Towards the Implementation of a Dynamic IDS for IoT: Anomaly Detection in MQTT Traffic

Abdoulaye Diallo[1](✉), Lionel Affognon[2], Chérif Diallo[1], and Eugène C. Ezin[2,3]

[1] Department of Informatique, UFR des Sciences Appliquées et de Technologies, Université Gaston Berger, Saint-Louis, Senegal
{diallo.abdoulaye8,cherif.diallo}@ugb.edu.sn

[2] Institut de Mathématiques et de Sciences Physiques (IMSP), Université D'Abomey-Calavi, Dangbo, Benin
lionel.affognon@imsp-uac.org, eugene.ezin@uac.bj

[3] Institut de Formation et de Recherche en Informatique, University d'Abomey-Calavi, Abomey-Calavi, Benin

Abstract. The proliferation of IoT is evident in numerous domains today. Its applications are expanding rapidly, offering significant benefits. Nonetheless, it is plagued by numerous security flaws that can hinder its adoption in critical areas. This paper addresses the development of an intrusion detection system (IDS) for IoT networks. Specifically, we implement an autoencoder, an unsupervised deep learning model frequently employed in anomaly detection. The model is trained using the MQTTSet dataset, which contains both normal MQTT traffic and attack data. Training focused exclusively on legitimate data. Testing was conducted on both benign and malicious data to assess the model's effectiveness. The results indicate detection rates of 99.86% and 98.56% for normal and attack data, respectively.

Keywords: Internet of things · intrusion detection system · anomaly detection · deep learning · autoencoder

1 Introduction

The Internet of Things (IoT) represents a major technological advancement, connecting the physical and digital worlds. IoT devices collect, exchange, and analyze data to automate and improve processes in sectors such as healthcare, industry, agriculture, and smart cities, enhancing operational efficiency and enabling real-time, data-driven decision-making [1].

IoT devices use various sensors and communication technologies (e.g., Wi-Fi, Bluetooth, Zigbee) to monitor and interact with their environment. They transmit data to servers for real-time analysis and alerts. IoT's potential lies in transforming raw data into actionable insights, applicable in supply chain management, remote health monitoring, industrial automation, energy management, and smart home [2].

K. Cheikh M. F. Kebe et al. (Eds.): InterSol 2024, LNICST 610, pp. 183–193, 2025.
https://doi.org/10.1007/978-3-031-86493-3_15

Despite benefits, increased interconnectivity presents significant security challenges. IoT devices are vulnerable to threats like data theft and service disruption due to the lack of unified security standards and diverse Technologies [3]. Data privacy is a major concern, as IoT devices often collect sensitive information. Inadequate security practices, like default passwords, exacerbate these risks.

These vulnerabilities stem from neglecting security during IoT device design. Many devices are sold with exploitable default settings. Manufacturers must adopt secure design practices to minimize vulnerabilities. However, IoT devices' limited computing and storage capacities make embedding sophisticated security mechanisms challenging. Innovative security solutions, like AI-based technologies (machine learning and deep learning), offer advanced capabilities to detect and respond to sophisticated attacks by analyzing network traffic for anomalies [4].

Anomaly detection in IoT networks is crucial for ensuring system integrity. Researchers are developing intrusion detection systems using machine learning and deep learning. Classical algorithms (e.g., SVM, KNN) are less effective with large data volumes typical of IoT. Deep learning models are more suitable for massive data volumes. Supervised learning models are precise in detecting known intrusions but fail against "zero-day attacks." Unsupervised learning models are more effective in detecting new attacks.

Message Queuing Telemetry Transport (MQTT), an OASIS standard for IoT messaging, is a lightweight publish/subscribe protocol designed for efficiently connecting remote devices with minimal code and network bandwidth [5, 6]. In this paper, we propose an anomaly detection method in MQTT-based IoT network using autoencoders, unsupervised deep learning models. This approach characterizes normal system functioning, with deviations considered anomalies, for anomaly detection in IoT networks.

The remainder of this paper is structured as follows: Sect. 2 reviews relevant literature, Sect. 3 outlines the methodology, including the deep learning model and dataset, Sect. 4 discusses the results and evaluates the model's performance, and Sect. 5 concludes the paper.

2 Related Works

Numerous researchers are investigating IDS for IoT networks [7], with increasing focus on machine learning and deep learning approaches. Traditional machine learning algorithms like Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) are effective with smaller datasets but struggle with larger volumes typical in IoT. Deep learning models, however, improve with more data, making them suitable for IoT applications. Deep learning models can be categorized into supervised and unsupervised learning. Supervised models excel in detecting known intrusions and are widely used for traffic classification to identify attacks.

For instance, a study by authors [8] implemented a Deep Neural Network (DNN) for MQTT-based intrusion detection, trained and tested on MQTT-IoT-IDS2020 dataset. They achieved high accuracy in detecting various attacks. Another study by E. Ciklabakkal et al. [9] used the PyOD library to implement several machine learning models

for anomaly-based IDS, focusing on MQTT attacks. The OCSVM model performed best for normal data, while K-Means and Random Forest excelled with attack data.

N. Elsayed et al. [10] proposed a smart home IDS using a hybrid deep learning model combining BiLSTM and CNN, demonstrating effective detection of network anomalies. Unsupervised learning models, such as autoencoders, are particularly adept at detecting new, unknown attacks, which prompted our focus on these models.

In this context, [11] implemented an autoencoder for anomaly detection in a smart home system. A. Legrand et al. [12] compared convolutional and recurrent autoencoders for anomaly detection in connected buildings, finding recurrent autoencoders to be more effective. A. Dawood et al. [13] conducted a comparative study between autoencoders and Restricted Boltzmann Machines (RBM) for anomaly detection, concluding that autoencoders performed better. I. Apostol et al. [14] proposed an unsupervised deep learning approach using autoencoders to detect IoT botnet activities, demonstrating the model's effectiveness under various conditions.

3 Methodology

Autoencoders, when used for anomaly detection, are trained on normal system patterns. Initially, we segregated normal traffic data from abnormal traffic and trained the autoencoder exclusively on normal data, allowing it to reconstruct normal patterns with minimal error. Anomalies result in higher reconstruction errors, exceeding a predefined threshold.

3.1 Dataset

We utilized the MQTTSet dataset [15], comprising IoT traffic data primarily using the MQTT protocol. This dataset was chosen for its inclusion of common IoT attacks like DoS and brute force, and its manageable number of features. The MQTT protocol is widely adopted for sensor or event data communication. MQTTSet captures traffic during various attack scenarios, making it suitable for our study.MQTT is a widely used publish–subscribe-based protocol for the communication of sensor or event data.

The scenario is assimilated to a smart home environment composed by 8 sensors such as temperature, light intensity, humidity, CO-Gas, motion, smoke, motion sensors and fan speed controller.

The traffic is captured from the broker. During the attack phases, the malicious node is directly connected to the broker in order to execute the cyber-attacks. The following picture show sensors' location (Fig. 1).

MQTTset includes the following types of attacks: DoS, MQTT Publish flood, SlowITe, malformed data, and brute force authentication. The following figure presents the taxonomy of MQTTSet attacks (Fig. 2).

The dataset contains over 12 million records. Benign data represents 98% of the dataset. As we can see MQTTSet dataset is strongly unbalanced. But it is very suitable for our use case.

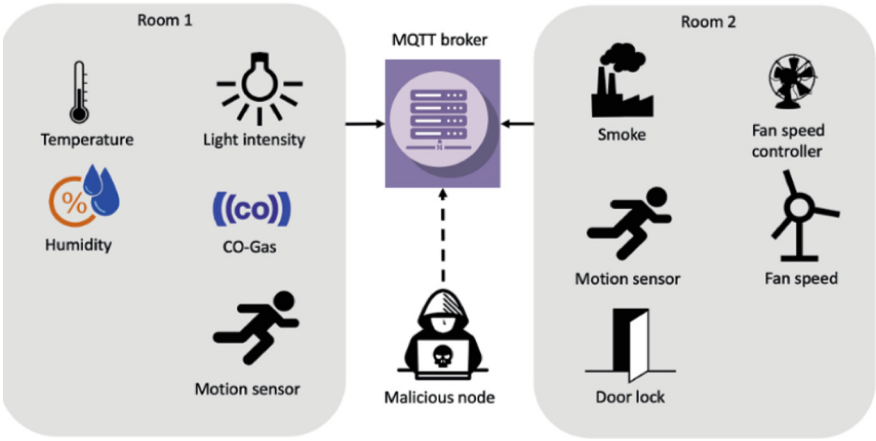

Fig. 1. The scenario considered in MQTTset [15]

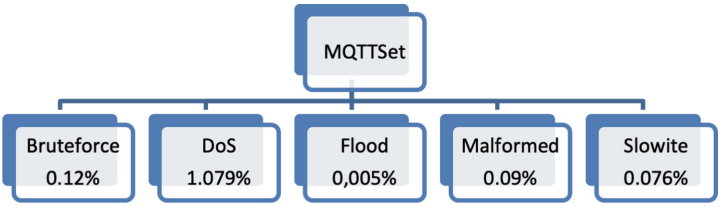

Fig. 2. MQTTSet attacks taxonomy

3.2 Autoencoder and Anomaly Detection

Autoencoders [17], a type of artificial neural network, are frequently employed in anomaly detection due to their ability to learn compact data representations and accurately reconstruct inputs similar to the ones they have been trained on. Typically, autoencoders are trained on data deemed normal, which represents typical or healthy behavior within a system. Their advantage lies in their capacity to learn meaningful data representations without explicit supervision for anomalies. The model inherently learns to compactly represent normal data in the latent space, making anomalies easier to detect due to their deviation from this learned representation (Fig. 3).

The model reconstructs the input ($\mathcal{X}$) into its output ($\mathcal{X}\prime$). Thus, the model is trained by minimizing reconstruction errors, which are defined as a loss ($\mathcal{L}oss$) function as follows:

$$\mathcal{L}oss(\mathcal{X}, \mathcal{X}\prime) = \|\mathcal{X} - \mathcal{X}\prime\|^2 \tag{1}$$

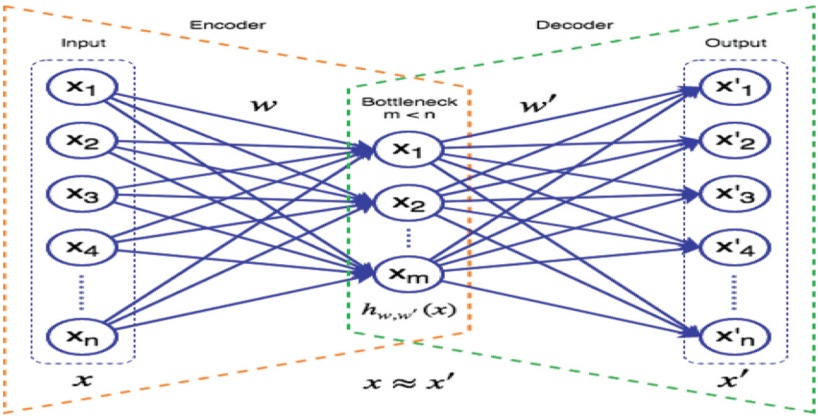

Fig. 3. Autoencoder architecture, image source [18]

During training, autoencoders minimize a loss function that measures the difference between the input and the model's reconstruction. Once trained, these autoencoders can be used to reconstruct new inputs and evaluate the quality of the reconstruction by calculating an error measure, typically based on the distance or divergence between the original inputs and their reconstructions. To determine whether an input is normal or abnormal, an error threshold is often established. Inputs with reconstruction errors exceeding this threshold are classified as abnormal, whereas those with errors below the threshold are considered normal.

The following equations demonstrates how the autoencoder model predicts the nature of an input:

$$x_{error} = \mathcal{L}oss(\mathcal{X}, \mathcal{X}\prime) \tag{2}$$

$$predictor(x) = \begin{cases} Normal, x_{error} < Threshold \\ Anomaly, x_{error} \geq Threshold \end{cases} \tag{3}$$

The following algorithm gives more details on anomaly prediction by autoencoders.

Algorithm 1: Anomaly detection by autoencoder

```
1: Function predictor(x,threshold,autoencoder):
2:     x' ← AE.predict(x)
3:     x_error ← calculate_error(x, x')
4:     if x_error < threshold then
5:           return "Normal"
6:     else
7:     return "Anomaly"
8: end

1: Function calculate_error(x, x'):
2:     x_error ← ||x - x'||^2
3:           return x_error
4: end
```

Choosing the appropriate error threshold is crucial and can be influenced by various factors, such as the specific nature of the anomaly detection problem, the data distribution, and the requirements for performance and tolerance to false positives and false negatives. In practice, this anomaly detection approach with autoencoders can be effective in numerous fields, including cybersecurity, fraud detection, predictive maintenance, and industrial system monitoring. It is important to note that the success of anomaly detection using autoencoders depends on several factors, including the quality and representativeness of the training data, the appropriate choice of model architecture and hyperparameters, and the careful selection of the error threshold. Additionally, it is often necessary to periodically reevaluate and adjust these parameters to maintain the model's performance in dynamic environments or when facing sophisticated attacks.

3.3 Our Autoencoder Architecture

The process of constructing the autoencoder model was meticulous and strategic, involving several key steps to achieve an optimal configuration. Initially, we conducted an exhaustive search for the best hyperparameters for the loss function and optimizer. This initial phase was crucial in determining the fundamental parameters that directly influence the model's outcomes. For this search, we utilized GridSearchCV.

After identifying the Mean Squared Error (MSE) loss function and the Adam optimizer as the best choices, we proceeded to determine the optimal structure of the autoencoder. This phase involved exploring different configurations for the number of neurons in each layer of the autoencoder, as well as the bottleneck. Again, we used GridSearchCV to systematically explore various parameter combinations and identify the optimal configuration.

Upon analyzing the results of this comprehensive search, we identified an optimal autoencoder configuration with 128 neurons in the first and seventh hidden layers, 64 neurons in the second and sixth hidden layers, 32 neurons in the third and fifth hidden layers, and 16 neurons for the latent space or bottleneck. These parameters were selected considering the dataset's complexity and the model's generalization capability.

Finally, with the best hyperparameters identified, we proceeded to build the autoencoder model using the Keras library. Each layer was configured according to the number of neurons determined during the hyperparameter search, and the ReLU and sigmoid activation functions were chosen based on the nature of the data and modeling objectives. Compiled with the Adam optimizer and the MSE loss function, the model is now ready to be trained and tested.

$$MSE = \frac{1}{n}.\sum_{i=1}^{n}(X_i - X\prime_i)^2 \tag{4}$$

X is a feature vector provided as input to the autoencoder, and $X\prime$ is the reconstructed output.

4 Results and Discussion

After the training phase, we performed two tests. The first test is done on normal data and we did the second test on attack data. We tested different thresholds but the best of them was obtained with the formula below:

$$Threshold = Mean(train\_loss) + Std(train\_loss) \tag{4}$$

The results are illustrated in the following figures. The abscissas represent the features and the ordinates are the values of the features.

The previous figures (Fig. 4a, Fig. 4b, Fig. 4c) present three examples of normal input data and their corresponding reconstructed output. They show the ability of the model to reconstruct normal data. The model achieves a perfect reconstruction of the normal data. The error rate at this level is very low. Its accuracy on detecting normal data is 99.86%.

On the other hand, the attack data is poorly reconstructed. The figures below show the behavior of the model on attack data (DoS, Bruteforce, Malformed, Slowite, Flood) (Figs. 5, 6, 7, 8 and 9).

By observing the previous figures which show the behavior of the model on attack data, we can see that the reconstruction error is very large. It exceeds the threshold that has been defined. Thus, there is a big difference between the reconstruction error of normal data and that of attack data. Therefore, these data are considered malicious. An alert can therefore be launched to report an intrusion. The detection rate of attack data is 98.56%

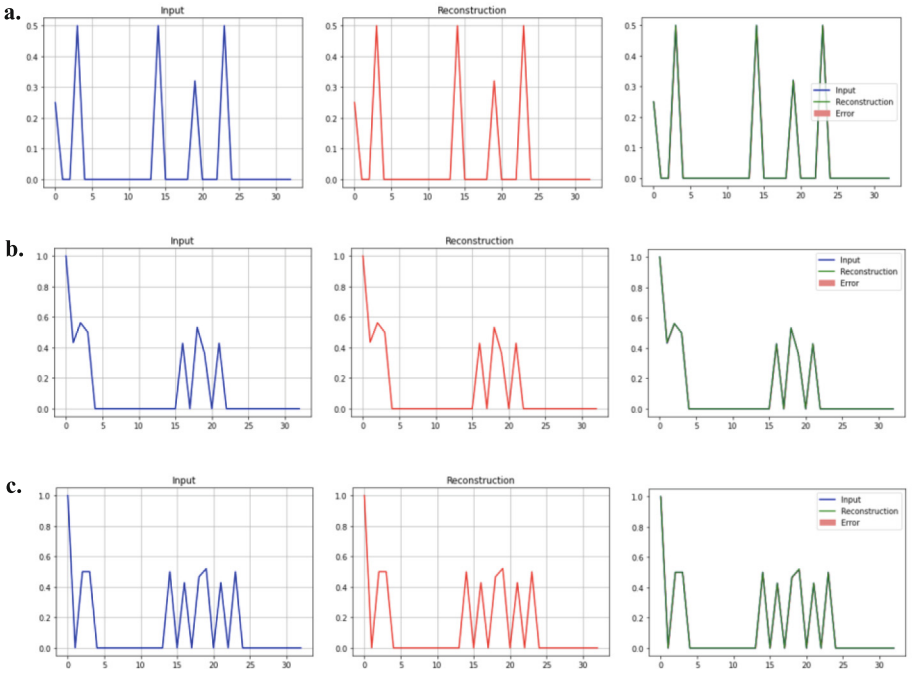

Fig. 4. **a.** Normal input and reconstructed output. **b.** Normal input and reconstructed output. **c.** Normal input and reconstructed output.

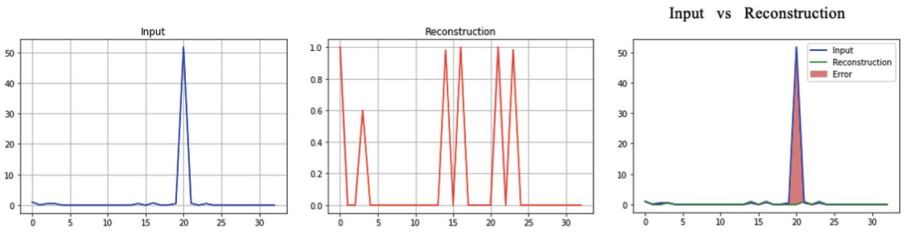

Fig. 5. Here the input is DoS attack data

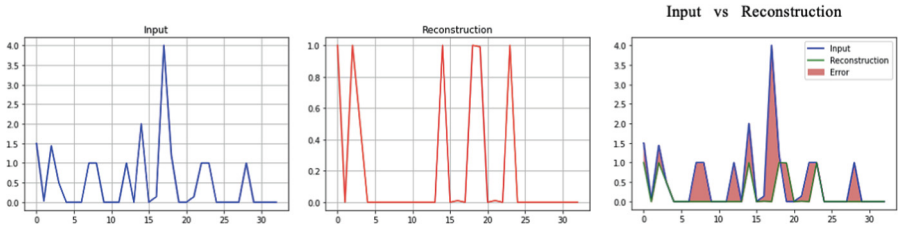

Fig. 6. When the model receives bruteforce attack data as input

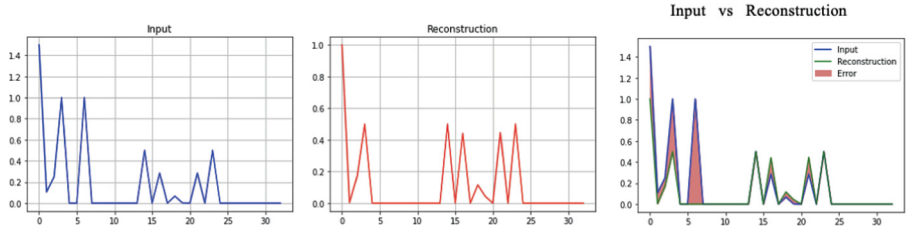

Fig. 7. When the input is malformed data

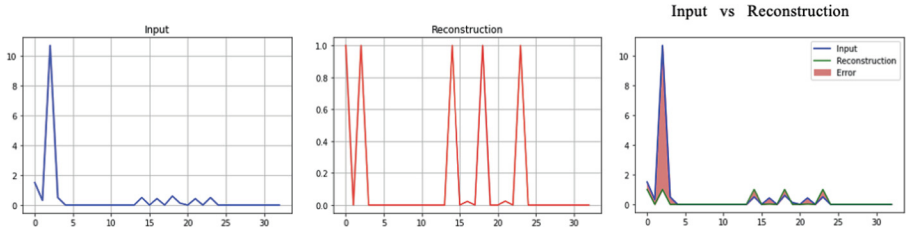

Fig. 8. The model receives Slowite attack data as input

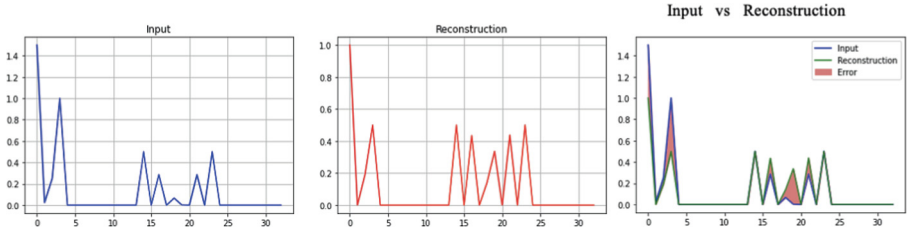

Fig. 9. This input is Flood attack data

5 Conclusion and Future Works

This paper presents an autoencoder-based IDS for IoT networks using MQTT protocol data, capable of detecting zero-day attacks. The model achieved high accuracy in distinguishing normal and malicious traffic. The model achieves an accuracy of 99.86% in recognizing normal traffic. It scores 98.56% in detecting malicious traffic. However, it must be said that autoencoders can have poor performance if the training data is not representative of normal system traffic. In our next work we intend to go further than in the detection of intrusions. Future work will focus on combining this model with others to not only detect but also identify specific intrusions.

Acknowledgments. This publication was made possible through the DSTN (Digital Science and Technology Network) supported by IRD (Institut de Recherche pour le Développement) and AFD (Agence Française de Développement).

References

1. Ramson, S.R.J., Vishnu, S., Shanmugam, M.: Applications of İnternet of Things (IoT)–an overview. In: 2020 5th İnternational Conference on Devices, Circuits and Systems (ICDCS). IEEE (2020)
2. Diallo, A., Diallo, C.: Human activity recognition in smart home using deep learning models. In: 2021 IEEE International Conference on Computational Science and Computational Intelligence (CSCI) – Las Vegas, USA. pp. 1721–1727 (2021). https://doi.org/10.1109/CSCI54926.2021.00326
3. OWASP IoT Top 10. https://owasp.org/www-chapter-toronto/assets/slides/2019-12-11-OWASP-IoT-Top-10---Introduction-and-Root-Causes.pdf
4. Hamed, T., Ernst, J.B., Kremer S.C.: A survey and taxonomy of classifiers of ıntrusion detection systems. Computer and Network Security Essentials (2018)
5. Nikolov, N.: Research of MQTT, CoAP, HTTP and XMPP IoT communication protocols for Embedded Systems. In: 2020 XXIX International Scientific Conference Electronics (ET), Sozopol, Bulgaria, pp. 1–4 (2020) https://doi.org/10.1109/ET50336.2020.9238208
6. Soni, D., Ashwin, M.: A survey on MQTT: a protocol of Internet of Things (IoT). In: International Conference on Telecommunication, Power Analysis and Computing Techniques (ICTPACT-2017) (2017)
7. Sekhar, C., Pavani, K., Rao, M.S.: Comparative analysis on ıntrusion detection system through ML and DL Techniques: Survey. In: 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA), Nagpur, India, pp. 1–5 (2021). https://doi.org/10.1109/ICCICA524589697291
8. Khan, M.A., et al.: A deep learning-based intrusion detection system for MQTT enabled IoT. Sensors **21**(21), 7016 (2021)
9. Ciklabakkal, E., Donmez, A., Erdemir, M., Suren, E., Yilmaz, M.K., Angin, P.: Artemis: an intrusion detection system for mqtt attacks in İnternet of Things. In: 38th Symposium on Reliable Distributed Systems (SRDS), pp. 369–3692 (2019)
10. Elsayed, N., Zaghloul, Z.S., Azhuma, S.W., and Li, C.: Intrusion detection system in smart home network using bidirectional lstm and convolutional neural networks hybrid model. In: 2021 IEEE Interna- tional Midwest Symposium on Circuits and Systems (MWSCAS), pp. 55– 58 (2021)
11. Cultice, T., Ionel, D., and Thapliyal, H.: Smart home sensor anomaly detection using convolutional autoencoder neural network. In: 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), pp. 67–70 (2020)
12. Legrand, A.,Niepceron, B., Cournier. A, Trannois, H.: Study of autoencoder neural networks for anomaly detection in connected buildings. In: 2018 IEEE Global Conference on Internet of Things (GCIoT), pp. 1–5 (2018). https://doi.org/10.1109/GCIoT.2018.8620158
13. Dawoud, A., Sianaki, O.A., Shahristani, S., Raun, C.: Internet of Things intrusion detection: a deep learning approach. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1516–1522 (2020)
14. Apostol, I., Preda, M., Nila, C., Bica, I.: IoT botnet anomaly detection using unsupervised deep learning. Electronics **10**(16), 1876 (2021)
15. Vaccari, I., Chiola, G., Aiello, M., Mongelli, M., Cambiaso, E.: MQTTset, a new dataset for machine learning techniques on MQTT. Sensors (2020)
16. Hindy, H., Bayne, E., Bures, M., Atkinson, R., Tachtatzis, C., Bellekens, X.: Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset) (2021)
17. Sewak, M., Sahay, S.K., Rathore, H.: An overview of deep learning architecture of deep neural networks and autoencoders. J. Comput. Theor. Nanosci. **17**(1). American Scientific

Publisher, pp. A82–188 (2020). https://subscription.packtpub.com/book/python/9781789348460/6/ch06lvl1sec37/variational-autoencoders

18. Diallo,A., Affognon,L., Diallo, C., Ezin, E.C.: Deep learning based binary and multi-class classification for anomaly detection. In: 2022 International Conference on Engineering and Emerging Technologies ICEET, Kuala Lumpur, Malaysia, pp. 1–6 (2022)