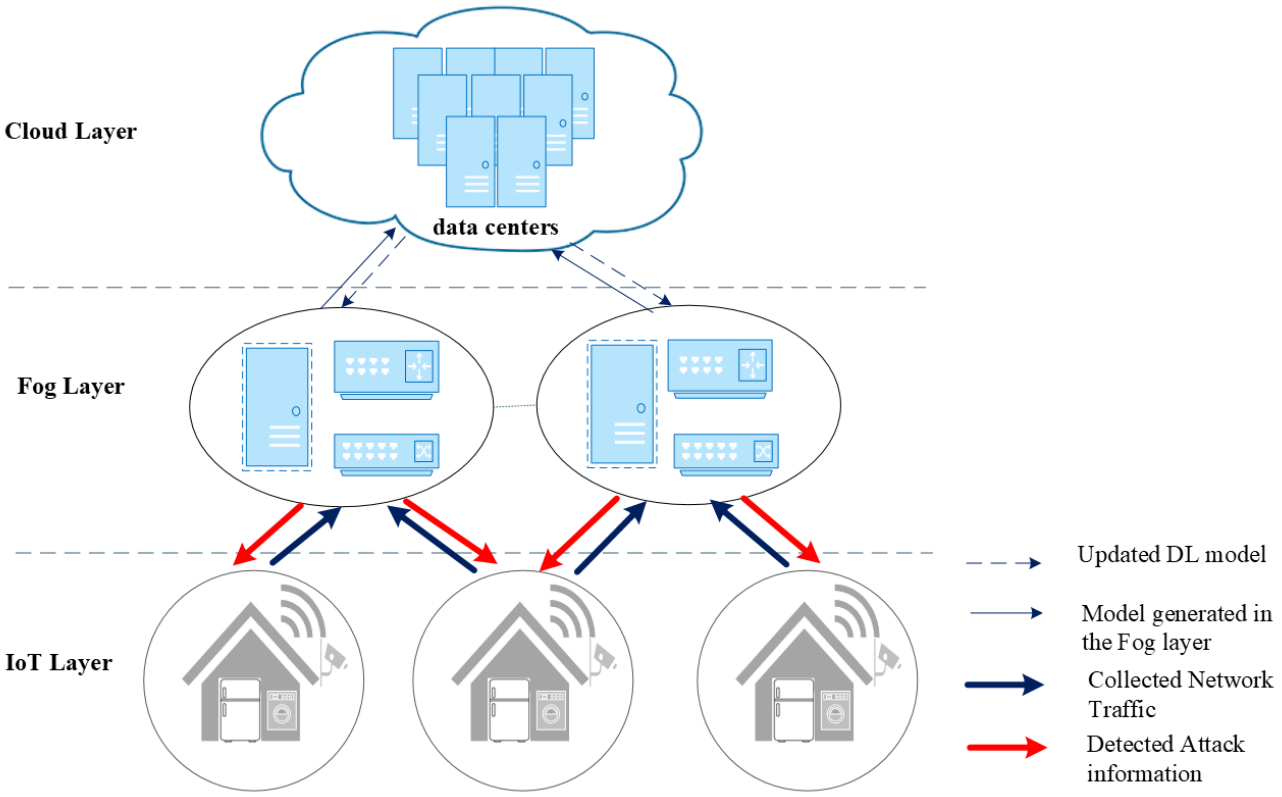


Fig. 2. FDeep architecture.

C. DL Model

To select the best DL model to deploy in FDeep, the performances of three DL algorithms are compared in terms of accuracy, precision, recall, and F1 score.

1) LSTM: Long Short-Term Memory Network (LSTM) is capable of learning order dependence in sequence prediction problems. It has feedback connections, that make it can process the entire sequence of data [18]. The state of an LSTM model is maintained over time by a memory cell named 'cell state'. This characteristic of LSTM makes it very suitable for environments where data should be processed sequentially like network traffic analysis, particularly, IoT systems. In intrusion detection, the major feature of LSTM is its ability to differentiate between benign and malicious traffic by only inspecting a small number of network data packets which makes it efficient for real-time analysis of the traffic [7].

2) CNN: CNN (Convolutional neural network) is among the most commonly used deep learning algorithm. The advantage of CNN is its ability to handle large datasets also its computational efficiency. CNN is also known for its efficiency in feature extraction. CNN consists of an input layer, a hidden layer, and an output layer. The hidden layer consists of many convolution layers, pooling layers, and a fully connected layer. First, the convolution layer extracts the features or information from the data using filters. The pooling layer is responsible for parameter reduction. Unlike conventional feature extraction algorithms, CNN can automatically learn the best features. Afterwards, the convolution maps are combined to form a unique vector known as CNN code [2]. Then, the classification layer receives the CNN code from the preceding levels and merges its characteristics to categorize the data [2].

3) CNN-LSTM: CNN-LSTM is a hybrid model that combines CNN and LSTM [19]. It is initially designed for visual time series predictions and textual generation from image sequences. In this model, the CNN is the features extractor, then the output is fed to LSTM which is the classifier. CNN-LSTM has been used also for intrusion detection [5].

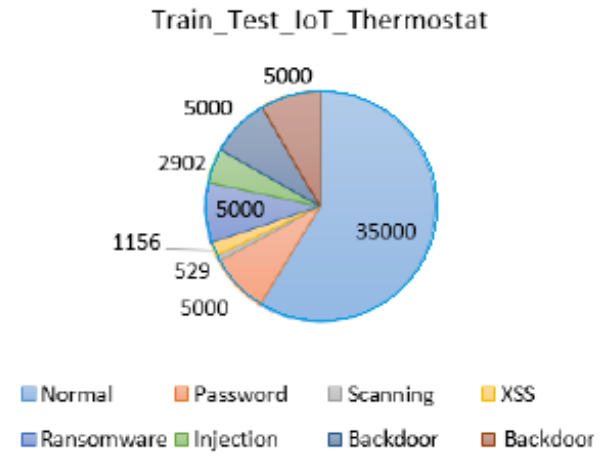
IV. Dataset

A new IoT dataset named TON/IIOT [20] is used. It is a recent dataset that consists of data collected from a real testbed of IoT devices like a Thermostat, smart fridge, weather sensor, motion light, etc. The dataset consists of many CSV files, each file corresponds to one device, and it contains data, features, and attacks that rely on the sensing capabilities of that device. The CSV files contain records that correspond either to benign or malicious traffic. The malicious records correspond to the following attacks: backdoor, DoS, DDoS, jamming, ransomware, scanning, password, XSS, cracking, injection, and man-in-the-middle [20]. In the current study, the results for only two devices are shown: the weather sensor and the Thermostat. Table I represents the set of features in the CSV file of the Thermostat.

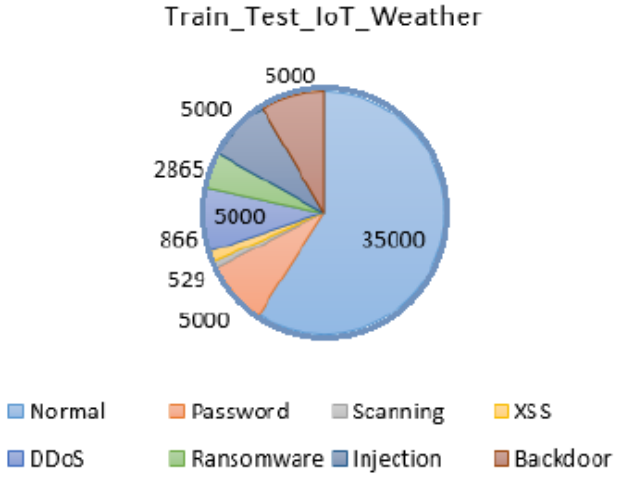
TABLE I. FEATURES IN THE THERMOSTAT CSV FILE

Feature	Description
Ts	Timestamp of sensor reading data
Date	Date of logging sensor's telemetry data
Time	Time of logging sensor's telemetry data
current_temperature	Current Temperature reading of a Thermostat sensor
thermostat_status	Status of a Thermostat sensor is either on or off
label	Identify normal and attack records, where '0' indicates normal and '1' indicates attacks
Type	A tag with normal or attack sub-classes, such as DoS, DDoS and backdoor attacks

The CVS file of the Thermostat contains 52774 where 17774 correspond to malicious traffic and the rest represent normal traffic. Similarly, the CSV file of the IoT Weather Sensor contains 54260 records whereas 19260 records correspond to attacks. The distribution of records among classes is shown in Fig. 3.



(a) Thermostat.



(b) IoT Weather sensor.

Fig. 3. Records distribution per attack [2].

V. EXPERIMENTS AND RESULTS

A. Experiments Set up

Google Colaboratory platform is used in all the experiments since it provides a panoply of Python libraries that support Deep Learning. The dataset is split into a training set that contains 75% of the total records, and 25% of the records are used in testing. DL models are implemented using Keras libraries. Softmax is used activation function since it is a multiclass classification problem and Adam is an optimizer to compute individual learning rates and enhance the accuracy of the models. The batch size is equal to 256, the total epochs number is equal to 300, and the LSTM output is equal to 100.

Since the attacks are classified into seven classes (attacks), we have opted for Categorical Cross Entropy as a loss function. The loss function is a way to evaluate to which extent the algorithm is modelling the dataset [21]. In the Categorical Cross Entropy, one category value encoded in binary is assigned to the output label. If it is in integer form, the 'keras.utils to_categorical' method is used to convert it into categorical encoding using.

B. Results Analysis

We have used the accuracy, precision, recall, and F1-score to evaluate the performance of the DL models[2]. The accuracy is the ratio of correctly classified inputs to the total number of inputs:

$$\text{Accuracy}= \frac{\text{\# correctly classified input}}{\text{\# of inputs}} \quad (1)$$

The precision is the ratio of correctly predicted positive observations to the total number of positive observations. It is computed as follows:

$$\text{Precision}= \frac{\text{\#True Positives}}{\text{\#True Positives + \#False Positives}} \quad (2)$$

Recall evaluates the proportion of malicious input correctly identified. Its mathematical equation is as follows:

$$\text{Recall}= \frac{\text{\#True Positives}}{\text{\#True Positives + \#False Negatives}} \quad (3)$$

F1 score shows if the model has correctly classified malicious input while minimizing false positives and false negatives rates:

$$\text{F1}=2* \frac{\text{Precision*Recall}}{\text{Precision+Recall}} \quad (4)$$

Fig. 4 and 5 represent the accuracy of LSTM, CNN, CNN-LSTM, and the LSTM model proposed in [19] for the IoT Weather sensor and the Thermostat. CNN-LSTM achieves the highest accuracy for both devices. It is about 98% for the IoT whether sensor and 75% for the Thermostat. The difference in performance between the devices is due to the unbalanced dataset for the Thermostat. Besides, as depicted in Fig. 4 and 5, the LSTM model proposed in [20] has the lowest accuracy among all models, it is about 85% for the IoT weather sensor and about 67% for the Thermostat. The accuracy of CNN-LSTM is close to LSTM which is about 97% for the IoT weather sensor.

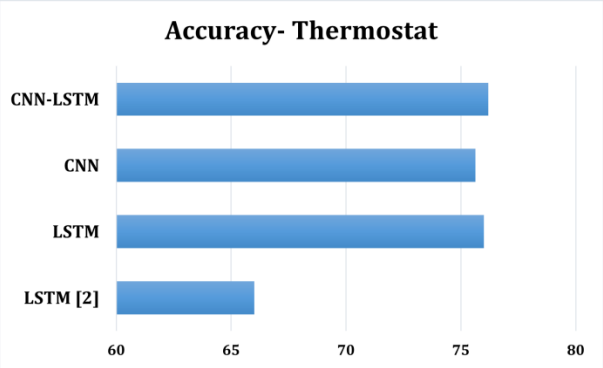


Fig. 4. Average accuracy–thermostat.

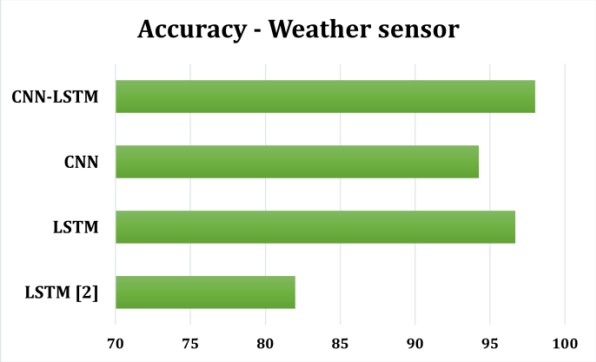


Fig. 5. Average accuracy-IoT weather sensor.

Fig. 6 and 7 plot the loss values for CNN-LSTM for the Thermostat and the IoT weather sensor respectively as a function of the epochs. The loss function is used to compute the distance between the current outputs of the algorithm training) and the expected output (testing). As shown in Fig. 6 and 7, as the training progresses, the value of the loss continuously decreases.

Fig. 7 shows that the loss decreases inversely to the epochs for the IoT weather sensor; it stabilizes and reaches a minimum of 0.05 after the first 100 epochs. Similarly, in Fig. 8 and 9, the accuracies of CNN-LSTM in the training and testing are compared for the Thermostat and IoT weather sensor respectively as a function of the epochs. As depicted in Fig. 8 and 9, the accuracies of the model in training and testing are close for both devices. It increases as the number of epochs increases to achieve the best accuracy at 300.

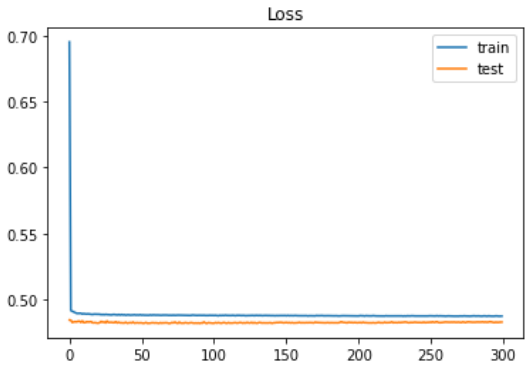


Fig. 6. CNN-LSTM loss values-thermostat.

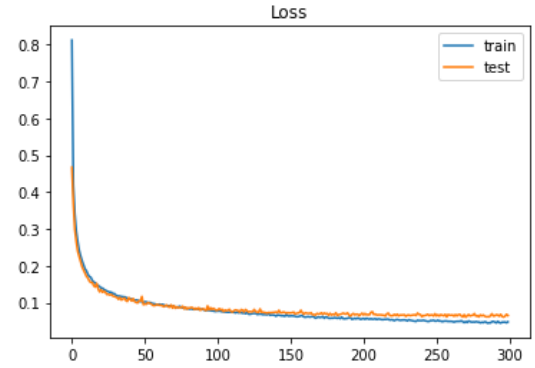


Fig. 7. CNN-LSTM loss values-IoT weather sensor.

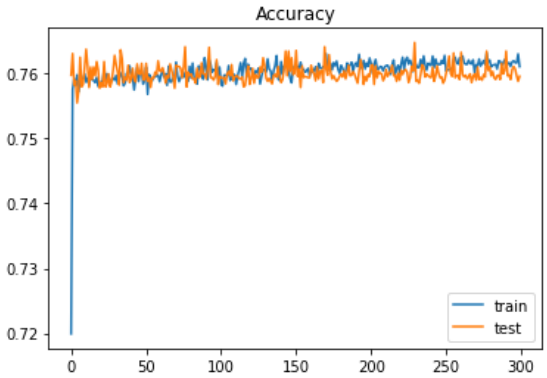


Fig. 8. Accuracy performance CNN-LSTM-thermostat.

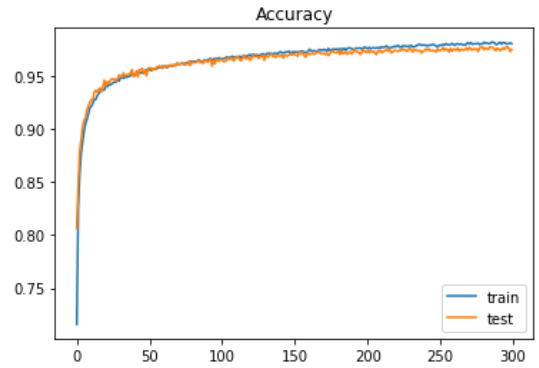


Fig. 9. Accuracy performance CNN-LSTM-IoT weather sensor.

Tables II and III show the values of accuracy, precision, recall and F1 score of the different DL models in multi-class classification for the Thermostat and the IoT weather sensor, respectively. Additionally, the performance of the proposed models is compared to the performance of the LSTM model proposed in [20] for the Thermostat and the IoT weather sensor. The results show that CNN-LSTM outperforms all the other DL models for both devices. It reaches the highest accuracy of 76.19% for the Thermostat and 98% for the IoT weather sensor. Additionally, the LSTM model proposed in [20] has the worst performance compared to other DL models.

TABLE II. PERFORMANCE OF DIFFERENT MODELS- IOT WEATHER SENSOR

	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)
LSTM[20]	82	82	80	81
LSTM	96.69	96.87	96.69	96.52
CNN	94.25	94.67	94.24	93.82
CNN-LSTM	98.02	98.02	98	97.97

TABLE III. PERFORMANCE OF THE DIFFERENT MODELS FOR IOT WEATHER SENSOR IN MULTICLASSIFICATION

	LSTM			CNN			CNN-LSTM		
	Precision (%)	Recall (%)	F1 score (%)	Precision (%)	Recall (%)	F1 score (%)	Precision (%)	Recall (%)	F1 score (%)
Backdoor	94	93	94	94	76	84	94	76	84
DDoS	92	88	90	78	90	83	78	90	83
Injection	94	95	94	96	89	92	96	89	92
Normal	100	100	100	100	100	100	100	100	100
Password	88	93	91	85	85	85	85	85	85
Ransomware	92	89	91	75	91	82	75	91	82
Scanning	99	96	98	99	96	97	99	96	97
XSS	87	88	87	89	77	83	89	77	83

It is also obvious from Tables II and III that CNN-LSTM has the best performance in terms of precision, F1-score and recall for both devices. Particularly, the performance of CNN-LSTM for the IoT weather sensor is better than the Thermostat. The outperformance of CNN-LSTM is due to the proven efficiency of LSTM in the classification in addition to the effective feature extraction performed by CNN.

For multi-class classification, every CSV file of a device has seven attacks in addition to the normal or 'benign class'. Only the performance of the IoT Weather Sensor is presented in detail for multi-class classification because of space limitations. Table IV shows the precision, recall, and F1 score for each DL model.

As depicted in Table IV, CNN-LSTM outperforms LSTM in terms of overall performance for 5 classes (among 8), however, the performance of CNN-LSTM is close to that of CNN. The unbalance of the dataset has a negative impact also on the performance of all DL models.

TABLE IV. PERFORMANCE OF DIFFERENT MODELS - THE THERMOSTAT

	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)
LSTM [20]	66	45	54	67
LSTM	76	100	79.4	66.28
CNN	75.63	100	79.31	66.17
CNN-LSTM	76.19	100	79.31	66.17

VI. CONCLUSION

In this paper, a new distributed IDS system based on Fog-computing to detect intrusions in a Smart Home is proposed. The architecture consists of three layers: IoT layer, Fog layer, and Cloud layer. The IDS detects the intrusion by inspecting the traffic collected from the IoT devices in the IoT layer. A DL model implemented in the Fog layer has been used to inspect the data packets. The implementation of the DL model in the Fog layer provides a high computation environment for the IDS, reduced latency, and avoids implementing the DL model in the end-user devices. To maintain and update the DL model, the model is re-trained periodically in the Fog layer using the data collected from the IoT layer. Many experiments have been conducted to evaluate the performance of many DL models to select the appropriate and efficient one to be implemented in the IDS. In the training, an IoT dataset named TON/IIoT is used, it contains the data collected from a real testbed composed of many IoT devices. The results show that CNN-LSTM has the best performance in terms of accuracy, precision, recall, and F1-score compared to other DL models and the model proposed in [20]. In future work, the performance of the architecture in terms of response time and robustness will be evaluated.

REFERENCES

[1] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," 2018 10th International Conference on Cyber Conflict (CyCon), 2018, pp. 371-390, doi: 10.23919/CYCON.2018.8405026.

[2] Asharf, Javed, Nour Moustafa, Hasnat Khurshid, Essam Debie, Waqas Haider, and Abdul Wahab. 2020. "A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions" Electronics 9, no. 7: 1177. https://doi.org/10.3390/electronics9071177.

[3] T. A. Abdullah, W. Ali, S. Malebary, and A. A. Ahmed, "A review of cyber security challenges, attacks and solutions for internet of things based smart home," Int. J. Comput. Sci. Netw. Secur., vol. 19, no. 9, p. 139,2019.

[4] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," 2018 10th International Conference on Cyber Conflict (CyCon), 2018, pp. 371-390, doi: 10.23919/CYCON.2018.8405026.

[5] A. Samy, H. Yu and H. Zhang, "Fog-Based Attack Detection Framework for Internet of Things Using Deep Learning," in IEEE

Access, vol. 8, pp. 74571-74585, 2020, doi: 10.1109/ACCESS.2020.2988854.

[6] M. De Donno, K. Tange and N. Dragoni, "Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog," in IEEE Access, vol. 7, pp. 150936-150948, 2019, doi: 10.1109/ACCESS.2019.2947652.

[7] B. Farhan, A. Jasim, Survey of Intrusion Detection Using Deep Learning in the Internet of Things. Iraqi Journal for Computer Science and Mathematics. 2021, 10.52866/ijcsm.2022.01.01.009.

[8] Susilo, Bambang, and Riri Fitri Sari. 2020. "Intrusion Detection in IoT Networks Using Deep Learning Algorithm" Information 11, no. 5: 279. https://doi.org/10.3390/info11050279.

[9] Rezvy S, Luo Y, Petridis M, Lasebae A, Zebin T (2019) An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks. In: 2019 53rd annual conference on information sciences and systems (CISS), IEEE, pp 1–6.

[10] Y. Otoum, D. Liu, A. Nayak, DL-IDS: a deep learning–based intrusion detection framework for securing IoT. Transactions on Emerging Telecommunications Technologies, Volume 33, Issue 3, March 2022 pp e3803 https://doi.org/10.1002/ett.3803.

[11] A. Diro and N. Chilamkurti, "Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications," in IEEE Communications Magazine, vol. 56, no. 9, pp. 124-130, Sept. 2018, doi: 10.1109/MCOM.2018.1701270.

[12] Kalaivani K., , and Chinnadurai M. "Ensemble Deep Learning Intrusion Detection Model for Fog Computing Environments," International Journal of Software Innovation (IJSI) 10, no.1: 1-14, 2022. http://doi.org/10.4018/IJSI.303587.

[13] Cristiano Antonio de Souza, Carlos Becker Westphall, Renato Bobsin Machado, Leandro Loffi, Carla Merkle Westphall, Guilherme Arthur Geronimo, Intrusion detection and prevention in fog based IoT environments: A systematic literature review, Computer Networks, Volume 214, 2022, 109154, ISSN 1389-1286.

[14] Illy, Poulmanogo & Kaddoum, Georges & Miranda, Christian & Kaur, Kuljeet & Garg, Sahil, Securing Fog-to-Things Environment Using Intrusion Detection System Based On Ensemble Learning, IEEE Wireless Communications and Networking Conference (WCNC), 2019. 10.1109/WCNC.2019.8885534.

[15] Sahar, Nausheen, Mishra, Ratnesh, Kalam, Sidra, Deep Learning Approach-Based Network Intrusion Detection System for Fog-Assisted IoT, Proceedings of International Conference on Big Data, Machine Learning and their Applications, Springer Singapore, pages 39--50, 2021, doi: //doi.org/10.1007/978-981-15-8377-3_4.

[16] A. Abeshu and N. Chilamkurti, "Deep Learning: The Frontier for Distributed Attack Detection in Fog-to-Things Computing," in IEEE Communications Magazine, vol. 56, no. 2, pp. 169-175, Feb. 2018, doi: 10.1109/MCOM.2018.1700332.

[17] Zhang, K., Song, X., Zhang, C. et al. Challenges and future directions of secure federated learning: a survey. Front. Comput. Sci. 16, 165817 (2022). https://doi.org/10.1007/s11704-021-0598-z.

[18] R.H. Hwang, M.C. Peng, V.L. Nguyen, Y.L. Chang, An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level. Appl. Sci. 2019, 9, 3414. https://doi.org/10.3390/app9163414.

[19] S.Rezvy, M.Petridis, A.Lasebae, T.Zebin. (2019). Intrusion Detection and Classification with Autoencoded Deep Neural Network. In: Lanet, JL., Toma, C. (eds) Innovative Security Solutions for Information Technology and Communications. SECITC 2018. Lecture Notes in Computer Science(), vol 11359. Springer, Cham. https://doi.org/10.1007/978-3-030-12942-2.

[20] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," in IEEE Access, vol. 8, pp. 165130-165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[21] https://www.analyticsvidhya.com/blog/2022/06/understanding-loss-function-in-deep-learning/ (consulted in 25-11-2022).