2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) | 978-1-6654-1262-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICECCME52200.2021.9590852

Intrusion Detection Using Deep Learning

Misbah Anwer
Dept. of computer science
National university of Computer & Emerging Sciences
Karachi, Pakistan
K200992@nu.edu.pk

Ghufran Ahmed
Dept. of computer science
National university of Computer & Emerging Sciences
Karachi, Pakistan
Ghufran.ahmed@nu.edu.pk

Adnan Akhunzada
Faculty of Comp & Informatics
University Malaysia Sabah
Kota Kinabalu 88400
Adnan.akhunzada@ums.edu.my

Shahbaz Siddiqui
Dept. of computer science
National university of Computer & Emerging Sciences
Karachi, Pakistan
Shahbaz.siddiqui@nu.edu.pk

***Abstract*—Advancement in network attacks requires strong and growing security mechanisms. Internet of Things (IoT) is the evolving technology connected billion of devices right now and it builds on a set of network devices therefore it is under serious threat concern. Identifying attacks is crucial and critical task. The authors propose a hybrid DL driven approach to detect the attacks, one is Cuda Deep Neural Network Long Short-Term Memory (CuDNNLSTM) and another is Long Short-Term Memory (LSTM) on kitsune dataset. There is alarming situation for protection of all the smart systems in terms of security. In this paper we implemented LSTM and cuDNN LSTM networks to identify attack. Results show that our technique cuDNNLSTM outperforms in comparison of deep learning technique LSTMs that shows 99.79% accuracy on 6GB dataset approx. (250lac) records.**

Keywords— Intrusion detection, LSTM, CuDNNLSTM

I. Introduction

Things are replacing the human in different tasks which is giving importance to the field of IoT. Previously machine learning was performing very well in intrusion detection but due to human intervention and manual feature extraction was not performing up to the mark. Some features fail to identify the data from the network's underlying precise patterns. Patterns that are more durable and reflect the true nature of network traffic. Deep learning (DL) has recently seen a lot of use in domains like image classification, object recognition, and natural language processing in huge data. LSTM networks can be used to learn features and patterns in network data in order to categories it as benign or malicious. It's also clear that the LSTM network is resistant to attacks because attackers can't alter feature learning algorithms to improve their breaching approaches.

Technology increases day by day, our communication and networking system get more advance and the production of data also increase every day into the internet [1]. This huge amount of data is known as big data. The large account of data mostly generated by social media, healthcare and industry. The main characteristics of big data are large volume, large velocity, large variety and large veracity. The wireless devices also increase day by day such as mobile phone and also IoT devices that generate the data in very huge amount [2]. That's mean the flow of information in cyberspace are growing frequently and reached to the alarming stage. Due to this increase in data, several issues are also have increased like Network traffic, Data storage capacity issue, Security issue and so on. Attack rate also increases day by day into our network due to large amount of data. One of the biggest issues is intrusion attack in network. Currently this is the hot topic in research and development. Many people gave the different solutions for intrusion detection and propose many different techniques to detect the intrusion by using the intrusion detection system, Machine Learning (ML), Artificial Neural Networks (ANN), Support Vector Machine (SVM) and many other algorithms use in intrusion detection systems. Some other combined algorithms or hybrid approaches were also used to detect the intrusion like Multi-scale Convolutional Neural Network with Long Short-Term Memory (MSCNN-LSTM) [3]. The intrusion detection system is the front line or first level of detection system for the intrusion attacks then its further divide into categories like Network base IDS, Host base IDs, mostly in host computers. HB-IDS for anomaly base detection and also for signature base recognition. Now a days Machine learning algorithms and deep learning algorithms are more power full and mostly use for the intrusion detection, data size is bigger in this detection, deep learning is more powerful because its work on large data set and as compare to the machine learning algorithm the computation power of deep learning algorithm is high. Some NN algorithm like DBN, CNN, LSTM WDLSTM works on big dataset and their performance is fast due to using of minimum computation power [4]. Feature detection is not an easy task especially in large data set that contain more features. Detecting the meaning full feature for CNN model is very important. In [4] authors used CNN to detect the useful feature for large data set. They also used LTMS for long term dependencies among selected features. Other algorithms can also be used to detect the intrusion. Many efforts have done in this domain using different strategy and different way to detect the intrusion [4].

II. LITERATURE REVIEW

IoT, IIoT and IoMT generating data exponentially and data generation process are increases day by day and they are large in volume that's why they known as big data and the number of wireless devises is increase day by day like IoT devices or other

978-1-6654-1262-9/21/$31.00 ©2021 IEEE

3G,4G devices that transmit data into the networks and data load were increase day by day. In big data network intrusions detection are more difficult and the attack can easily perform and it is crucial and critical to detect the network intrusion. In [5] many solutions discussed for intrusion detection.

Neural network (NN) solves many difficult problem or complex problem easily in less interval of time using the Convolutional Neural Network (CNN), extract the meaning full feature form the network and it require less preprocessing than other classification algorithms. In [6] Benchmark dataset for this model contain '2540044' rows or entries and '49' features to train and test this model. This dataset contains multi classes and the percentage of normal class is grater then the other classes. In [10] they used regularization technique to avoid the over fitting and biasness into the model and to reduce the skewness in the data. This model contains five layers and the functionality of activation means "Relu" (x) = max $(0, x)$" and the other layer which is also called output layer use "SoftMax" to estimate classes. Fifty epochs were used for training, the model has 32 and 65 number of neurons in first two layers.

NN has changed the work flow and boost up the AI workflow special in the field of CNN, NLP, and Computer vision. For detection of intrusion, they proposed hybrid model with the combination of Multi Class Convolutional Neural Network to analyze the special feature in data set and Long Shot Term Memory use for the to process the temporal features. In this model they have used UNSWNB15 dataset for training and testing and the accuracy of this model is better for the classification. This dataset contains '9' type of classification and the '49' feature for each row that contain only '1' label. The CNN model uses back propagation algorithm with G.D method and this model is '3' different size of kernel 1x1, 2x2 and S3x3 there are '175341' data point use in training and '82332' data point use in validation and '5576' data point use in testing. Accuracy of this mode is 95.6% and other model like Lenet-5 accuracy is 63,9% and HAST accuracy is 85.7% that's clearly shows that this model performs well in large amount of data and give the more accuracy than other models. [2,6]. Table 1 show the literature and previous efforts done by others.

TABLE I. SUMMARY OF RELATED WORK

Research Works	Approach	Contribution
Sowah.et al. [3]	ANN	Detect MITM attack in MANETs and MANETs with internet
Diro.et al. [4]	LSTM	Identifying distributed cyber-attacks using LSTM in fog to things communication
Hassan.et al. [6]	CNN and WDLSTM	Efficiently detect attacks using hybrid deep learning approaches
Yahalom et al. [7]	FFDNN, WFEU	High accuracy for Efficiently detect attacks using hybrid deep learning approaches
Rahman et al [8]	SDN	Distributed Block building foe secure data transfer
Tooba.et al. [9]	LSTM	SDN-enabled control plane based orchestration

III. METHODOLOGY

To identify attacks, we setup our experimental environment to set all traffic from different devices then we apply LSTM and cuDNN LSTM on ARP MITM kitsune dataset. Dataset has been divided into 70%:30% and 80%:20% for training and testing respectively.

A. DataSet

The state-of-the-art dataset named *Kitsune*, Kitsune is a cyber-security dataset which works for nine different network attacks, especially on an IOT network. It can detect Denial of service (DoS), "Man in the Middle attack" (MitM) and bot type attacks. These three are very common attacks in computer network system. By applying machine learning algorithms, we can build models which are predictive and those models can easily detect network attacks. If we talk about man in the middle attack as per our analysis detection of each network attack depends on the detailed study of machine learning algorithms. We selected the chunk for MitM and ARP (Address Resolution Protocol). Basically, through an ARP attacker behind a fake IP address can be detected easily. The chunk we are working is divided into three parts. We have done network intrusion detection through these chunks. ARP MitM_pcap. pcapng: it works on pcap and pcap files. There is a raw pcap which have N number of original packets. The packets are truncated to some bytes due to privacy reasons. another is ARP MitM_dataset.csv: basically, a CSV file which describes size of packets as well as size of matrix and last isARP MitM_labels: it is a csv file which indicates about the packets if they are malicious or not. It has value of '0-1' Where '0' means not malicious and '1' means malicious. We have created a model of network intrusion detection on Kitsune.

B. LSTM

In light of the intermittent engineering in the organization, the purpose of RNN is to manage time grouping information. However, RNNs have trouble learning to link information as the amount of information required develops if the distance between the unit containing the relevant data and the unit where it is stored is too great. Long Short-Term Memory (gate and output gate) (LSTM) Networks were developed as a result of the implementation of a three-gate design (input gate, forget gate, and output gate). [8]. LSTM is actually an ANN (recurrent neural network) architecture used in deep learning unlike in contrast to standard neural organizations, LSTM has an input association. It can deal with singular information focuses (like pictures), yet additionally whole information sequences (like video or speech). Commonly the LSTM unit consists of some "forgotten gates", "input gates", "output gates", and "cells". Cells memorize values randomly, and three gates organize the data flow towards and from the cell. The LSTM network is perfect for classifying processes, and forecasting which depends time series information, as there may be delays in the amount of time among some essential events in a series of times. LSTM has been designed to address the problem of disappearing gradients that can be solved by studying traditional RNN.

C. *cuDNNLSTM*

NVIDIA CUDA is a library presented by Deep (Profound) NN by “cuDNN”. cuDNN is a GPU-accelerated library of primitives for deep neural networks. cuDNN allow most versatile performance for worthy setup in DNN like pooling, activation layers, forward and backward convolution and normalization. For high-performance GPU acceleration profound learning specialists and Framework designers all over the world depends on cuDNN. cuDNN permit them to focus on developing software applications and training neural networks. cuDNN always broadly used deep (profound) learning systems (framework) such as “KERAS”, “MATLAB”, “TENSORFLOW” etc. There are multiple types of recurrent layers; these include Long short-term memory (LSTM) and CuDNNLSTM. ***CuDNNLSTM*** is a Fast LSTM implementation backed by CuDNN. It Can only be run on GPU, with the TensorFlow back-end. CuDNNLSTM is designed for CUDA parallel processing and cannot run without GPU.

IV. IMPLEMENTATION DETAILS

Since we are working on LSTM using cuDNN 5, The equations which easily through with forward propagation of information can be done using LSTM.

$$it= \sigma(Wixt + Riht-1 +bi)$$
$$ft= \sigma(Wfxt + Rfht-1 +bf)$$
$$Ot= \sigma(W0xt + R0ht-1 +b0)$$
$$C't= \tanh(Wcxt + Rcht-1 +bc)$$
$$ct= ftoct-1 +itoc't$$
$$ht=oto \tanh (ct)$$

Along with LSTM, we have also used Convolution networks, which equations are:

$$Yn,k,p,q = \sum Cc \sum Rr \sum SsXn,c,p+r,q \times Wk,c,r,s + s$$

All the work has been done on TensorFlow GPU environment on anaconda navigator (Jupiter Lab). Dropping some rows to get accurate and equal shapes. We divided our dataset into test size and train size. First of all, we decided 70% train size from the chunk of dataset (on which we applied ARP) and the test size was of 30%. Then for model creation we have applied 2D convolution along with Long Short-Term Memory (LSTM). After creation of model, we have trained our dataset over that model. In this training phase we have converted our X and Y (test, train) chunks into an array so conversion from letters to tensor could be possible. In training phase, we have achieved an accuracy of ‘99.76%’ on 50 epochs, our model on training dataset and then again, we achieved accuracy of ‘99.761%’. And a minimal loss of ‘0.01’.

Accuracy is a measurement metric which evaluate classification model. Accuracy is basically a fraction of predictions that we are going right or our model is right. As can see in a confusion metrics the accuracy is ‘0.991249’ it means that our model predicts values which are ‘99%’ correct. In the confusion matrix there is another value just below the accuracy. Recall means the number of positive class predictions of all positive examples in the dataset. So, our recall value is ‘0.9912’ which is as same as accuracy. It means ‘99%’ true positives have been recalled in our dataset. In the confusion matrix the value of precision is mentioned. It means if formula is incorrect the following number of returned hits were true positive. Now let come towards the f1score. F1score is the weighted average of precision and recall. In our dataset f1score containing the following (given in confusion matrix) numbers of false positive and false negative. When our test size was ‘20%’ f1score was ‘99.4%’ accurate but when we maximize our test size to ‘30%’ it didn’t give the accurate result. F1 score is more than accuracy useful in uneven class distribution.

V. RESULTS AND DISCUSSION

Confusion matrix is shown in Table 2. Basically, confusion table is a classification table which use to describe the performance of a model. The confusion matrix for above training set is given in Table 2.

TABLE II. SHOWS THE CONFUSION MATRIX

Confusion Matrix	
Name	***Values***
Accuracy	0.997949
Recall	0.997249
Precision	0.997568
F1 Score	0.997324

- TPR, true positive rate used to measure the percentage of actual positives.
- TNR, true negative rate is the proportion of samples that test negative or the model that correctly predict negative class.
- PPV, positive predictive value is the small part of pertinent occurrences among the recovered occasions.
- FPV, false positive rate when model predict false positive class.
- FNV, false negative rate is an output when model predict false negative class.
- And ACC is overall accuracy of the dataset. (trained)

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

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

In the model creation phase, we have used 2D Convolutional along with LSTM (Long Short-Term Memory) so that we can construct the output of system for any arbitrary input signal over the 2D input data.

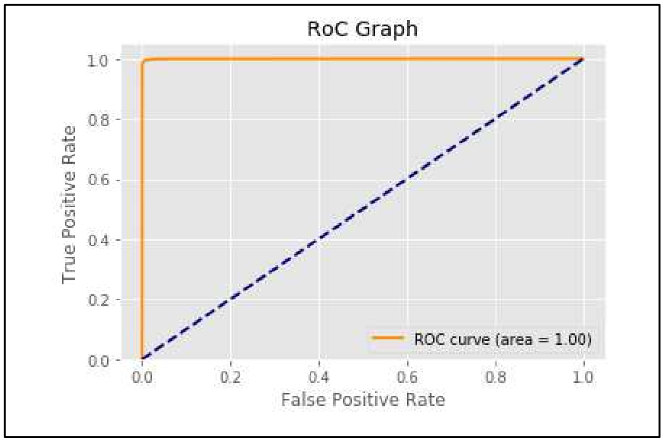

Fig. 1. True and false positive rate of the model.

To show the performance of classification model at all thresholds use ROC, as mentioned in Fig.1, Receiver Operating Curve Graph. True positive and true negative are the two parameters of this curve. Fig. 1. show the ROC curve, ROC rates of true positives and true negatives. It is the rule of ROOC that AUC value 1 denotes that our ROC is good and correct and classifier is excellent. The accuracy of true positive rate is 0.9942 and 0.9906. It means up to this percentage our model can correctly classify the actual positives which are identified correctly. Same as for true negative, the rate of true negative here which means that a program has been set on test data where has successfully predicted the negative outcomes.

The effect of number of epochs shown to our dataset accuracy. Epoch basically is a term we use in Machine Learning which indicates the number of the whole train dataset where the ML algorithm has been completed. On first epoch our train and test accuracy were ‘99.465’ in ‘22’minutes and on ‘40’ minutes train accuracy became ‘0.926%’ and test accuracy became 99.06% and more witness from the table that our test and train size accuracy percentage is directly proportional to number of epochs. When we applied ‘50’ number of epochs to our dataset we got perfect accuracy on our train and test size ‘99.99%’ in ‘71’ minutes

TABLE III. TRAIN AND TEST ACCURACY WITH LOSS

Data Size	**Epoch**	**Batch**	**Layers**	**Loss**	**Train Accuracy**	**Test Accuracy**	**Time (Minutes)**
2500k	1	32	3	0.0153	0.9946	99.461 %	22
2500k	1	64	4	0.0091	0.9956	99.46 %	40
2500k	10	500	1	0.02502	0.9969	99.62 %	22
2500k	30	1000	2	0.0221	0.9971	99.69 %	45
2500k	50	1000	3	0.0183	0.9979	99.80 %	71

A. Evaluation Metrics

To calculate actual and predicted values Fig. 4 shows the model accuracy and loss. For comparison with different dataset, we compare with KDD dataset and the accuracy is gradually increasing and then again decreasing while the ARP MITM shows the constant accuracy and loss throughout the epochs as mentioned in Fig. 2. and Fig.3. respectively. Model was trained and test on different number of nodes and hidden layers with Relu, and SoftMax activation function. Similarly, Fig. 3. shows the model loss when compared with KDD dataset that gradually decreasing the loss.

B. Training on Bigger Dataset

Dataset again divided into 20% test size and 80% test size dataset. Table 3 shows the accuracy on 80% training dataset on all records.

Training and testing on all 2500k records show the accuracy of 99.79 with approximate 99.80 testing and 0.005 loss. In this training phase we have converted our X and Y (test, train) chunks into an array so conversion from letters to tensor could be possible. In training phase, we have achieved an accuracy of ‘99.76%’ on 50 epochs, our model on training dataset and then again, we achieved accuracy of ‘99.761%’.

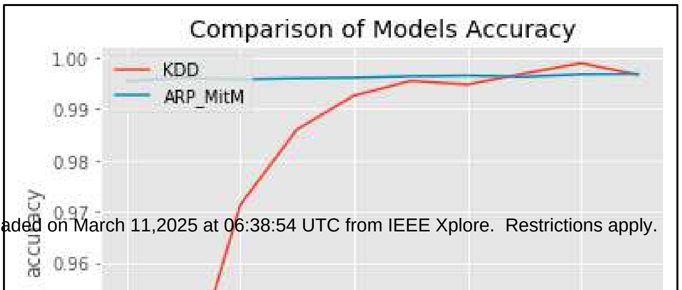

Fig. 2. Comparison of two models Accuracy

Similarly, Fig. 3. shows the model loss when compared with KDD dataset that gradually decreasing the loss.

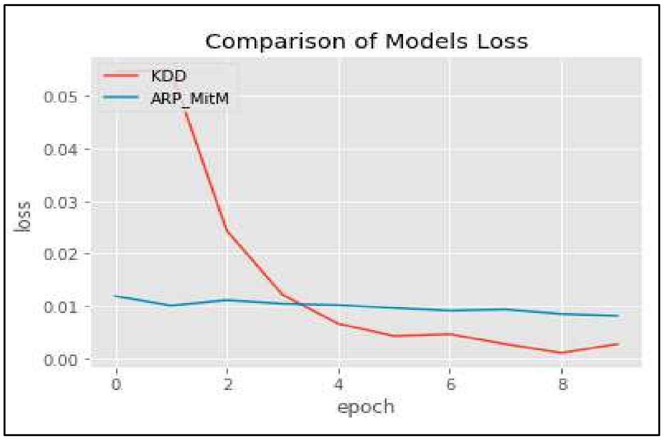

Fig. 3. Comparison of two models' loss

A minimal loss of '0.01'. On first epoch our train and test accuracy were '99.465' in '22'minutes and on '40' minutes train accuracy became '0.926%' and test accuracy became 99.06% and more witness from the table that our test and train size accuracy percentage is directly proportional to number of epochs. When applied '50' number of epochs to our dataset and got perfect accuracy on our train and test size '99.99%' in '71' minutes.

CONCLUSION

LSTM was used here to train the model. CuDNNLSTM and long short-term (LSTM) memory is basically an extension of RNN. LSTM can remember data for longer periods. It has a complex architecture which consist of '4' hidden layers. In our dataset the core feature of LSTM is the cell state. Means to add or remove information from the cell state. To improve quality of raw data processing and filtering is required which increased data efficiency for our dataset. LSTM network classified, processed and made prediction in our dataset on time series data. It also has ability to eliminate information from and to the cell state. The output of our model gave binary results 0 for non-malicious and 1 for malicious data. Time required for training or testing data is calculated as evolution time of model. LSTM in intrusion detection network worked as anomaly detection system in our model. Different parameters of classification have been calculated through LSTM here which has mentioned above in confusion matrix. proportional to number of epochs. When we applied '50' number of epochs to our dataset we got perfect accuracy on our train and test size '99.99%' in '71' minutes.

REFERENCES

[1] M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino. "A hybrid deep learning model for efficient intrusion detection in big data environment." Information Sciences 513 (2020): 386-396.

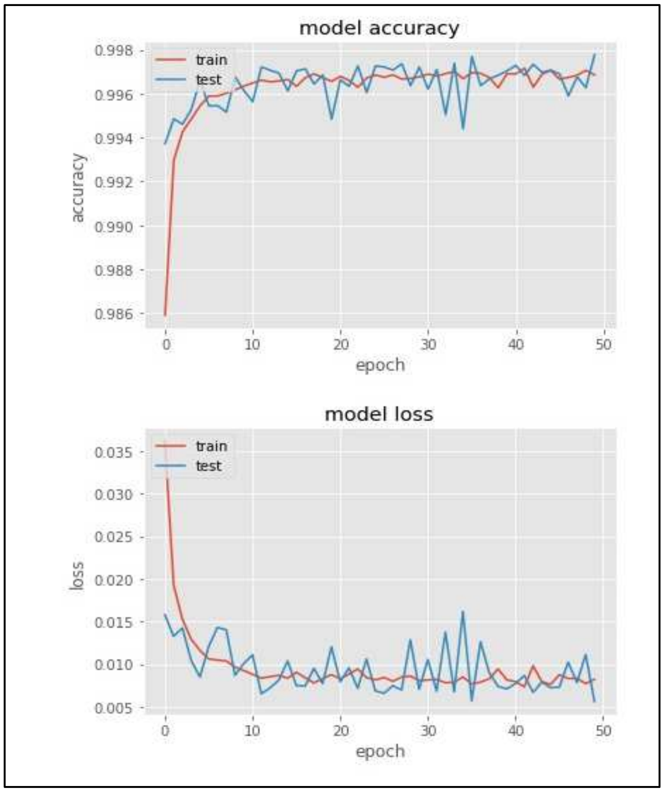

Fig. 4. Shows the model accuracy and loss.

[2] S. M. Kasongo and Y. Sun. "A deep learning method with wrapper based feature extraction for wireless intrusion detection system." Computers & Security 92 (2020): 101752.

[3] J. Zhang, Y. Ling, X. Fu, X. Yang, G. Xiong, and R. Zhang. "Model of the intrusion detection system based on the integration of spatial-temporal features." Computers & Security 89 (2020): 101681.

[4] R. A. Sowah, K. B. Ofori-Amanfo, G. A. Mills, and K. M. Koumadi. "Detection and prevention of man-in-the-middle spoofing attacks in MANETs using predictive techniques in Artificial Neural Networks (ANN)." Journal of Computer Networks and Communications 2019 (2019).

[5] A. Diro and N. Chilamkurti. "Leveraging LSTM networks for attack detection in fog-to-things communications." IEEE Communications Magazine 56, no. 9 (2018): 124-130.

[6] A. Sebbar, K. Zkik, Y. Baddi, M. Boulmalf, and M. D. E. Kettani. "MitM detection and defense mechanism CBNA-RF based on machine learning for large-scale SDN context." Journal of Ambient Intelligence and Humanized Computing 11 (2020): 5875-5894.

[7] R. Yahalom, A. Steren, Y. Nameri, M. Roytman, A. Porgador, and Y. Elovici. "Improving the effectiveness of intrusion detection systems for hierarchical data." Knowledge-Based Systems 168 (2019): 59-69.

[8] H. Xue, D. Q. Huynh, and M. Reynolds. "SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction." In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1186-1194. IEEE, 2018.

[9] A. Rahman, M. K. Nasir, Z. Rahman, A. Mosavi, S. Shahab, and B. M. Bidgoli. "Distblockbuilding: A distributed blockchain-based sdn-iot network for smart building management." IEEE Access 8 (2020): 140008-140018.

[10] C. Yin, Y. Zhu, J. Fei, and H. Xinzheng. "A deep learning approach for intrusion detection using recurrent neural networks." Ieee Access 5 (2017): 21954-21961.