Intrusion Detection in IoT Network Security with Deep Neural Network

Nafiz Rifat
Dept. of Computer Science
North Dakota State University
Fargo, ND, USA
nafiz.rifat@ndsu.edu

Mostofa Ahsan
Dept. of Computer Science
North Dakota State University
Fargo, NS, USA
mostofa.ahsan@ndsu.edu

Md. Minhaz Chowdhury
Dept. of Computer Science
East Stroudsburg University
East Stroudsburg, PA, USA
mchowdhur1@esu.edu

Rahul Gomes,
Dept. of Computer Science
University of Wisconsin-Eau Claire
Eau Claire, WI, USA
gomesr@uwec.edu

***Abstract*—This study presents deep neural network learning models for cyber security in IoT (Internet of Things) networks using the CICIDS-2017 dataset. With the growing popularity and use of IoT (Internet of Things), the cybersecurity risk increased exponentially. In the recent past, DDoS (Distributed Denial of Service) attacks have caused damage to many IoT networks, resulting in massive losses. Effective deep learning models for demonstrating cybersecurity knowledge in IoT networks have been explored, including DenseNetwork, CNN, and a hybrid model of CNN and LSTM. Finally, we offer some suggestions for future research studies.**

***Index Terms*—IoT, CICIDS, DDoS, Dense Network, CNN, LSTM**

I. Introduction

IOT- The Internet of Things is modern and one of the most promising technologies that use the internet to connect everything worldwide. IoT technology promises to improve and assist our personal, professional, and societal lives. The Internet of Things typically consists of sensors, actuators, and processors that may connect with one another to achieve shared goals/applications using unique IDs based on the Internet protocol (IP) [1]. IoT gives the convenience of utilizing smart items with less human intervention; however, it is vulnerable to cyber assaults like any other network. Simple endpoint devices are more vulnerable to cyber-attacks within the network than IoT systems. (e.g., home security cam, home appliance) are more constrained in computation, storage, and network capacity than the more complex endpoint devices (e.g., router, smartphones, laptops) within the IoT infrastructure [2]. By compromising and exploiting a significant number of these vulnerable IoT devices, attackers can now launch wide-scale assaults against Internet resources such as spamming, phishing, and Distributed Denial of Service (DDoS) [3], [4]. Moreover, DDoS is a recent cyber-attack that has wreaked on IoT networks, causing significant losses by overloading target servers and crashing the system. This study detecting threat on IoT our CNN, and a hybrid model of CNN and LSTM performed really well; however, DenseNet outperformed with the accuracy of 97.77% on your test data. Therefore, the detection of the DoS alongside other major attacks plays a vital role in the design of our deep neural network-based Intrusion Detection system. This IDS adds an extra layer of security to the company, allowing it to fend off a range of threats. This paper aims to
1) propose a deep learning method for cyber security in IoT networks and
2) assess the suggested models using the CICIDS2017 dataset.
3) Compare the performance of deep learning algorithms to that of machine learning algorithms,

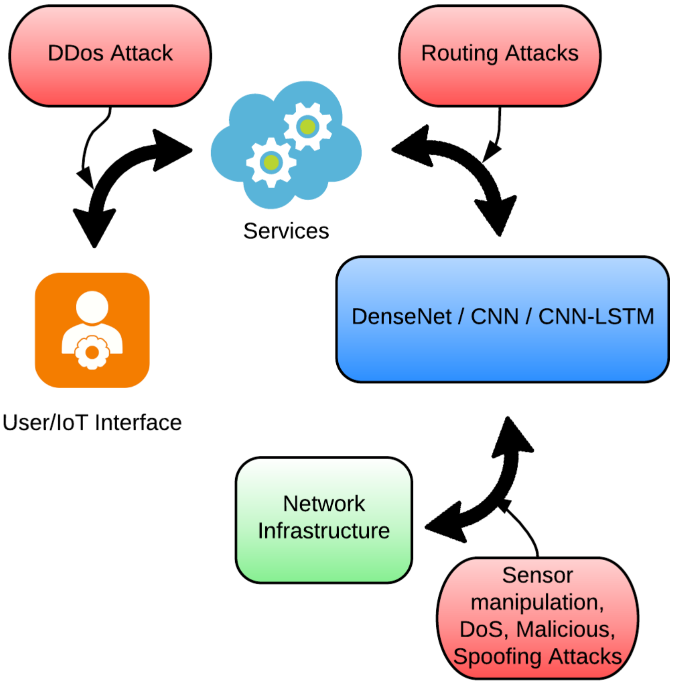
DDos Attack
Routing Attacks
Services
DenseNet / CNN / CNN-LSTM
User/IoT Interface
Network Infrastructure
Sensor manipulation, DoS, Malicious, Spoofing Attacks

Fig. 1: IoT Network Flow

II. Related Work

IoT devices communicate among them using various communication technologies and different kinds of protocols. Because of this heterogeneity, applying cybersecurity approaches to IoT networks presents some issues. In addition, the physical capacity of IoT devices and the volume of data generated by IoT devices make cybersecurity more challenging. Moreover, Deep Learning approaches have been used in fields such as image processing, speech recognition, and healthcare, among others, and have outperformed traditional machine learning methods. Researchers in [5] suggested an ensemble network intrusion detection technique that relies on well-known statistical flow features. IoT services use network protocols including DNS, HTTP, and MQTT, and attackers use tactics like polymorphic code, DNS spoofing, DNS cache poisoning, Denial of Service (DoS), Distributed DoS (DDoS), and URL interpretation to try to exploit weaknesses in these protocols. The primary step of their research entails a thorough examination of the TCP/IP model, followed by selecting a collection of features from the network traffic protocols MQTT, HTTP, and DNS. They used three Machine Learning (ML) algorithms: decision tree (DT), Naive Bayes (NB), and artificial neural network (ANN). Researchers in [6]–[8] propose a hypervisor-level distributed network security framework using the Random Forest classifier technique to identify and classify an attack by evaluating the applied network traffic. The researchers used the UNSW-NB15 dataset and the CICIDS-2017 dataset to conduct their study. The pre-trained Random Forest classifier algorithm outperforms in terms of real-time validation, detecting a range of assaults, quick detection, and high network traffic management. However, the proposed approaches still have a high percentage of false alarms and cannot precisely define a standard profile's baseline. In this study [9], on the CI-CIDS2017 dataset, applied a feature reduction method in IDS with Ada-boost t as a classifier. In addition, they combined the Synthetic Minority Oversampling Technique (SMOTE) [10], Principal Component Analysis (PCA), and Ensemble Feature Selection (EFS) to improve the performance. Moreover, it enables the ML-based IDS to train faster with low computation and high accuracy. As a result, with an accuracy of 81.83 percent and an F1 score of 90.01 percent, the results show that their proposed technique outperforms existing literature for 25 selected features. Researchers in [11] proposed a method using Recurrent Neural Network (RNN) deep learning. Alongside using LSTM structures to analyze OpCode sequences. This method achieved an accuracy of 98.18 percent. Another study in [12] proposed a deep learning model for botnet identification based on Bidirectional Long Short Term Memory based Recurrent Neural Network(BLSTM-RNN) [13] and compared to LSTM is an RNN model. The author has generated a dataset for this work for including four attack vectors as used by the Mirai botnet. They have tested and validated their proposed method on four attack vectors: Mirai, UDP, DNS, and ACK with 99%, 98%, 98% validation accuracy.

III. DATASETS

This paper used a publicly available dataset achineLearningCSV, a piece of the CICIDS-2017 dataset from ISCX Consortium. It consists of eight real-world traffic monitoring sessions in a comma-separated value (CSV) file. This dataset was collected and distributed by researchers from the Canadian Institute of Cyber Security. It contains up-to-date attacks containing both benign and malicious network traffic traces. It is a labeled dataset including 84 features. Very last feature of the dataset is the class label, which classifies the sample as an attack or benign traffic. There are 14 kinds of attacks in this dataset. Some of the attack types are significant in aspect IoT detection. For example, a distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt a targeted server's, service's, or network's regular traffic by flooding the target or its surrounding infrastructure with Internet traffic. Port scan attack helps cybercriminals find open ports and determine whether they receive or send data. It can also reveal whether an organization uses active security devices like firewalls. Bot attack is when a website, application, API, or end-users are manipulated, defrauded, or disrupted via automated online requests. SQL Injection attacks (or SQLi) alter SQL queries with inputs, injecting malicious code by exploiting application vulnerabilities. Finally, a Web attack - brute force, is a method of guessing login information, encryption keys, or locating a hidden web page by using trial-and-error. We noticed that this dataset has a few weaknesses. For example, Because of the large amount of data, loading and processing take longer, missing class labels and few missing information, and the somewhat class imbalance. However, this dataset has several valuable characteristics; for example, the payload remains unchanged and not removed for privacy, including diverse attacks; it provides all the necessary protocols. In addition, it is labeled and instructive, allowing for accurate and dependable analysis.

TABLE I: DESCRIPTION OF CICIDS2017 DATASET

File Name	**Available Attack**
Monday-WorkingHours.pcap_ISCX.csv	Benign
3*Tuesday-WorkingHours.pcap_ISCX.csv	Benign
	FTP-Patator
	SSH-Patator
6*Wednesday-workingHours.pcap_ISCX.csv	Benign
	DoS GoldenEye
	DoS Hulk
	DoS Slowhttptest
	DoS slowloris
	Heartbleed
4*Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Benign
	Web Attack – Brute
	Web Attack – Sql In
	Web Attack – XSS
2*Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv	Benign
	Infiltration
2*Friday-WorkingHours-Morning.pcap_ISCX.csv	Benign
	Bot
2*Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Benign
	PortScan
2*Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Benign
	DDoS

IV. ALGORITHMS USED

Deep learning (DL) is a hierarchical representation of Machine Learning (ML) that is built on Artificial Neural Networks (ANNs). In the disciplines of image processing, pattern recognition, and computer vision, deep learning (DL) offers numerous improvements in traditional AI applications. Furthermore, deep networks can significantly improve classification and prediction accuracy in these complex tasks. We used four different deep learning models for classification and compared them with various machine learning algorithms in this paper.

A. Dense Network

A Dense network is a kind of convolutional neural network that uses Dense Blocks to connect all layers (with matching feature-map sizes) directly to each other, resulting in dense connections between layers. Each layer takes extra inputs from all preceding levels and passes on its feature maps to all following layers to maintain the feed-forward nature. Neural network-based Dense networks have become an increasingly popular solution for intrusion detection. The previous research using dense networks [14] achieved outstanding accuracy of 99.9% for Flooding detecting intrusion or attacks on IoT networks.

B. Convolutional Neural Network (CNN)

CNN architecture is a type of deep, feed-forward artificial neural network often used for image classification, segmentation, and also for different correlated data. Research [15] on the same CICIDS2017 dataset applied CNN alongside MLP for analysis. In our study, CNN requires the input shape of data in the 3d form, batch, steps, and channels. Therefore, we transformed the data to 3d shape accordingly.

C. CNN+LSTM model

This proposed model is implemented as a hybrid CNN with the LSTM model. The initial layer in this model is a 1-d CNN layer with ReLU activation, followed by an LSTM layer with Softmax Activation function [7]. For multiclass classification, the LSTM layer produces better accuracies with a sparse categorical entropy loss function. Since LSTM is an RNN variation, adaptive optimization like Adam performs well to adjust the learning rate. The remaining parameters are equivalent to those used in CNN and LSTM models.

V. Experiment

This section elaborates upon the details of the experimental setup and implementation of the research project to detect spam text.

A. Data Processing

The dataset is collected in different segments of time. But the features were always the same. We merged all the data available from multiple timelines. We found 1362 rows containing “NaN” values. This is less than one percent of the entire dataset. So we removed all the instances that included null and NaN values. For the training purpose, we split the dataset randomly as 80% training and 20% testing dataset. During the training period, we also introduced a random validation split of 10% of our training dataset. To feed the dataset into deep learning models, we encoded the categorical variables into integer values using a label encoder ranging from 0 to 14. The normalization was performed inside of the deep learning models. But, we standardized the dataset with scalar transformation before the training process started.

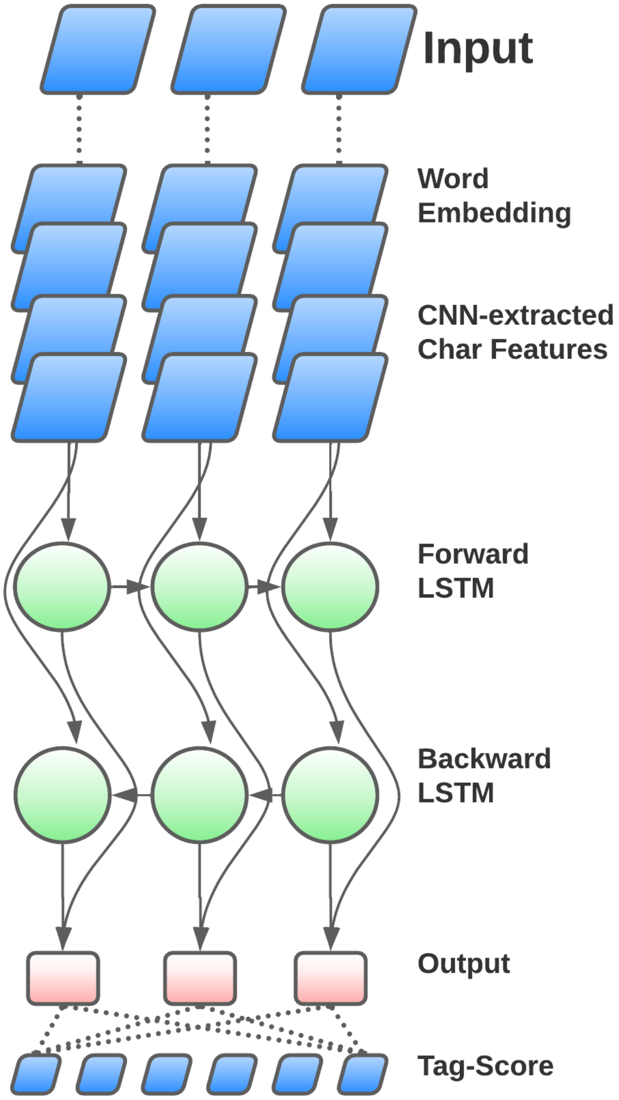

Fig. 2: Illustration of the CNN and LSTM architecture

B. DenseNet Implementation

DenseNet is divided into multiple dense blocks where we place a number of different filters. But the dimensions within each block are similar. The transition layer applies batch normalization using the downsampling technique. Out input shape was (78 x 1). We included three transition layers as 128, 64,32 sequentially. These transition layers were connected using an activation function, “Relu”. Right after the transition layers, we added a classification layer stating 15 outputs and added a SoftMax function as the final predictor. The dense blocks were (1 x 1 and 3 x 3). The second layer receives a total input as 79 parameters (78 predictors and 1 target). The third layer has (10112 = 78+1 x 128). The total parameters including the classification layer is 20,943 (0+10112+8256+2080+495). We used Adaptive Moment Optimizer (Adam) and Sparse categorical cross-entropy to calculate our loss. We choose Adam optimizer because, like other loss functions, it does not have a static learning rate. Adam adapts to the new parameter learning rate based on the calculation of the exponential moving average of gradient and the squared gradient. We used sparse categorical cross-entropy for our loss and validation accuracy metrics. We know that categorical cross-entropy performs better than other loss functions in the case of multiclass classification. We choose sparse categorical cross-entropy over the general cross-entropy since it also defines the mutual exclusiveness of your classes. To run the model, we predefined the batch size of 1024. And since we have a large amount of data, we set the epochs as 125 with the early stop parameter enabled, which eventually helps the model to stop training process if the consecutive accuracy does not improve.

C. *CNN Implementation*

To compare the effectiveness of our experimental flow, we have set up an experiment for Convolutional Neural Network (CNN). We tried to match the layer parameter as much as possible to compare the outcome of the architectural effectiveness. The CNN started with an input shape of (9 x 9 x 1). Since our dataset has 78 columns, we added 3 extra columns as padding containing 0. To come up with the same parameter size of 20,943, we added dense transition layers as (120 x 2), (60 x 3), and (30 x 4). The activation layers had 15 parameters similar to DenseNet. The transition layers used Ralu as an activation function, and the final classification layer used SoftMax. The loss and accuracy metrics were evaluated by sparse categorical cross-entropy. We used the adam optimizer as well. The batch size was 1024, and the epochs count was 125. We used the validation split as 90:10 for all of our experiments.

D. *CNN-LSTM Implementation*

Our primary target was to find a better architecture that would achieve the best performance for intrusion detection. So we developed a hybrid algorithm combining CNN and LSTM. We initiated the first single-dimensional convolution layer with (128 x 3) with 78 inputs. Again we used a similar size of the layer to blend the weights. Then we introduced a max-pooling layer of size 2. After the max-pooling layer, we again added the exact same single convolution layers twice. We added another max-pooling layer of size 2 again before we fed the params to the LSTM model. The LSTM layer started with 256 nodes. Then we introduced a dropout layer of 0.1 which reduced our unused node size. After the dropout, we simply included the classification layer for 15 output. To compare the architectural efficiency, we used the softmax function in the classification layer. All the other convolution layers used Relu as an activation function. We used sparse categorical cross-entropy for loss and accuracy function to match the comparison setup. Adam optimizer was used for adaptive learning rate based on the weight updates. We reduced the batch size to 512 and kept the same epochs count as 125 for CNN-LSTM.

VI. RESULT ANALYSIS AND COMPARISON

In this study, we tried to compare the neural network architectural effectiveness over the CIC-IDS2017 dataset. Our primary motivation is to deploy the machine learning model to IoT infrastructure. So instead of introducing more deep layers to our neural networks, we tried to find out suitable architecture which would perform better with less computation power. DenseNet achieved the highest accuracy of 97.77% on our test data. We primarily believe that without the dropout and convolution used in CNN and CNN-LSTM, our data remains intact and contribute efficiently to the classification layer [16]. Since our loss function also considers the mutual exclusiveness among 15 categorical inputs, it is always better to provide all the combinations of a feature vector to the classification layer. The CNN and CNN-LSTM did a very good job achieving 96.87% and 96.92%. The precision and recall also evident that all these three algorithms are perfectly capable of performing effectively on IoT infrastructure after deployment. The retraining process also seems very constant. The CNN really did great in optimizing the loss and keeping it stable over all the 125 epochs. In contrast, DenseNet and CNN-LSTM acted a little bit skewed in middle of the training, which can be considered as insignificant [17]. The model loss and accuracy loss for each of the deep learning algorithms are plotted and shown in Fig 5 and Fig 6. The performance metrics for each of the attack types are provided in Table II.

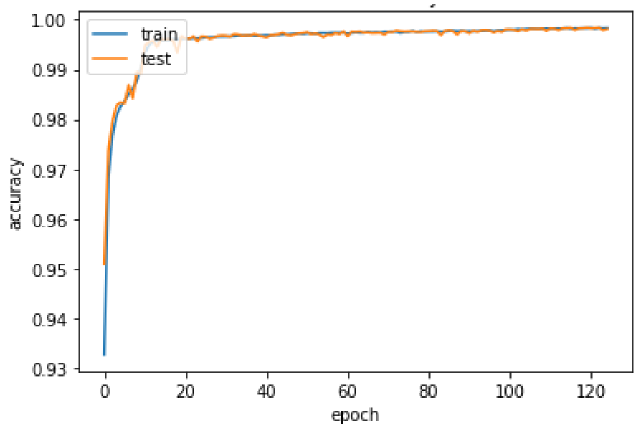

Fig. 3: CNN Accuricy.png

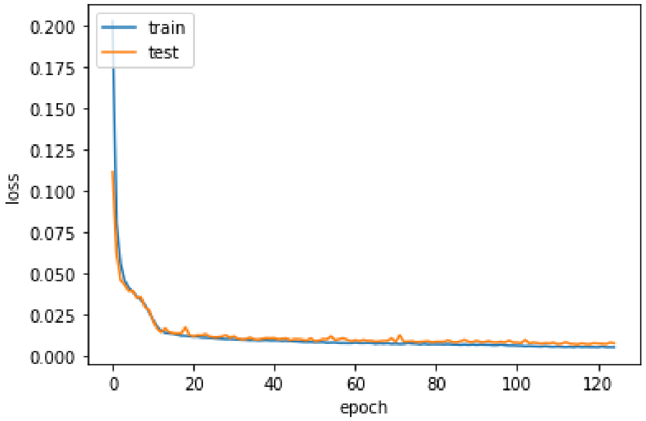

Fig. 4: CNN loss

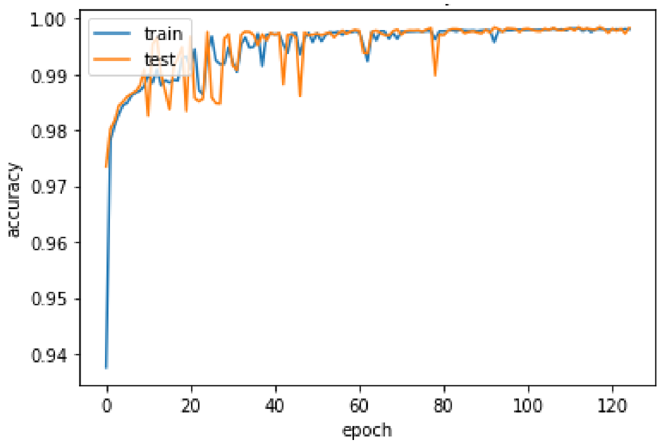

Fig. 5: CNN-LSTM Accuracy

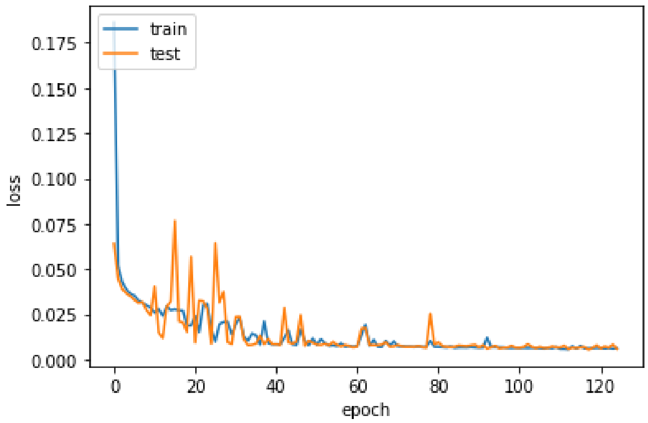

Fig. 6: CNN-LSTM loss

We can clearly observe that our models are very much efficient in detecting DDoS attacks and classifying benign traffic flows from the attack types.

VII. Conclusion

Modern IoT technologies and systems provide more appealing chances for integrating the digital world's technology, services, and management capabilities. Because of the integrated development of technology, people are becoming increasingly linked and intuitive. It is critical to ensure security in order to give them confidence and ease of usage.In this study, we have proposed a solution to intrusion detection in IoT network security by comparing various deep neural network architectures. Finally, DenseNet performs better than the rest of the deep learning models and machine learning algorithms with an accuracy of 97.16%. Also, we analyzed and showed all various attack types on IoT devices marking the noteworthy ones. We want to improve the present models' model prediction performance and test them against more routing attack types for future work. We're looking at adding more features to create a single model that can detect many forms of attacks for this purpose.

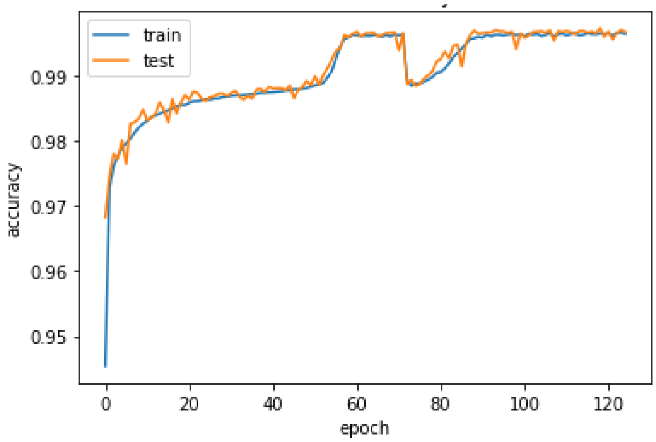

Fig. 7: DenseNet Accuracy

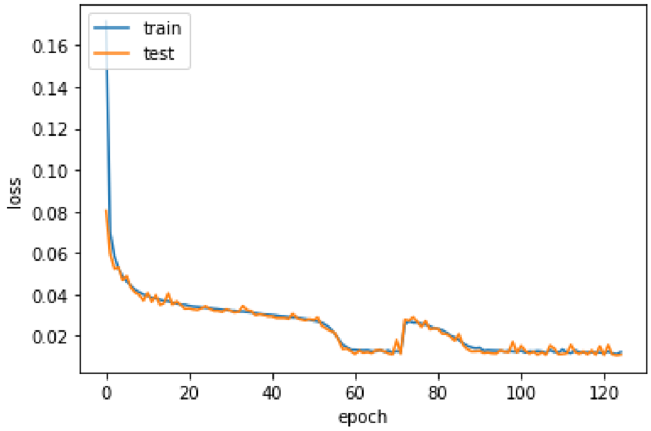

Fig. 8: DenseNet Loss

VIII. Acknowledgment

References

[1] Bhagya Nathali Silva, Murad Khan, and Kijun Han. Internet of things: A comprehensive review of enabling technologies, architecture, and challenges. *IETE Technical review*, 35(2):205–220, 2018.

[2] Rahul Gomes, Mostofa Ahsan, and Anne Denton. Random forest classifier in sdn framework for user-based indoor localization. In *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pages 0537–0542. IEEE, 2018.

[3] Sophia Moganedi and Jabu Mtsweni. Beyond the convenience of the internet of things: Security and privacy concerns. In *2017 IST-Africa Week Conference (IST-Africa)*, pages 1–10. IEEE, 2017.

[4] Kendall E Nygard, Aakanksha Rastogi, Mostofa Ahsan, and Rashmi Satyal. Dimensions of cybersecurity risk management. In *Advances in Cybersecurity Management*, pages 369–395. Springer, 2021.

[5] Nour Moustafa, Benjamin Turnbull, and Kim-Kwang Raymond Choo. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet of Things Journal*, 6(3):4815–4830, 2018.

[6] Rajendra Patil, Harsha Dudeja, and Chirag Modi. Designing an efficient security framework for detecting intrusions in virtual network of cloud computing. *Computers & Security*, 85:402–422, 2019.

[7] Mostofa Kamrul Ahsan. *Increasing the Predictive Potential of Machine Learning Models for Enhancing Cybersecurity*. PhD thesis, North Dakota State University, 2021.

[8] Mostofa Ahsan, Rahul Gomes, Md Chowdhury, Kendall E Nygard, et al. Enhancing machine learning prediction in cybersecurity using dynamic

TABLE II: PERFORMANCE COMPARISON OF RESULT.

	Attack Name	Precision	Recall	F1-Score	Support
15*DENSE Model	BENIGN	0.99	0.98	0.99	454406
	Bot	0.98	0.4	0.56	402
	DDoS	1	0.99	0.99	25802
	DoS GoldenEye	1	0.85	0.92	2009
	DoS Hulk	0.99	0.96	0.98	46081
	DoS Slowhttptest	0.97	0.93	0.95	1129
	DoS slowloris	0.98	0.87	0.92	1163
	FTP-Patato	1	0.52	0.68	1593
	Heartbleed	1	1	1	2
	Web Attack-Sql Injection	0.01	0.67	0.01	6
	PortScan	0.82	1	0.9	31646
	SSH-Patator	0.12	0	0	1190
	Web Attack-Brute Force	0.92	0.04	0.07	297
	Infiltration	0	0	0	6
	Web Attack-XSS	0.5	0.03	0.06	145
15*CNN Network	BENIGN	0.96	1	0.98	454406
	Bot	0.29	0.39	0.34	402
	DDoS	1	1	1	25802
	DoS GoldenEye	1	0.98	0.99	2009
	DoS Hulk	0.99	1	1	46081
	DoS Slowhttptest	0.98	0.99	0.98	1129
	DoS slowloris	0.99	0.99	0.99	1163
	FTP-Patato	1	1	1	1593
	Heartbleed	1	1	1	2
	Web Attack-Sql Injection	1	0.67	0.8	6
	PortScan	1	0.51	0.67	31646
	SSH-Patator	1	0.51	0.68	1190
	Web Attack-Brute Force	0.4	0.06	0.1	297
	Infiltration	0	0	0	6
	Web Attack-XSS	0.53	0.06	0.1	145
15*CNN+LSTM Network	BENIGN	0.96	1	0.98	54406
	Bot	0.8	0.41	0.54	402
	DDoS	1	1	1	25802
	DoS GoldenEye	0.99	0.99	0.99	2009
	DoS Hulk	0.99	1	1	46081
	DoS Slowhttptest	0.76	0.99	0.86	1129
	DoS slowloris	1	0.7	0.82	1163
	FTP-Patato	1	1	1	1593
	Heartbleed	1	1	1	2
	Web Attack-Sql Injection	0.2	0.67	0.31	6
	PortScan	1	0.51	0.68	31646
	SSH-Patator	1	0.51	0.68	1190
	Web Attack-Brute Force	0.65	0.88	0.75	297
	Infiltration	0	0	0	6
	Web Attack-XSS	0.25	0.02	0.04	145

feature selector. *Journal of Cybersecurity and Privacy*, 1(1):199–218, 2021.

[9] Arif Yulianto, Parman Sukarno, and Novian Anggis Suwastika. Improving adaboost-based intrusion detection system (ids) performance on cic ids 2017 dataset. In *Journal of Physics: Conference Series*, volume 1192, page 012018. IOP Publishing, 2019.

[10] Mostofa Ahsan, Rahul Gomes, and Anne Denton. Smote implementation on phishing data to enhance cybersecurity. In *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pages 0531–0536. IEEE, 2018.

[11] Hamed HaddadPajouh, Ali Dehghantanha, Raouf Khayami, and Kim-Kwang Raymond Choo. A deep recurrent neural network based approach for internet of things malware threat hunting. *Future Generation Computer Systems*, 85:88–96, 2018.

[12] Christopher D McDermott, Farzan Majdani, and Andrei V Petrovski. Botnet detection in the internet of things using deep learning approaches. In *2018 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2018.

[13] Mostofa Ahsan and Kendall E Nygard. Convolutional neural networks with lstm for intrusion detection. In *CATA*, volume 69, pages 69–79, 2020.

[14] Shahadate Rezvy, Yuan Luo, Miltos Petridis, Aboubaker Lasebae, and Tahmina Zebin. 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)*, pages 1–6. IEEE, 2019.

[15] Gavin Watson. A comparison of header and deep packet features when detecting network intrusions. Technical report, 2018.

[16] Mostofa Ahsan, Rahul Gomes, and Anne Denton. Application of a convolutional neural network using transfer learning for tuberculosis detection. In *2019 IEEE International Conference on Electro Information Technology (EIT)*, pages 427–433. IEEE, 2019.

[17] Nafiz Imtiaz Rifat. Feature engineering on the cybersecurity dataset for deployment on software defined network. 2020.