Hybrid Deep Learning for Botnet Attack Detection in the Internet of Things Networks

Sk.Nasreen, Department of CSE, Narayana Engineering College, Gudur.
Ms.Dr.V.Sucharitha, Department of cse,Narayana Engineering College Gudur

Abstract :The rapid proliferation of Internet of Things (IoT) devices has led to increased security vulnerabilities, particularly in the form of botnet attacks. These attacks can compromise the functionality and security of IoT networks, necessitating robust detection mechanisms. This paper proposes a hybrid deep learning approach for botnet attack detection in IoT networks. Byintegrating Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, the proposed model leverages both spatial and temporal features of network traffic data. The performance of the hybrid model is evaluated using a well-known IoT dataset, demonstrating superior detection accuracy and robustness compared to traditional machine learning methods.

INDEX TERMS— Botnet Iot Attack, LSTM, CNN, Auto encoder.

I.INTRODUCTION

INTERNET OF THINGS (IoT) devices and systems are becoming commonplace and, hence, they are increasingly targeted by attackers, for example, by identifying and exploiting vulnerabilities in IoT software and hardware, or their implementation, to facilitate unauthorized and malicious activities. Such devices have also been exploited to create a botnet network to generate Distributed Denial-of Service (DDoS) traffic. DDoS represents a critical network oriented cyberthreat, whose trend has been steadily rising over the last decade [1], [2]. For example, the DDoS attacks targeting Amazon AWS in Q1 of 2020 reportedly had a peak volume of 2.3 Tbps

IoT devices and systems are found not only in an organizational or government setting but also in our homes (e.g., smart homes). Smart homes are one of the fastest-growing IoT applications, and the deployed devices are extremely heterogeneous. Such devices are often shipped with minimal or nonexistent security mechanisms, and in an effort to make these devices user friendly, the security requirements are often reduced [4]. In addition, most of the devices in a smart home are inexpensive and do not have significant computational capabilities and, consequently, they can be easily compromised to facilitate a broad range of nefarious activities, including generating DDoS traffic [5]. In a typical smart home ecosystem, there are several stakeholder groups, such as end users (homeowners or tenants within a home)

Internet/telecommunication service providers, device manufacturers, and service providers (e.g., third-party service providers such as a monitored security service). For example, it is in the interest of Internet/telecommunication service providers to promptly detect any unauthorized behavior/activities within a smart home environment, to protect their own network infrastructure and prevent the compromised devices/systems to be used as a launch pad against other devices and systems (with associated legal and financial implications).

II. FUNCTIONAL OVERVIEW

Boosting-based DDoS (Distributed Denial of Service) detection in Internet of Things (IoT) systems leverages machine learning techniques to identify and mitigate malicious traffic aimed at overwhelming IoT devices and networks. Boosting is an ensemble learning technique that combines the outputs of multiple weak learners to create a strong classifier. Here's a functional overview of how boosting-based DDoS detection works in IoT systems:

Key Components and Functional Flow

1. Data Collection and Preprocessing:

Data Sources: Network traffic data, including packet headers, payloads, and flow statistics, are collected from IoT devices and network gateways.

Feature Extraction: Relevant features are extracted from the raw data. These features might include packet size, frequency, source/destination IP addresses, and time intervals between packets. **Data Cleaning**: Noise and irrelevant data are removed to ensure quality and accuracy.

2. Training Data Preparation:

Labeling: The collected data is labeled as either normal or malicious (DDoS attack) using historical data or through manual annotation.

Data Splitting: The labeled dataset is split into training and testing sets to evaluate the model's performance.

3. Boosting Algorithm:

Weak Learners: Simple classifiers (weak learners) such as decision trees are trained on subsets of the data. These learners individually may have low accuracy.

Iterative Training: Boosting algorithms, such as AdaBoost or Gradient Boosting, iteratively train these weak learners, each time focusing on the samples that were misclassified in previous iterations.

Weighting: In each iteration, misclassified samples are given higher weights, prompting subsequent weak learners to focus more on these difficult cases.

Combining Learners: The outputs of the weak learners are combined (e.g., through a weighted majority vote) to form a single strong classifier.

4. Model Deployment:

Real-time Monitoring: The trained boosting model is deployed in the IoT network to monitor incoming traffic in real-time.

Anomaly Detection: The model continuously analyzes traffic patterns and flags anomalies that resemble DDoS attack signatures.

Alerting and Mitigation: When a potential DDoS attack is detected, alerts are generated, and automated mitigation strategies are triggered (e.g., traffic filtering, rate limit.

III. DESIGN

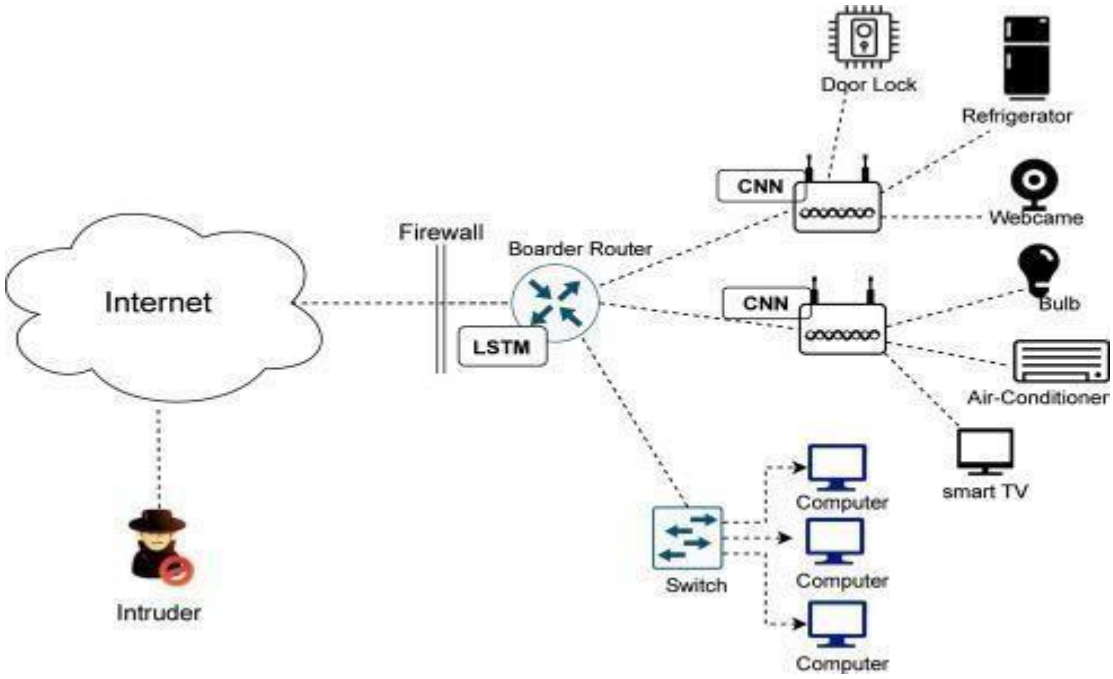


Fig. 1. Smart home testbed.

The underpinning communication infrastructure and software-hardware platform are also setup to enable traffic collection that can be used to train DDoS detection models. In addition to the primary data collected in this research, we also used secondary data from [23], including a larger number of various SHIoT devices (i.e., greater device heterogeneity). The Fortinet AP 221C wireless access point, the Cisco 2960 Catalyst 48 Power over Ethernet (PoE) switch, the HP Pavillion dm1, and Microsoft HP 10 10.0.17134 build 17 134 workstations have been set up to capture traffic using port mirroring, x64 processor architecture, AMD E-350, 1600-MHz two cores, 4-GBRAM).

Testbed Setup

The setup of our smart home laboratory environment is shown in Fig. 1, and also presented in [24]. These ports are setup as sources, which ensures that all traffic to and from them are mirrored (mapped) to the destination contact port (FA0/2). A traffic collection workstation is connected to this port a legitimate traffic profile of a SHIoT device, it is crucial to have a data set that includes DDoS traffic. These two sets form the basis for developing an effective model for detecting network traffic anomalies such as DDoS traffic generated by SHIoT devices.

Given that legitimate traffic comes from the primary and secondary sources, where the author does not have access to the secondary source devices, a key challenge is the manipulation of SHIoT devices to generate DDoStraffic. Therefore, in this research, for generating DDoS traffic BoNeSi (the open- source software tool) was used [25]. The virtual workstation was used to generate DDoS traffic and create a data set of illegitimate traffic. The virtual workstation's configuration is as follows: Linux Ubuntu 19.04 operating system with dedicated 4 GB of RAM, an Intel Core i7-5500U processor (4x2.40 GHz).

In Fig. 1, the virtual machine and BoNeSi tool denote the SHIoT device in the local smart home network generating DDoS traffic. For practical reasons, the BoNeSi tool was used to simulate illegitimate traffic generated by the SHIoT device in order to minimize the risk of compromising the real device. BoNeSi is not just a network traffic generator (as the tool's documentation suggests), it is also a powerful and efficient DoS and DDoS generator and simulator tool. Hence, our choice for using it to simulate traffic similar to those generated by an individual SHIoT device as part of a botnet. In addition, the illegitimate traffic was generated in an isolated environment to avoid breaking the laws of the Republic of Croatia, the European Union, and the United States.

A. Defining Legitimate Traffic Profiles for Classes of SHIoT Device

As discussed earlier, SHIoT is a dynamic and ubiquitous environment, where new consumer IoT devices with different functionalities are constantly introduced to the market. Therefore, new, unknown SHIoT devices may have functionalities different from those of the currently available SHIoT devices.This presents a challenge in identifying such devices and knowing their legitimate behavior, which forms the basis for detecting behavioral anomalies such as generating DDoS traffic

IV. WORKING

Working Principle of the Developed Model for Detection of Illegitimate DDoS Network Traffic. The work of the developed model of illegitimate DDoS traffic detection takes place in two phases. The first is a prerequisite for later detection of DDoS traffic in the second phase and involves the classification of SHIoT devices based on generated traffic flow. The multiclass classification model results show that the SHIoT device can be classified into one of the four predefined classes concerning the traffic flows it generates with an accuracy of 99.79%.

In doing so, the values of traffic flow classified into certain predefined classes also become part of the profile of legitimate traffic of these classes. Depending on the corresponding class of SHIoT devices, an individual LMT model can detect devia- tions or anomalies from the existing normal traffic profile with high accuracy (LMT-C1 99.99%, LMT-C2 99.92%, LMT- C3 99, 97%, and LMT-C4 99.95%) and using different sets of independent traffic flow characteristics.

RESULTS ANALYSIS AND DISCUSSION:

The development of a DDoS detection model based on traf- fic characteristics and device class indicates the importance of recognizing the class to which the SHIoT device belongs as a fundamental activity of further recognizing anomalies in network traffic such as DDoS traffic. According to the model presented in the previous section, it is clear that not all inde- pendent features are equally important in detecting anomalies for a particular class. Likewise, certain features in one class may be relevant while viewed from the aspect of another class they do not have to An example is seer each class differs according to the num- ber of relevant independent features, and it is also evident that the same features are not relevant in the detection of anomalies for each class.

Furthermore, an individual independent feature's thresh- old value that determines the decision tree's branching dif- fers for individual classes. As shown in Figs. 3 and 4, branching in the decision tree occurs based on the thresh- old value of the feature z24, representing the standard deviation of the
The interarrival packet times in the observed traffic flow expressed in microseconds (Âµs). In doing so, the algo- rithm C4.5 is used, which selects the threshold value of the independent feature that allows the purest division of the feature vector in the set [29]. Thus, for example, the threshold value of the z24 feature in the LMT model for class C2 differs from the threshold value of the same feature for class C4.

To evaluate the behavior of the model over data not included in the learning process, each version of the LMT model was validated using the k-fold cross-validation approach with k 10. Cross-validation is a mathematical technique for evaluating the success of machine learning models on new, unknown data. This approach is used to test the model's output on data that was not used during the learning pro- cess. The model is iteratively extended k times over the data set in this way. The data set is split into k sections in each iteration.

The remaining k 1 portions of the set are grouped into a subset for model learning, while one part of the set is used to test the model [30]. Validation metrics (accu- racy, kappa statistics, true-positive rate (TPR), falsepositive rate (FPR), precision, F-measure, ROC-Receiver Operating Characteristics, and PRC-PrecisionRecall Curve) are often used to test machine learning classification models.

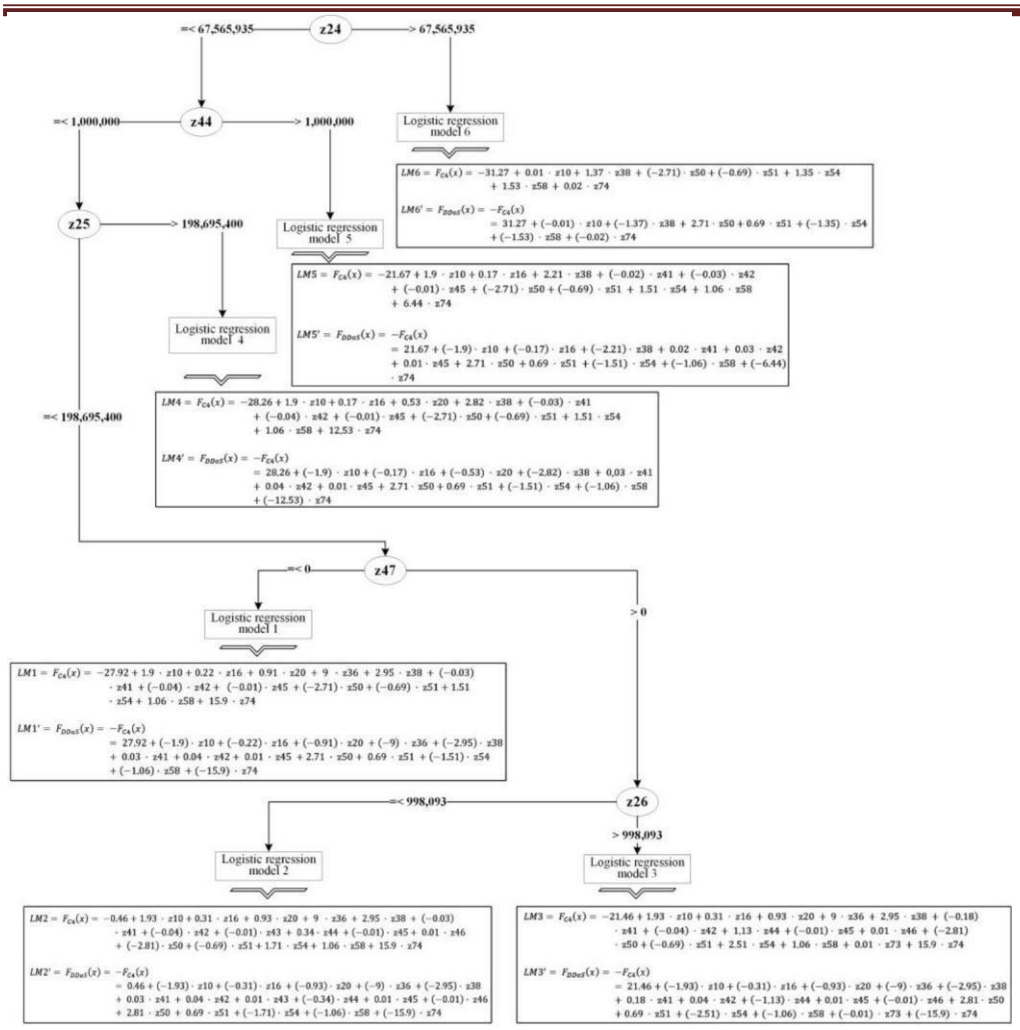


Fig. 2. LMT model of the DDoS detection model for class C4.

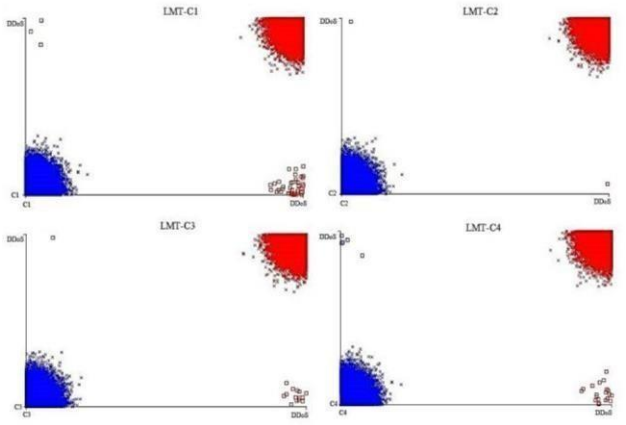


Fig. 3. Error visualization of LMT classification models for the corresponding classes.

A. Accuracy of Developed LMT Classification Models

True-positive (TP) examples, true-negative (TN) examples, false-positive (FP) examples, and falsenegative (FN) examples reflect the share of correctly classified examples in the set of all examples

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

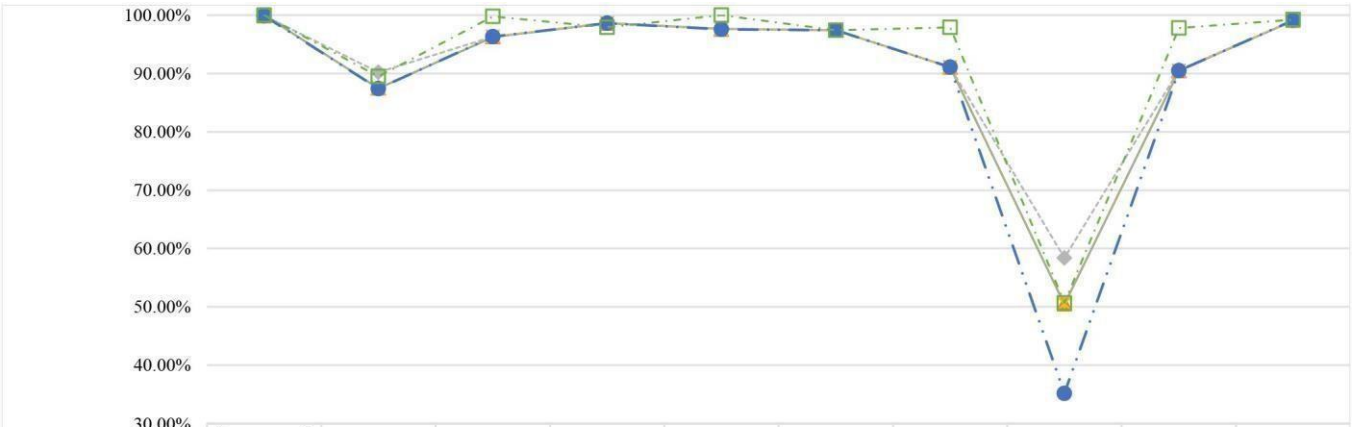
Acc proportion of accurately classified examples in the set of all examples;
TP number of true positive examples;
TN number of true negative examples;
FP number of false positive examples; FN
number of false negative examples.
According to the classification's accuracy, all four mod- els show high performance, which means that based on the observed flow, they can determine with high accuracy whether the traffic flow is the result of legitimate communication of the device, or the device generates DDoS traffic. According to Table VI, the high accuracy of all four versions of the LMT model developed for each class of SHIoT devices can be observed. Errors in the classification of all four versions of the LMT model were visualized and shown in Fig. 3.

Fig. 3 shows that the detection model is most accurate for class C2 and the lowest performance is observed in the LMT- C1 model. From the given figure, it is observed that errors for all four models are prevalent for classifying DDoS traffic instances, indicating the need for better modeling of this class in future research.

To more clearly show the accuracy of the classification, a confusion matrix was used for all developed versions of the model. The confusion matrix is a performance metric for machine learning classification models with two or more classes as output, and it serves as the foundation for other metrics. Thus, the LMT model for device class C1 shows an accuracy of 99.9216%, or 56 092 accurately classified traffic flows, as a DDoS or traffic flow that legitimately belongs to a SHIoT device from class C1. A total of 44 traffic flows were misclassified, i.e., 0.0784% in the total set of 56136.

In summary, to evaluate the effectiveness of the LMT method applied in this study, we applied several frequently used machine learning methods over the same data set. Specifically, we compared the performance of our pro- posal with those of multilayer perceptron (MLP), k-Nearest Neighbors (kNN), Random Tree (RT), Bagging, AdaBoostM1, stochastic gradient descent (SGD), dense layer, Recurrent Neural Network (RNN), and GravesLSTM, in terms of accu- racy, TPR, Precision, Recall, F-measure, and ROC.

For imple- menting mentioned methods, we used WekaDeeplearning4j package for WEKA platform [32]. Find the comparision results presented in Fig. 6, one can see that our approach generally outperforms the other applied methods.



	Our approach (LMT)	MLP	kNN	Radom tree	Bagging	AdaBoostM1	Dense layer	RNN	SGD	GravesLSTM
							Deep learning			
Accuracy	99.99%	87.47%	96.26%	98.64%	97.63%	97.41%	91.10%	50.46%	90.50%	99.06%
TPR	0.999	0.875	0.963	0.986	0.976	0.974	0.911	0.505	0.905	0.991
Precission	0.999	0.902	0.963	0.987	0.977	0.974	0.911	0.584	0.906	0.991
Recall	0.999	0.875	0.963	0.986	0.976	0.974	0.911	0.505	0.905	0.991
F-Measure	0.999	0.874	0.963	0.986	0.976	0.974	0.911	0.352	0.905	0.991
ROC	1	0.895	0.998	0.979	1	0.974	0.979	0.507	0.978	0.992

Fig. 6, one can see that our approach generally outperforms the other applied methods.

V. CONCLUSION

The DDoS detection model presented in this article deviates from the typical network traffic anomaly detection approaches. For example, prior approaches are largely based on generating a legitimate traffic profile that is assumed to apply to all terminal devices. Such an approach is logical in environments comprising conventional devices, whose traffic generates characteristics that are reflective of the operation of the installed applications on the devices and the way the users use such devices.

However, inexpensive IoT devices are somewhat limited in terms of their functionality, which is reflected in the characteristics of the traffic they generate. There are also IoT devices which are more computationally capable. Hence, existing non-IoT approaches may not be suitable, partly due to the diversity of IoT devices (and consequently, behavior). In other words, some devices will always generate similar traf- fic, while other devices that are capable of supporting greater interactions with the user may generate traffic that is irregular. Compounding this challenge is the significant growth in the number of devices in an IoT environment.

The problem of detecting DDoS traffic based on device classes has been reduced to binary classification,

where different versions of the same model are developed for each class of SHIoT devices. This is why each class of SHIoT devices' traffic has different characteristics, which is evident from the presented versions of the model, each differing in the number of independent features used, the size of the decision tree and the threshold values of its branching. Our performance evaluation showed that the approach achieves high performance, in terms of accuracy, TPR, FPR, F1 rating, precision, ROC and PRC.

VI. REFERENCES

[1] K. Anoh, A. Ikpehai, D. Bajovic, O. Jogunola, B. Adebisi, D. Vukobra- tovic, and M. Hammoudeh, "Virtual microgrids: a management concept for peer-to-peer energy trading," in Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, 2018, pp. 1–5.

[2] A. O. Akmandor, Y. Hongxu, and N. K. Jha, "Smart, secure, yet energy- efficient, internet-of-things sensors," IEEE Transactions on Multi-Scale Computing Systems, vol. 4, no. 4, pp. 914–930, 2018.

[3] Z. Wang, Y. Liu, Z. Ma, X. Liu, and J. Ma, "Lipsg: Lightweight privacy- preserving q-learning based energy management for the iot-enable smart grid," IEEE Internet of Things Journal, vol. 7, no. 5, pp. 3935–3947, 2020.

[4] R. J. Tom, S. Sankaranarayanan, and J. J. Rodrigues, "Smart energy management and demand reduction by consumers and utilities in an iot- fog-based power distribution system," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 7386–7394, 2019.

[5] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) towards 5g wireless systems," IEEE Internet of Things Journal, vol. 7, no. 1, pp. 16–32, 2020.

[6] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5g era: Enablers, architecture, and business models," IEEE Journal on Selected Areas in Communications, vol. 34, no. 3, pp. 510–527, 2016.

[7] W. Feng, J. Wang, Y. Chen, X. Wang, N. Ge, and J. Lu, "Uav-aided mimo communications for 5g internet of things," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1731–1740, 2018.

[8] P. Verma and S. K. Sood, "Fog assisted-iot enabled patient health monitoring in smart homes," IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1789–1796, 2018.

[9] L. Liu, J. Xu, Y. Huan, Z. Zou, S.-C. Yeh, and L. Zheng, "A smart dental health-iot platform based on intelligent hardware, deep learning and mobile terminal," IEEE journal of biomedical and health informatics, vol. 24, no. 3, pp. 898–906, 2020.

[10] R. K. Pathinarupothi, P. Durga, and E. S. Rangan, "Iot-based smart edge for global health: Remote monitoring with severity detection and alerts transmission," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2449– 2462, 201