Garima jain,2020, 8:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Protected Deep Neural Network for 5G Wireless Systems

M.Tech.Scholar Garima Jain, Dr. Sudhir Agrawal (Dean Academics),Prof. Ankit Shrivastava
Dept. of Electronics and Communication Engineering
SAGE University, Indore,MP,India
garimajain2812@gmail.com, dean.academics@sageuniversity.in, ankit.shrivastava18@gmail.com

***Abstract*- The current generation network is being rapidly replaced by programmable 5G network. 5G wireless network is also expected to be core network for various industrial use cases. Any kind of security breach will lead to disruption of service to the users, increase in expenditure to the service providers and many other unfavorable effects. So, to prevent the disruption of networks, we have proposed a secure deep learning-based framework that is intelligent to detect attacks and prevent propagation of attack. The framework was tested on emulated wireless network and the result showed an accuracy of 97.8% which proved the efficiency of the framework.**

***Keywords:*- 5G, Software Defined Network, Wireless Network, Machine Learning, Deep Learning, Recurrent Neural Network.**

I. INTRODUCTION

Our era of communication began centuries back with signal flags, smoke used to send signals and other innovative wireless signaling techniques. Wireless communication started to play a major part in our life with Guglielmo Marconi's invention and other breaking inventions. Right then our humankind has witnessed several generations of communication network that is constantly getting fine tuned based on human expectation.

At present the wireless field is the backbone of various industries starting from activating simple home appliances from remote location to satellites very far from the Earth atmosphere communicating with scientists on Earth [1]. In particular the next generation programmable 5G wireless network promises to realize the dreams of the crucial industry verticals like crucial factory works, human health, driver-less automobiles etc. through its programmable architecture [2]. Our world's major daily works revolves around the wireless network. Any form of simple security breach in 5G network will lead to adverse effects to both the users and the service providers for instance, interception of the payment message and changing the account number will lead to huge loss for the users and also the bank service provider losing the business [3]. Not only in terms of expenditure, in some cases it could be a danger to several lives (for instance, a modification attack on the robotic surgery system will lead to adverse effect in fraction of seconds) [4]. It is necessary to secure the whole programmable 5G wireless architecture [5].

The programmable trait is to be provided by various novel networking paradigms like Software Defined Network (SDN), Network Function Virtualization (NFV), edge computing etc [6] [7]. Of these various technologies, SDN strives to separate the data plane and the control plane entirely [8]. The control plane sets the rules instructing what the nodes in the data plane has to do. This enables new modules to be quickly latched onto the 5G network. And also as the data plane only does forwarding, delay in their job is reduced considerably [9]. And also we are witnessing novelty even in attack vectors which can not be detected by traditional detection system [10].

It is necessary for an intelligent system to provide efficient detection of the security breach. So in this paper we have proposed a novel secure framework for SDN based 5G wireless network that trains the system with deep learning algorithm to detect malicious traffic and mitigate its propagation. Deep

International Journal of Science, Engineering and Technology

An Open Access Journal

learning is preferred to the single layer learning as various layers of reasoning help to identify intricate changes in the attack vector.

II. RELATED WORK

There are various research works done in the field of 5G wireless network security. Some of the works are listed in this section.

Cox et al. [11] have provided a network flow guard methodology to detect rogue access points in the wireless networks by leveraging SDN over wireless network. The authors have proposed the usage of various metrics in detecting the rogue access point like time to live check, multiple MAC-IP associated to each port etc. In case of rogue access point detection the traffic is directed to a trusted agent or flow rules are set to entirely drop the packets across the rogue device.

Liyanage et al. [12] have proposed architecture to secure the 5G software defined mobile networks to handle security breaches occurring across various functioning levels. The proposed architecture comprises of five components namely secure communication based on establishment of secure host identity protocol tunnels across the northbound interface communication channels; policy based communication that involves setting policies based on two levels of operation namely customer edge switching based edge to edge trust and attack evidence collection; security information and event management component gathers necessary flow statistics to update the access control list, make changes to the flow table, set rules for firewall etc.; deep packet inspection to monitor malicious traffic; security management and monitoring component does load balancing, traffic mirroring, collect malicious network behavior.

Varadharajan et al. [13] have proposed a secure framework for monitoring of patients who tend to wander around the hospital environment. The framework utilizes the SDN and wireless network to install policies on the wearable sensors of the patients. The policies dictate the locations the patient should not enter and in case of violation an alarm is generated to the staff.

Sharma et al. [14] have proposed OpCloudSec architecture that uses Deep Belief Network (DBN) to detect attacks and the SDN framework is used to set policies for the packets reaching the cloud. The module verifies if it is new packet it is sent to the attack detection module to classify if it is malicious or not. If it is not new the switch does the action set in the flow rule.

Dai et al. [15] have proposed a TNGuard that secures the tenant networks with the aid of SDN running in its core. There are various zones in the TNGaurd. Of the zones, the control zone comprises of controller module to control the tenant networks, cloud and a hypervisor that is used to manage the privileges set for the network. Wireless network's security is not robust enough due to lack of necessary infrastructure. With today's improved computational power and availability of a cluster at less cost, it is possible for a breaker to break the security protocols of the wireless networks.

Thus there is a requirement for an intelligent algorithm to detect the attack. Existing solution utilize machine learning algorithms such as DBN. But this DBN is not efficient when input is not certain [16]. Wireless network that has less infrastructure and with mobile devices in the network, we cannot expect certain inputs all the time. Motivated by these facts we pondered over the security breach in wireless networks and evolved a novel framework named PrDeN that utilizes deep machine learning algorithm to detect the attack in wireless network. Our Protected Deep Neural (PrDeN) framework provides the following key contributions in securing the 5G wireless network:

- Our PrDeN does not make any modifications to the core working of the SDN network model. It is an independent add on module that does not depend on any of the other modules.
- Our PrDeN does not overload the north bound interface bandwidth for detection of the malicious traffic through the introduction of time frame component.
- Dynamic nature of the PrDeN framework in collection of the traffic improves the network performance by avoiding unnecessary traffic collection.
- Learning model incorporated on to the PrDeN aids in detecting attack based on previous observed traffic thereby improving the efficiency of the framework.

International Journal of Science, Engineering and Technology

An Open Access Journal

III. PROTECTED DEEP NEURAL(PrDeN)

Our PrDeN framework module runs on the SDN controller. In general the controller should be a high performance computing system in order to have a smooth working of the networks. So the PrDeN is placed in the controller. In order to avoid the break down of the network due to failure of the central controller, we propose the usage of decentralized controllers i.e. there is an alternative controller which is present to take over in case of any failure. PrDeN comprises of three major components namely monitoring, attack detection and attack mitigation.

1. Monitoring : Even before the deployment of the module onto the live network, the controller should be trained with attack and normal traffic. Here we have utilized the LSTM RNN to train the system. A common attack strategy to bring about a full fledged disruption of the network the attacker has to send a series of attack packets or use any other series of strategy (for instance in low rate Denial of Service attack, the packets are not flooded at once, slowly attack packets are sent to the server to consume their constrained resources which is very difficult to identify). So to identify such types of attacks, the controller has to keep track of the previous instances of the network traffic. So, we propose the usage of LSTM RNN out of the various machine learning algorithms. As LSTM RNN's inherent characteristic is to keep the previous instances of the incoming data, it will suit the attack identification by observing from the start of the attack packet propagation. Algorithm 1 illustrates the pre training process to get the controller module ready to combat the attack on the 5G wireless network.

Once the trained system is tested with test dataset it is deployed onto the live 5G network, the monitoring work begins. A predetermined time frame is set. The time frame can be varied across the time of the day depending on the traffic intensity, critical traffic propagation time and network usage statistics (for instance at night time the hits to the server or any other traffic will be relatively very less compared to the day time so the attack probability is also less as it does not disrupt the network availability to the users). At the end of the time frame, the traffic received across the access point is sent to the controller. Algorithm 2 illustrates the monitoring phase of the PrDeN framework.

2. Attack detection : The RNN trained controller module keeps the previous observed traffic instances. Based on the previous traffic instance and the current traffic instance, the RNN controller module predicts if it is attack or normal traffic. Algorithm 3 illustrates the attack detection module of the PrDeN framework. In case, the detection module detects the traffic to be normal no further action is taken the module goes back to listen state waiting for the end of next time frame. In case the detection module detects the traffic to be malicious, the necessary traffic statistics are extracted from the attack traffic and is sent to the mitigation module. The extracted features include the packet MAC address, IP address, source port and destination port.

Algorithm 1: Training of controller

input : Training dataset (D_{Tr})
output : Trained controller

Training dataset $\leftarrow (\vec{T}, \vec{class})$ /* T is the vector of network traffic; class is the vector of classification of traffic if it is normal or attack; Training dataset is sequential traffic observed */
$\eta \leftarrow$ Learning rate /* A small value is initialized as learning rate */
$N \leftarrow \#(D_{Tr})$
$I_F \leftarrow$ #(Input features of D_{Tr})
$P \leftarrow$ #(Epoch)
$L \leftarrow$ #(LSTM RNN units)
$F \leftarrow$ Forget gate in LSTM unit
$I \leftarrow$ Input gate in LSTM unit
$C_{up} \leftarrow$ Modulation for cells in LSTM unit
$C \leftarrow$ Cells in LSTM unit
$O_{up} \leftarrow$ Modulation for output gate in LSTM unit
$O \leftarrow$ Output gate in LSTM unit
$W_{ij} \leftarrow$ Weight matrix
$i, j \in F, I, C, O$
$b_m \leftarrow$ bias of gate in LSTM unit
$m \in F, I, C, O$
$a_l \leftarrow$ activation output of each LSTM unit
$l \in 1...L$
$E_n \leftarrow$ error observed per training data instance
$n \in 1...N$
$err \leftarrow$ Error aggregated at the end of running an epoch
For K = 1 to P
$E_n \leftarrow 0$
$err \leftarrow 0$
foreach $Z \in (\vec{T}, \vec{class})$ **do**
  For Q = 1 to L
    $a_0 \leftarrow 0$ /* Initialize activation vector to be 0 */
    $C(0) \leftarrow$ Initialize to random small values
    $C_{up}(Q) \leftarrow \frac{e^{((W_{CZ}*a_{(Q-1)})+(W_{CZ}*Z)+b_C)} - e^{-((W_{CZ}*a_{(Q-1)})+(W_{CZ}*Z)+b_C)}}{e^{((W_{CZ}*a_{(Q-1)})+(W_{CZ}*Z)+b_C)} + e^{-((W_{CZ}*a_{(Q-1)})+(W_{CZ}*Z)+b_C)}}$
    $I(Q) \leftarrow \frac{1}{1+e^{-((W_{IZ}*a_{(Q-1)})+(W_{IZ}*Z)+b_I)}}$
    $F(Q) \leftarrow \frac{1}{1+e^{-((W_{FZ}*a_{(Q-1)})+(W_{FZ}*Z)+b_F)}}$
    $C(Q) \leftarrow (I(Q) * C_{up}(Q)) + (F(Q) * C(Q-1))$
    $O_{up}(Q) \leftarrow \frac{1}{1+e^{-((W_{OZ}*a_{(Q-1)})+(W_{OZ}*Z)+b_O)}}$
    $O(Q) \leftarrow O_{up}(Q) * \frac{e^{C(Q)} - e^{-C(Q)}}{e^{C(Q)} + e^{-C(Q)}}$
    $a_Q \leftarrow O(Q)$
  end
  $E_Z \leftarrow O(L) * (1 - O(L)) * (actual - O(L))$ /* actual represents the actual output class in training data */
  $err \leftarrow err + E_Z$
end
$W \leftarrow W - (\eta * err * input)$/* Weight is updated based on the error observed at the end of an epoch; W represents the weight matrix across various gates in the LSTM unit; input represents the input across various gates in the LSTM unit */
end

International Journal of Science, Engineering and Technology

An Open Access Journal

3. Attack mitigation : Algorithm 4 illustrates the attack mitigation phase of the PrDeN framework. The extracted features is given as input to the mitigation module. In majority of the cases the attack is launched by spoofing the IP address. So the mitigation module utilizes the MAC address to eliminate the malicious device from the network. The mitigation module sets flow rules on the openflow switch connected to the access point to drop all the packets emanating from the MAC address. The remaining features like the port details can be used to analyze the target of the attackers. Figure 1 shows the flow of traffic across the modules, when the attacker attacks and the PrDeN mitigating it.

Algorithm 2: Monitoring

```
input  : Traffic (T)
output : Previous Traffic (P_T), Current Traffic (C_T)
F_n ← Dynamic time frame
n ∈ 1...24
S ← Current system time
H ← Current hour
while (true) do
    Get S
    Extract H from S
    n ← H
    if (end of F_n)
        Pull T from access point
        C_T ← T
        Attack_detection(P_T, C_T)
    else
        wait
    Remove 1st traffic matrix from P_T
    P_T ← P_T ∪ C_T
end
```

Algorithm 3: Attack detection

```
input  : Previous Traffic (P_T), Current Traffic (C_T)
output : List of MAC address (MAC), IP address (IP), source port (SP)
         and destination port (DP)
N ← Total number of classes of attack
status ∈ 0,1,..N /* 0 normal traffic; 1...N malicious
  traffic                                              */
status ← trained_controller(P_T, C_T)
if status == 1 then
    Extract MAC, IP addr, src & dest port from P_T, C_T
    Return MAC, IP, src & dest port
end
if status == 0 then
    wait
end
```

Algorithm 4: Attack mitigation

```
input  : List of MAC address (MAC), IP address (IP), source port (SP)
         and destination port (DP)
output : Flow Rules (FR)
TOP ← Network topology /* Network topology comprises
  of list of switches and the access points
  connected to it and the devices associated with
  the access points                                    */
AP ← Access point
SW ← Openflow switch
T ← Flow rule timeout value
while (true) do
foreach a ∈ MAC do
    Search a in TOP_MAC
    AP_a ← Access point associated with a
    SW_a ← Openflow switch connecting the access point
    FR ← src MAC=a; src IP=*; action=drop; hard_timeout=T
    Install FR on SW_a
end
end
```

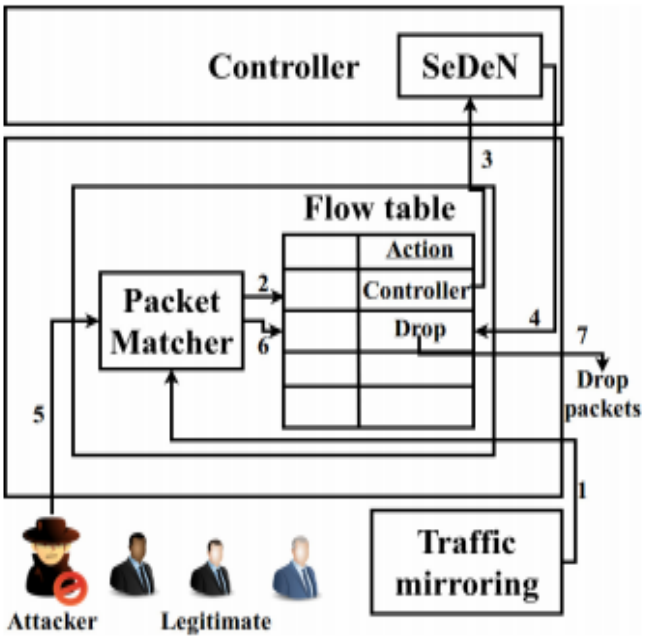

Fig 1. PrDeN- attack mitigation.

IV. EVALUATION

To evaluate the PrDeN framework we evaluated the framework in mininet-wifi emulator. Figure 2 illustrates the topology emulated for evaluation purpose. The emulated topology was run on the system with the configuration- "Ubuntu 16.04 64 bit

International Journal of Science, Engineering and Technology

An Open Access Journal

OS; Intel core i7-7500U CPU@ 2.70 GHz and 8GB RAM". Pox controller was used as the SDN controller to set rules on the wireless network. The controller was trained with AWID dataset [17]. AWID dataset is a wireless intrusion detection dataset that contains both captured normal wireless traffic as well as traffic captured under various attack scenarios such as amok, beacon, arp etc. This dataset was used to train the pox controller with LSTM RNN algorithm. For experimentation purpose we generated arp flooding attack, beacon flooding attack, evil twin attack with Nping and Wifi phisher tools across the wireless host in the data layer.

We compared our proposed framework with neural network running at the core of theframework to train the controller i.e. neural network takes up only the current traffic and does not take into account the previous traffic. The framework was also compared with no PrDeN framework running in the controller in order to evaluate the effectiveness of the network performance after the installation of the PrDeN in the controller. The various performance metrics used are as follows:

- Attack detection time: Attack detection time metric is used to find out how efficient the framework is in detecting the attack and avoiding the disruption of normal working
- Accuracy: Accuracy illustrates how meticulous the framework is in evaluating the malicious wireless nodes.
- Packet loss rate: Packet loss rate illustrates the percentage of loss of genuine packets.

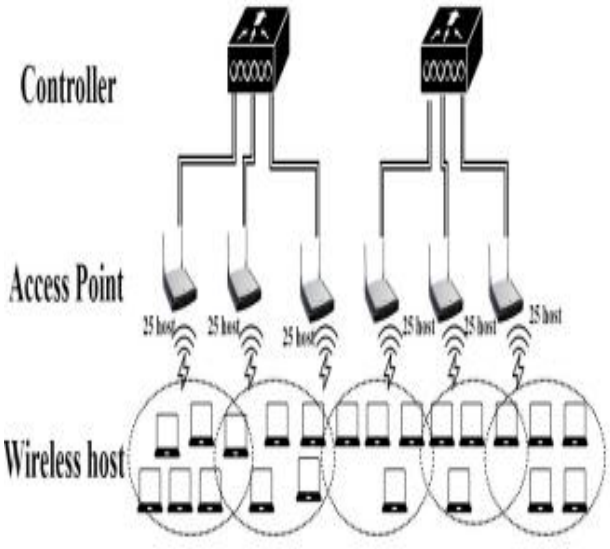

Fig 2. Emulated topology.

V. RESULTS AND DISCUSSION

1.Training- For the system to be trained effectively there are various parameters that need to be set to optimal value. The various parameters include count of epochs, count of samples to be binned and traffic parameters that are correlated with attack detection. Figure 3 shows the optimal count of epochs required for training the controller. The optimal count of epochs is the point at which the mean square error (η) in training is least. Mean square error is given by equation 1. Least η indicates the efficiency in detection of attack. It can be seen from figure 3 that η is least when count of epoch is 1000.

$$\eta = \frac{(\sum_{a=1}^{\#test\ data}(actual\ class - detected\ class)^2)}{\#test\ data} \quad (1)$$

Figure 4 shows the optimal number of samples to be binned for the controller to evolve into better trained model. Binning is necessary for efficient training of the sequential traffic data. As seen from figure 4 the optimal number of samples is found to be 96 that provides least error. Figure 5 shows the correlation obtained for few traffic parameters. For instance parameters like data rate, interval of beacon, retry, length of frame etc. showed positive correlation.

Fig 3. Optimal count of epoch.

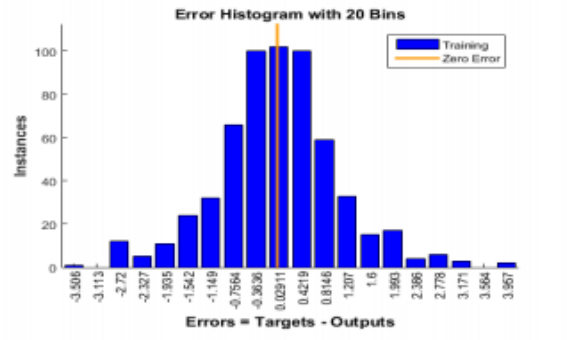

Fig 4. Optimal count of samples for binning.

International Journal of Science, Engineering and Technology

An Open Access Journal

While the parameters like marking of frame, time shift of frame, frame interface id etc. showed negative correlation. The parameters that showed positive correlation was taken up for training and other parameters were ignored for training the controller.

2. Performance Analysis: Figure 6 shows the time taken to detect the attack with RNN and neural network trained controller over several runs of the test. RNN took less time than the neural network because the RNN detects attack based on the previous observed traffic. Thus Pr DeN framework is able to detect earlier when the attack starts propagating itself.

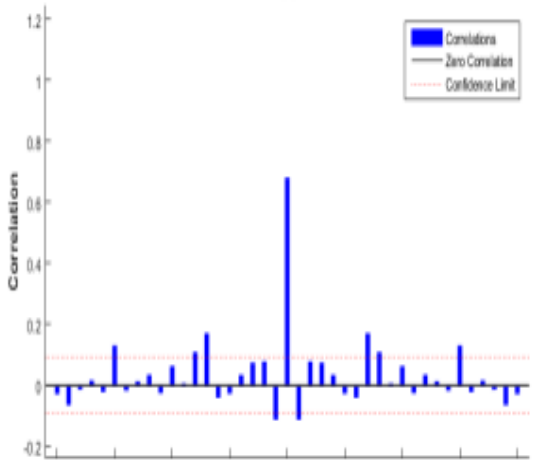

Fig 5. Correlation.

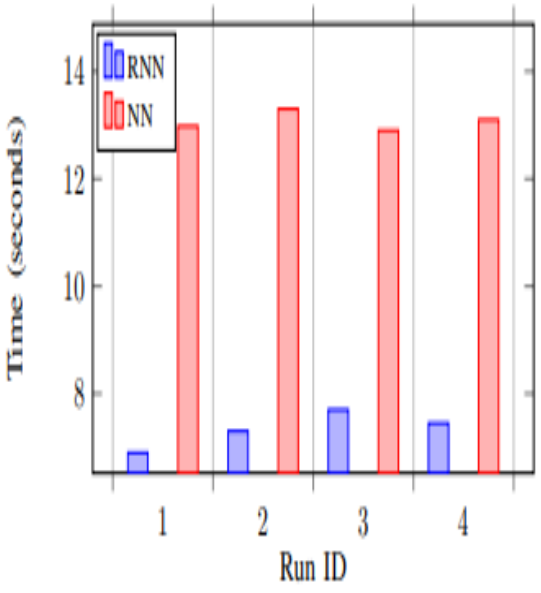

Fig 6. Attack detection time- RNN. NN.

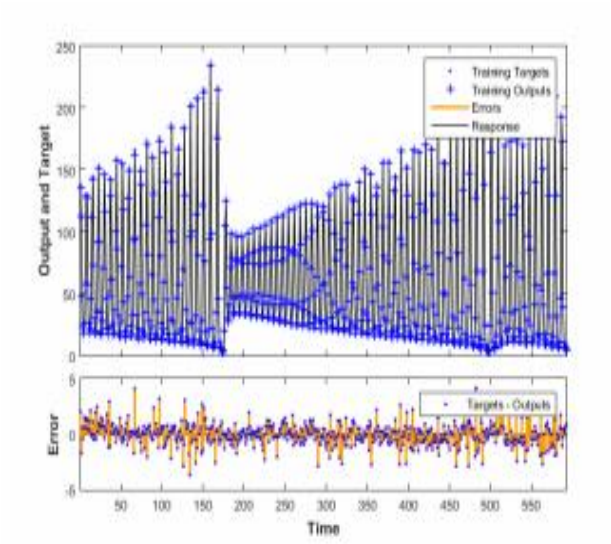

Fig. 7. Detection of class of traffic

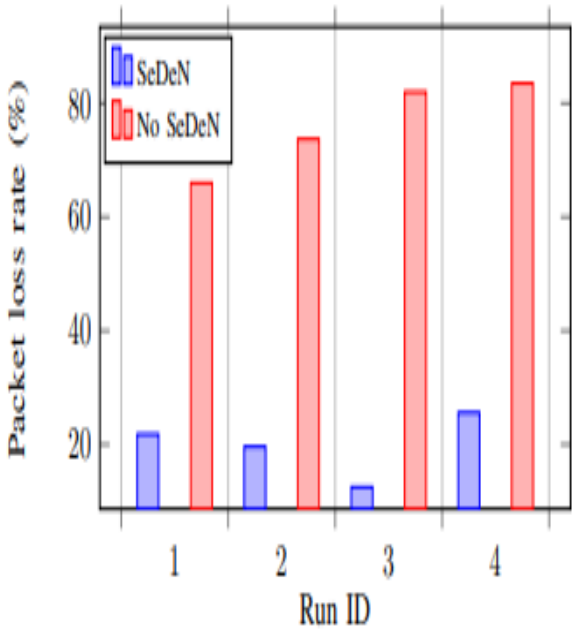

Fig 8. Packet loss %- PrDeN. No PrDeN.

Figure 7 shows the detection of traffic and error observed over a period of time (seconds) during evaluation. We were able to achieve 97.8% accuracy in detecting the attack and the controller was able to mitigate it by setting flow rules for dropping packets emanating from the malicious nodes. Figure 8 shows the packet loss rate in percentage observed during attack times in case of PrDeN framework running in the controller and without the PrDeN framework running in controller. It can be seen that in cases of PrDeN framework the attack was detected and mitigated which led to minimal packet loss percentage. But without PrDeN, the packet loss

percentage was high as the controller did not detect malicious traffic and it set flow rules to allow the huge malicious traffic. This eventually led to the drop of legitimate data packets.

VI. CONCLUSION AND FUTURE WORK

We have designed a secure deep learning based framework for 5G wireless network and the results showed better accuracy in attack detection. The SDN was able to mitigate the attack at the source without further propagation. As a future work we plan on appending modules onto the framework to fine tune the attack detection. Like probing onto other features that is strongly correlated with the attack. Then further workon evolving deep learning models suitable for 5G wireless networks.

REFERENCE

[1]. M. Agiwal, A. Roy and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," in IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 1617-1655, third quarter 2016.

[2]. A. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies," in IEEE Access, vol. 3, pp. 1206-1232, 2015.

[3]. N. Ravi and M. S. Selvaraj, "Te FENS: Testbed For Experimenting Next Generation-Network Security," 2018 IEEE 5G World Forum (5GWF), Silicon Valley, CA, 2018, pp. 204-209.

[4]. D. Fang, Y. Qian and R. Q. Hu, "Security for 5G Mobile Wireless Networks," in IEEE Access, vol. 6, pp. 4850-4874, 2018.

[5]. Mohamed Amine Ferrag, Leandros Maglaras, Antonios Argyriou, Dimitrios Kosmanos, Helge Janicke, "Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes," in Journal of Network and Computer Applications, vol. 101, pp. 55-82, ISSN 1084-8045, 2018.

[6]. Ian F. Akyildiz, Shih-Chun Lin, Pu Wang, "Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation," in Computer Networks, vol. 93, Part 1, pp. 66-79, ISSN 1389-1286, 2015.

[7]. Z. Zaidi, V. Friderikos, Z. Yousaf, S. Fletcher, M. Dohler and H. Aghvami, "Will SDN Be Part of 5G?," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3220-3258, Fourthquarter 2018.

[8]. Habib Mostafaei, Michael Menth, Software-defined wireless sensor networks: A survey, Journal of Network and Computer Applications, vol. 119, pp. 42-56, ISSN 1084-8045, 2018

[9]. S. Khan Tayyaba and M. A. Shah, "5G cellular network integration with SDN: Challenges, issues and beyond," 2017 International Conference on Communication, Computing and Digital Systems (C-CODE), Islamabad, 2017, pp. 48-53.

[10]. Liyanage, M., Ahmad, I., Okwuibe, J., de Oca, E. M., MAI, H. L., Perez, O. L., & Itzazelaia, M. U., "Software Defined Security Monitoring in 5G Networks. A Comprehensive Guide to 5G Security," pp. 231-243, 2018.

[11]. J. H. Cox, R. Clark and H. Owen, "Leveraging SDN and WebRTC for Rogue Access Point Security," in IEEE Transactions on Network and Service Management, vol. 14, no. 3, pp. 756-770, Sept. 2017.

[12]. M. Liyanage et al., "Enhancing Security of Software Defined Mobile Networks," in IEEE Access, vol. 5, pp. 9422-9438, 2017.

[13]. V. Varadharajan, U. Tupakula and K. Karmakar, "Secure Monitoring of Patients With Wandering Behavior in Hospital Environments," in IEEE Access, vol. 6, pp. 11523-11533, 2018.

[14]. Sharma, P. K., Singh, S., & Park, J. H. (2018). OpCloudSec: Open cloud software defined wireless network security for the Internet of Things. Computer Communications, 122, 1-8.

[15]. W. Dai et al., "TNGuard: Securing IoT Oriented Tenant Networks Based on SDN," in IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1411- 1423, June 2018.

[16]. Salakhutdinov, R. and Larochelle, H., "Efficient learning of deep Boltzmann machines", In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 693-700, 2010.

[17]. C. Kolias, G. Kambourakis, A. Stavrou and S. Gritzalis, "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset," in IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 184-208, First quarter 2016.