Intelligent Distributed Privacy-Preserving DDOS Attack Detection Using Software Defined Networks

Dr.P.Panimalar[#1], R.Bhuvaneswari*[2]

[#1]Assistant Professor, Department of Computer Science, A.V.C Collage (Autonomous), Mannampandal, Mayiladuthurai.

[*2]Research Scholar, Department of Computer Science, A.V.C Collage (Autonomous), Mannampandal, Mayiladuthurai.

[1]panimalarkp@gmail.com,[2]bhuvaneswari7694@gmail.com

Abstract-In the Software Defined Networking (SDN)-based wireless network is an enabling technology in modern transportation systems for providing safety and valuable information, and yet vulnerable to a number of attacks from passive eavesdropping to active interfering. In realism, anomalous traffic usually affects many network domains. Thus, cross-domain attack detection has been proposed to improve detection performance. However, when participating in finding, the domain of each SDN wants to make available a large quantity of real traffic information, from which confidential information may be dripped. To this conclusion, dispersed machine learning is a proper outline for the design of scalable and implementable combined detection algorithms over the wireless system. One primary barrier to combined learning is the confidentiality concern as nodes swap data among them. A malevolent node can obtain sensitive in order of other nodes by understanding from the observed data. In this paper, we propose privacy-preserving Machine Learning based Collaborative Intrusion Detection System (ML-CIDS) for SDN based Wireless Networks. The projected algorithm deals with the Alternating Direction Method of Multipliers (ADMM) to a class of Empirical Risk Minimization (ERM) trouble and trains a classifier to detect the interruptions in the wireless systems. We use the differential privacy to capture the privacy notation of the ML-CIDS and propose a method of dual variable perturbation to provide dynamic differential privacy. We analyze theoretical performance and characterize the fundamental trade-off between the security and privacy of the ML-CIDS. Efficiency of the proposed architecture is validated in the widely used network simulator- NS-2. Our results demonstrate the network throughput enhancement achieved by the proposed ML-CIDS architecture.

Key words-Software Defined Network, Cross domain attack detection, Privacy preserving Machine Learning based Collaborative Intrusion Detection System , Alternating Direction Method of Multipliers, Empirical Risk Minimization.

I. INTRODUCTION

Software Defined Networks (SDNs) have emerged as a new networking paradigm that is liberated from vertical integration in traditional networks and offers programs and networks flexibility via a centralized logical network controller. The controller abstracts an entire network view into network services and provides an easy-to-use interface for operators to facilitate customization of privatization applications and realize the logical management of a network. Users of an SDN do not need to worry about the technical details of the underlying devices. SDNs simplify network management and adapt to current situations in which the network size continues to rapidly expand.

A. Software Defined Networking

Software Defined Networking (SDN) technology is an approach to cloud computing that facilitates network management and permits programmatically economical network configuration so as to boost network performance and observance as shown in figure 1. SDN is supposed to deal with the actual fact that the static design of ancient networks is redistributed and sophisticated whereas current networks need a lot of flexibility and simple troubleshooting.SDN suggests integrative network intelligence in one network part by disassociating the forwarding method of network packets (data plane) from the routing method (control plane). The management plane consists of 1 or a lot of controllers that are thought of because the brain of SDN network wherever the full intelligence is incorporated. However, the intelligence centralization has its own drawbacks once it involves security, quantifiability and snap and this can be the most issue of SDN.

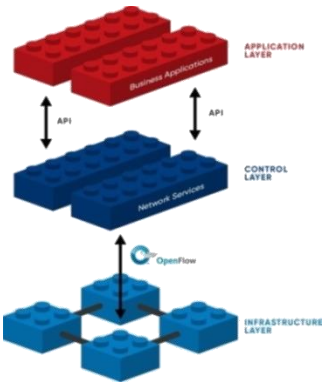

Figure 1: SDN architecture

SDN was ordinarily related to the OpenFlow protocol (for remote communication with network plane parts for the aim of deciding the trail of network packets across network switches) since the latter's emergence in 2011. However, since 2012 OpenFlow for several firms is not {any|isn't any} longer an exclusive resolution, they value-added proprietary techniques. These embody Cisco Systems' Open Network atmosphere and Nicira's network virtualization platform.

B. DDoS

DDoS is brief for Distributed Denial of Service. DDoS could be a sort of DOS attack wherever multiple compromised systems, that area unit usually infected with a Trojan, area unit wont to target one system inflicting a Denial of Service (DoS) attack. Victims of a DDoS attack comprises each the tip targeted system and every one systems maliciously used and controlled by the hacker within the distributed attack.

A Distributed Denial-of-Service (DDoS) attack could be a malicious conceives to disrupt traditional traffic of a targeted server, service or network by overwhelming the target or its close infrastructure with a flood of net traffic as shown in figure 2. DDoS attacks accomplish effectiveness by utilizing multiple compromised pc systems as sources of attack traffic. Exploited machines will embody computers and different networked resources like IoT devices. From a high level, a DDoS attack is sort of a tie up impeding up with route, preventing regular traffic from inward at its desired destination.

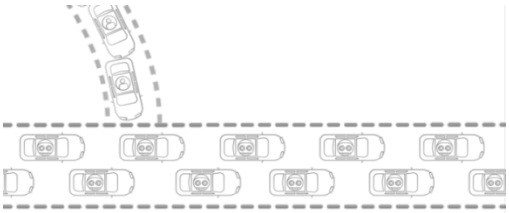

Figure 2: DDoS

In a DDoS attack, the incoming traffic flooding the victim originates from many various sources – doubtless many thousands or additional. This effectively makes it not possible to prevent the attack just by obstruction one scientific discipline address; and, it's terribly tough to differentiate legitimate user traffic from attack traffic once unfold across such a large amount of points of origin.

A DDoS attack needs relate in attention assaulter to achieve management of a network of on-line machines so as to hold out relate in attention attack. Computers and different machines (such as IoT devices) square measure infected with malware, turning all into a larva (or zombie). The assaulter then has remote over the cluster of bots, which is termed a botnet.

Once a botnet has been established, the assaulter is ready to direct the machines by causing updated directions to every larva via a way of remote. Once the scientific discipline address of a victim is targeted by the botnet, every larva can respond by causing requests to the target, doubtless inflicting the targeted server or network to overflow capability, leading to a denial-of-service to traditional traffic. As a result of every larva could be a legitimate net device, separating the attack traffic from traditional traffic is tough.

Cross-domain attack detection will cause privacy leakage, whereas the introduction of privacy protection is usually characterized by excessive time consumption and a low detection rate. We should address these challenges when detecting DDoS attacks in a cross-domain. The first challenge is to conduct cross-domain DDoS attack detection in SDNs without revealing privacy of each network domain. Attack detection classifiers require detailed traffic data, and SDNs domains do not trust each other. We must address the privacy issue when multiple SDNs domains work together to perform anomaly detection. The second challenge is to ensure efficient and accurate DDoS attack detection while preserving privacy. The trade-off for strong privacy protection in multi-party cooperation is low accuracy and high time consumption, and assigning priority to involved parties is challenging. To address these dilemmas, we decouple the detection into two step— disturbance and detection—and introduce two servers that collaborate to complete the detection process.

The remainder of the paper is organized as follows: We review related work in Section II and describes the proposed ML-CIDS architecture in Section III also its presents the model of the collaborative learning over a wireless network for ML-CIDS. We present the implementation and the experimental results in Section IV. We present a results and discussion in Section V and also conclude this paper.

II. LITERATURE SURVEY

M. Shen, B. Ma, [1] Constrained Shortest Distance (CSD) questioning is one in every of the elemental graph query primitives, that finds the shortest distance from associate degree origin to a destination in a very graph with a constraint that the whole value doesn't exceed a given threshold. CSD querying contains a big selection of applications, like routing in telecommunications and transportation. A completely unique graph secret writing theme that allows approximate CSD querying.Connor is made supported associate degree economical, tree-based cipher text comparison protocol, and makes use of symmetric-key primitives and therefore the somewhat homomorphic secret writing, creating it computationally economical.

Y. Khettab, M. Bagaa, [2]The future 5G systems ought to meet diverse requirements of new industry verticals, such as Massive web of Things (IoT), broadband access in dense networks and immoderate reliable communications. Network slicing is a vital idea that's expected to support these 5G verticals and deal with the conflicting necessities of their various services. associate design which will explore however each

Network perform Virtualization (NFV) and package outlined Networking (SDN) could also be leveraged to secure a network slice on-demand, addressing the new security issues obligatory to the network management by the pliability and physical property support.

D. B. Rawat, [3] SDN is a rising paradigm, that breaks the vertical combination in ancient networks to supply the pliability to program the network through (logical) centralized network management. SDN provides security, energy potency, and network virtualization for enhancing the general network performance. We have a tendency to gift numerous security threats that area unit resolved by SDN and new threats that arise as a results of SDN implementation. The recent security attacks and countermeasures in SDN are summarized within the type of tables. We have a tendency to conjointly offer a survey on the various ways that area unit enforced to realize energy potency and network security through SDN implementation.

Also [4] many data-driven personalized services require that private data of users is scored against a trained machine learning model. In this protocol for privacy preserving classification of decision trees, a popular machine learning model in these scenarios. By merging several of the structure blocks for our assessment tree categorization procedure, we also progress formerly proposed solutions for categorization of support vector machines and logistic regression models. These protocols are information theoretically secure and, unlike previously proposed solutions do not require modular exponentiations. We show that our protocols for privacy-preserving classification lead to additional resourceful grades from the direct of vision of computational and statement complexities.

M. Shen, M. Wei, L. Zhu, [5] with a profusion of network applications, traffic classification plays a crucial role in network management and policy-based security control. The wide used encoding transmission protocols, like the Secure Socket Layer/Transport Layer Security (SSL/TLS) protocols, cause the failure of ancient payload-based classification ways. AN attribute-aware encrypted traffic classification technique supported the second-order Andrei Markov Chains. We tend to begin by exploring approaches which will more improve the performance of existing ways in terms of discrimination accuracy, and build promising observations that the appliance attributes written word, that consists of the certificate packet length and also the initial application knowledge size in SSL/TLS sessions, contributes to application discrimination.

X. Shu, D. Yao, [6] Statistics from security firms, research institutions and government organizations show that the number of data-leak occurrence has grown rapidly in recent years. Among various data-leak cases, human mistakes are one of the important causes of data loss. In this privacy preserving data-leak detection (DLD) solution to solve the issue where a special set of sensitive data digests is used in detection. The advantage of our method is that it enables the data owner to safely delegate the detection operation to a semi honest provider without revealing the sensitive data to the provider.

Due to decoupled control and data plane, SDN can handle this increasing number of attacks by blocking those network connections at the switch level. However, the challenge lies in determine the set of rules on the SDN controller to block malicious network connections. Using machine learning algorithms, prepared on historical network attack data, to identify the potential malicious connections and potential attack destinations.

A. Existing System

Many schemes for DDoS attack detection in traditional networks have been proposed and have demonstrated promising results. Extensive solutions for DDoS attack detection in SDNs, which employ various detection techniques, such as Bayesian networks, the entropy variation of the destination IP address, and the Support Vector Machine (SVM) model, have also been proposed. Although these DDoS attack detection schemes are usually restricted to a single domain, few studies have considered cross-domain attack detection.

Bian et al. proposed a scheme for cross-domain DDoS attack detection in SDNs using a Self Organizing Map (SOM) as the traffic classifier. The calculation in the training and the test phases, however, is complicated and requires multiple vector multiplication or complex division. Secure Multi-party Computation (SMC) may enable secure cross-domain anomaly detection (e.g., secure addition protocol, secure multiplication protocol, and secure compare protocol). These protocols, however, require numerous interactions among the participants and calculations on cipher text, which undoubtedly consumes a significant percentage of the controller's bandwidth.

A Denial of Service (DoS) attack can expire the resources of a system on the target computer, stop services and leave its normal users inaccessible. When hackers use two or more compromised computers on the network as puppet machines to launch DoS attacks on a specific target, these attacks are referred to as DDoS attacks.

Numerous studies of DDoS attack detection exist due to the severity and prevalence of DDoS attacks. Here, we briefly summarize related studies from two perspectives, i.e., DDoS attack detection in conventional networks and DDoS attack detection in SDNs.

1. Detection in Conventional Networks

Detection approaches of DDoS attacks in conventional networks have been extensively investigated; methods such as entropy-based, SVM, naive Bayesian, neural network, cluster analysis, artificial neural network (ANN), and kNN methods, are employed as classifiers.

2. Detection in SDNs

An SDN controller collects information about flow table and uses selected classifiers to classify network traffic flows as either normal or abnormal. Based on the capability of a logical centralized controller and the programmability of a network, network administrators can immediately respond to attacks. Classic classification methods, such as Bayesian networks and SVMs, as well as neural networks of SOMs and deep learning, are employed as traffic classifiers in SDNs.

An SDN domain in Predis refers to a controlled domain in an SDN architecture, which is a network domain with the deployment of SDN techniques that can be independently controlled by operators. The domains of the SDNs conduct centralized control of data forwarding. The domains of the multiple SDNs described in our article collaborate and may be adjacent to a physical or geographical location. The control plane of a centralized SDN domain sends a flow table to a specified location (i.e., computing server). The computing server provides the DDOS detection service and returns detection results controllers. Traditional network domains for traffic forwarding are a distributed control and cannot achieve centralized control.

III. PROPOSED SYSTEM

In this paper the proposed system should address following challenges when detecting DDoS attacks in a cross-domain. The first challenge is to conduct cross-domain DDoS attack detection in SDNs without revealing privacy of each network domain. Attack detection classifiers require detailed traffic data, and SDNs domains do not trust each other. We must address the privacy issue when multiple SDNs domains work together to perform anomaly detection. The second challenge is to ensure efficient and accurate DDoS attack detection while preserving privacy. The trade-off for strong privacy protection in multi-party cooperation is low accuracy and high time consumption, and assigning priority to involved parties is challenging. To address these dilemmas, we decouple the detection into two step— disturbance and detection—and introduce two servers that collaborate to complete the detection process.

Each node is equipped with one local ML-CIDS agent as shown in Figure 3 to monitor its local activities including the ones in the AU and the communications via the OBU. Conceptually, the collaborative system consists of three main components, namely, pre-processing engine, a local detection engine, and Privacy-preserving Collaborative Machine Learning (P-CML) engine.

We employ perturbation encryption to protect the privacy of each network domain. With a careful design, the cipher text produced by perturbation encryption can be directly calculated in servers without the need to involving complex secure computation protocols. We leverage the Transport Layer Security (TLS) protocol to provide privacy and communications data integrity among the two servers and the SDN controllers. We decompose the distance calculation process into two step, which enables the well-known Euclidean distance1 using the perturbed traffic information. Our contributions are summarized as follows:

- We propose a machine-learning-based CIDS architecture to enable the collaborative information exchange and knowledge sharing in wireless networks.
- ML-CIDS employs the features of SDNs and the improved machine learning algorithm to accurately detect DDoS attacks within an effective time and applies perturbation encryption to provide confidentiality and ensure a participant's privacy.
- We use ADMM to capture the distributed nature of a network and construct a collaborative learning over a wireless network based on a regularized algorithm.
- By a rigorous security analysis, we prove that the traffic data provided by each participant are indistinguishable for a potential adversary.
- We conduct extensive experiments utilizing multiple authoritative datasets to demonstrate the timeliness and accuracy of Predis.

- We demonstrate that our scheme not only can determine if traffic is abnormal but also can identify abnormal traffic at the early stages of DDoS attacks.
- We provide a design principle to find the optimal value of the privacy parameter by solving an optimization problem to manage the tradeoff between security and privacy of a network.
- The results indicate that ML-CIDS is more accurate than an existing detection schemes and is capable of protecting participants' privacy.

A. DDoS Attacks

A Denial of Service (DoS) attack can expire the resources of a system on the target computer, stop services and leave its normal users inaccessible. When hackers use two or more compromised computers on the network as puppet machines to launch DoS attacks on a specific target, these attacks are referred to as DDoS attacks.

DDoS architecture that contains control puppets and attack puppets, as shown in Figure 3. Attackers usually employ a vast number of geographically distributed compromised hosts as puppet machines to launch DoS attacks on a specific target.attackers can simultaneously control several computers and create an attack

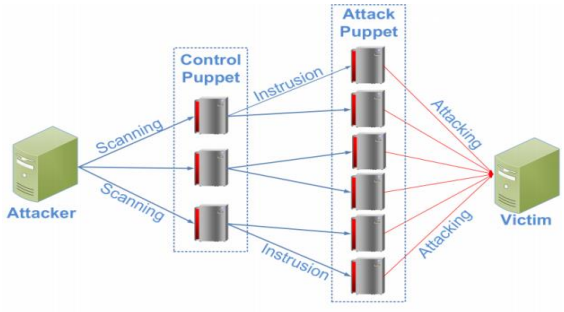

Figure 3: structure of DDoS Attacks

Traditional attack architecture is analogous to a dumbbell-shaped structure, in which an intermediate network is only responsible for data forwarding and security events and control functions are entirely conducted by management, whereas the network does not have the ability to detect and handle network attacks quickly

B. SYSTEM MODEL

Proposed Predis based SDN contains three roles: Computing Server (CS), Detection Server (DS) and SDN domains, as exhibited in Figure (a). Domain Dn is the n-th domain, which participates in attack detection, provides data to CS and DS.

The system sequence diagram is shown in Figure (b). Each domain sends traffic information to CS for calculation and receives the detection results from DS. CS provides computing service and sends the intermediate results to DS, where the latter provides detection service based on the intermediate results and replies the detection results to each domain. Thus, CS and DS perform collaboration to perform computations. Details of the computation and encryption in CS and DS are described.

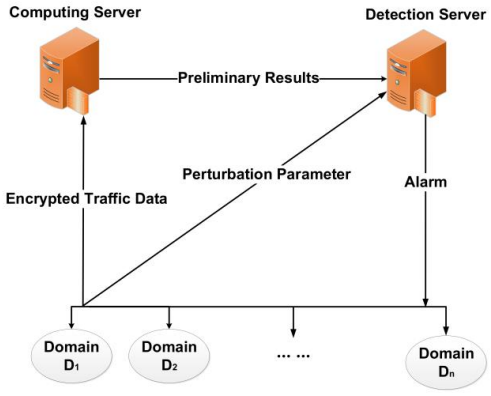

Figure 4: (a) System Overview

It will provide accurate DDoS attack detection service for domains, where each domain is unwilling to share privacy traffic information.

The information in the flow table pertains to domains that participate in the detection. Specifically, privacy includes the IP Source, IP Destination, Source Port, Destination Port, Length, and Flow Packets. We define the basic operations in Predis for the three roles as three functions with inputs and outputs. Each function is designed to run on continuous inputs in real time using the data partitioned into a certain time interval. SDN has a set of n input peers who want to jointly compute the final result of Predis on their private data without the slightest relevant disclosure. Which are referred to as privacy peers; they perform the computation of Predis by simulating a Trusted Third Party (TTP). Domains comprise both input peers and privacy peers, whereas both CS and DS are privacy peers.

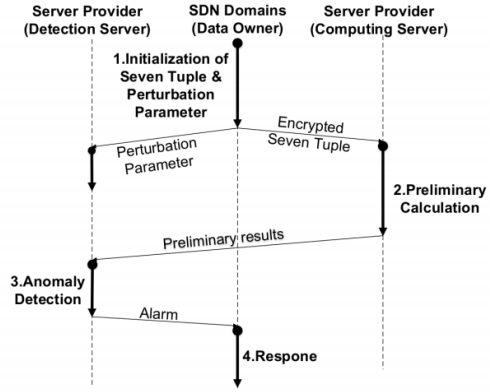

Figure 4: (b) System Flow

1. *Threat Model*

We abstract the cross-domain privacy-preserving DDoS attack detection problem with a threat model. The threat model has two types of adversaries, namely, the external adversary and the semi-honest adversary.

2. External adversary

Adversaries that illegally obtain the data in the transmission process for their purposes via Internet eavesdropping or data interception and other means.

3. Semi-honest adversary

A curious participant who properly follows the protocol to fulfil service functions but attempts to infer sensitive or private information from the intermediate results of calculation or even colludes with other participants.

In proposed, we employ TLS to build this secure channel, which provides the privacy and data integrity of two communicating entities. We received the semi-honest assumption for all privacy peers. Honest privacy peers follow the protocol and do not combine their information. Semi honest privacy peers follow the protocol but try to infer the input peers' privacy from the values that they learn and by combining their information. Domains hope to obtain the correct results of attack detection. While following the proper steps, some domains may try to infer other domains' privacy for certain purposes. CS and DS will provide the appropriate calculation service but may use the intermediate results that are generated by the intermediate steps in the calculation to infer and spy privacy from the domains.

The purpose of this paper is to obtain accurate cross domain DDoS attack detection results under the premise of privacy protection. Privacy peers may steal privacy as external adversaries or semi-honest adversaries. Privacy peers may collude with each other. In our solution, we allow at least one domain to collude with each other, with CS, and with DS. We make the following assumptions:

1. Each domain honestly performs functions but may have an interest in the private information of other domains.
2. CS or DS correctly performs the calculation process but may have an interest in obtaining domains' private information.
3. CS or DS may collude with at least one domain. Semi honest privacy peers follow the protocol but attempt to infer peers' privacy from the values that they learn. Thus, CS or DS may collude with at least one domain.
4. CS and DS do not conspire with each other. In actuality, DS and CS can be deployed by altered operators. Operators are likely to have difference of concern; thus, CS is assumed to not join together with DS.

C. Proposed Detection Algorithm

The logical flow of ML-CIDS is illustrated in Algorithm 1. The pre-processing engine gathers and pre-processes the real-time dynamic network system data that describe the system activities in a vehicle. The pre-processed system

Algorithm: ML-CIDS

Input: Real-time wireless node system data: Local audit data flow and activity logs.

Step 1: The pre-processing engine collects and pre-processes the real-time network system data, by numerical transformation, features selection, and data normalization.

if The classifier needs update then

Step 2: The P-CML engine is initiated and local training dataset is loaded. And updated classifier is obtained.

Step 3: The local detection engine uses the newly updated classifier to analyze the real-time network system data. If any activities are classified as intrusions, the local detection engine triggers the alarm.

else

Step 4: The local detection engine uses the current classifier to analyze the real-time network system data and triggers the alarm when any activities are classified as intrusions.

end if

To quantify these characteristics, five parameters are used in the feature selection module, including MPF, MBF, PCF, GOP and GSI, which are elaborated as follows:

1) *Median of Packets per Flow (MPF),* which describes the median number of packets in every n flow. We rank the flows X = {X1, X2 . . . . Xn} in ascending order based on the number of packets per flow, and then compute the median value.
2) *Median of Bytes per Flow (MBF),* which describes the median number of bytes in every n flow. We rank the flows in ascending order based on the number of bytes per flow and then compute the median value.
3) *Percentage of Correlative Flow (PCF),* which describes the number of flows with interactive features in every n flow. We define the flow X as X = (srcIP = A, dstIP = B) and flow Y as Y = (srcIP = B, dstIP = A), where X and Y use the same protocol. P CF = inactN/n, where inactN is the number of X flows in addition to the number of Y flows.
4) *Growth of Ports (GOP),* which describes the growth rate of the number of ports within a fixed time. GOP = portN/t, where t is the fixed time interval and portN is the number of port growth.
5) *Growth of Source IP Addresses (GSI),* which describes the growth rate of the number of source IP addresses within a fixed time. GSI = src IP N/t, where src IP N is the number of source IP addresses.

IV. SIMULATIONS

We simulate the user and system activities, the communication of proposed ML-CDS based network coding scheme and corresponding alternative route(s) are simulated in NS-2 discrete event simulator. The network consists of 30 nodes deployed in a 967m × 600m area. The mobility model is set as "Random Way Point Mobility Model" from NS-2 library. All the wireless nodes in the data plane are managed by a SDN controller.

In this paper, we first analyze the privacy link failure prediction error rate for five data flows with a flow rate of 4 Kbps, when the nodes move randomly with a speed of 0 to 5 meters/sec. In the prediction error rate is displayed in terms of the false alarm rate and the missed failure detection rate.

Table-I

SIMULATION PARAMETERS

Parameter	Value
Simulator	NS2
Simulation time	25s
Area	967 X1 600
Number of node	30
Physical Layer	IEEE 802.11
Routing protocol	AODV
Mobility model	Random way point
Radio type	802.11a/g
Transmission rate	1000 packets/s
Packet Size	1000
Pause time	0s

We have used following performance metrics for evaluating effects of attack and effectiveness of our detection algorithm:

Throughput:

It is the ratio of the total number of bits transmitted (Btx) to the time required for this transmission, i.e. the difference of data transmission end time and start time (tstart). This metric depicts how the congestion control mechanism at the source node is affected by the packet losses caused by malicious-nodes. A decrease in throughput is an outcome of any attack.

$$\text{Throughput} = (B_{tx})/ (t_{end} - t_{start}) \text{ bps}$$

Packet Delivery Ratio:

This is defined as the number of packets received at the destination and the number of packets sent by the source. Here, $pktd_i$ is the number of packets received by the destination node in the ith application, and $pkts_i$is the number of packets sent by the source node in the ith application.

Average End-to-End Delay:

It is average transmission delay of packets transmitted from source to destination. D is computed as the ratio of the sum of individual delay of each received data packet to the total number of data packets received. This metric is used to evaluate impact of a attack on delay-sensitive applications of TCP-based wireless networks. By intentionally discarding, delaying or reordering packets, a node can increase the value of this metric; increase being caused by re-transmissions of such packets due to timeout at TCP source.

D= no.of received packed/total time

V. Results and Discussion

In this module, to enable all the nodes to get the global information to train model, we propose a proposed algorithm that under general dynamic model to update the number of path available, there exists an optimal policy consisting of time-invariant routing probabilities in the network and these can be obtained by Link Failure Measure and Perdition.

Table II

Numerical Results of Network Parameters for Existing and Proposed Techniques

Particulars	**Delay**	**Throughput**	**Remain. Energy**
Existing	0.107	19931.47	78.21
Proposed SDN	0.041	28050.00	93.22

In this module, the performance of the proposed SDN-Network coding method is analyzed. Based on the analyzed results X-graphs are plotted. Throughput, delay, energy consumption are the basic parameters considered and X-graphs are plotted for these parameters.

At last, the results obtained from this module is compared with previous results and comparison X-graphs are plotted. Form the comparison result, final RESULT is concluded.

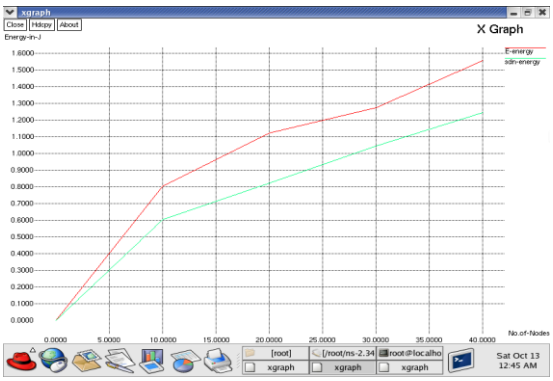

Figure 5: Energy vs. No. Of Nodes

To further investigate performance of the proposed DDoS prediction model, we consider multiple data flows with different data rates and mobility. The Figure 5 shows the utilization of energy in the SDN ML-CIDS Network proposed topology.The performance of proposed TLS scheme with traditional distributed network control.

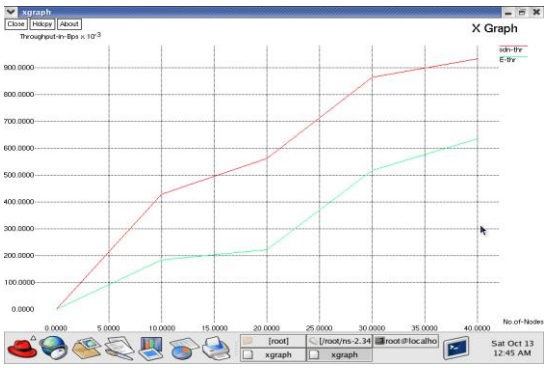

Figure 6: Throughput vs. No. Of Nodes

Figure 6 shows the response time for node mobility events, and is compared with the AODV-based SDN ML-CIDS networks. This experiment considers the link failure issue for a single data flow, and the influence of other background traffic is not considered here. As shown in figure 7, the proposed system switches the path quickly when a mobility event occurs at 30s. Due to the precise mobility detection in our scheme, the nodes can find the alternative paths in advance and avoid unnecessary traffic congestion due to link failure. On the other hand, the AODV-based SDN takes around 10 seconds to resume the data flow transmission.

Our ML-CIDS scheme is designed to deal with network uncertainty (due to node mobility), in terms of optimizing the traffic load in a global view. To validate this functionality, the maximum link utilization is observed, which represents the upper bound of utilization for every link in the network.

Finding new routes can introduce significant route discovery overhead due to node mobility in dynamic network topology. To control the routing overhead, the ML-CIDS scheme finds alternative routes based on the

link failure prediction, instead of frequently broadcasting the route discovery messages.

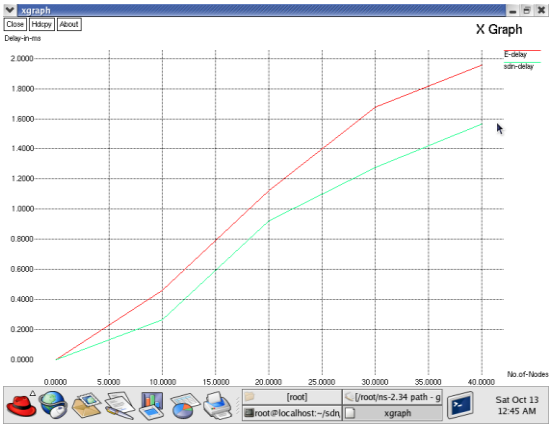

Figure 7: Delay vs. Nodes

As shown 7 in Figure, the proposed ML-CIDS scheme generates lower normalized routing overhead than the existing routing.

VI. CONCLUSION

In this paper, we have described architecture for a collaborative intrusion detection system using privacy-preserving distributed machine learning. The privacy-preserving scheme for the distributed collaborative-based learning is essential for achieving a private collaboration; otherwise, the distributed machine learning itself creates privacy leakage of the training data. We have proposed a privacy-preserving Machine-Learning based Collaborative Intrusion Detection System (ML-CIDS) for SDN based Wireless Networks. The Alternating Direction Method of Multipliers (ADMM) approach is used to decentralize the Empirical Risk Minimization (ERM) problem that models the collaborative learning into the distributed ERM well-suited to the nature of the wireless network system. Extensive experimental results revealed that Predis is capable of detecting cross-domain anomalies while preserving privacy with low time consumption and high accuracy. The entire system was validated in the NS-2 simulator, and our results showed that the performance of software defined wireless network is significantly better than conventional schemes.

REFERENCES

[1] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu, "Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection," IEEE Transactions on Information Forensics and Security, vol. 13, no. 4, pp. 940–953, April 2018

[2] Y. Khettab, M. Bagaa, D. Dutra, T. Taleb, and N. Toumi, "Virtual security as a service for 5g verticals," in IEEE Wireless Communications and Networking Conference, 2018.

[3] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: A survey," IEEE Communications Surveys Tutorials, vol. 19, no. 1, pp. 325–346, Firstquarter 2017

[4] M. D. Cock, R. Dowsley, C. Horst, R. Katti, A. Nascimento, W. S. Poon, and S. Truex, "Efficient and private scoring of decision trees, support vector machines and logistic regression models based on precomputation," IEEE Transactions on Dependable and Secure Computing, vol. PP, no. 99, pp. 1–1, 2017.

[5] M. Shen, M. Wei, L. Zhu, and M. Wang, "Classification of encrypted traffic with second-order markov chains and application attribute bigrams," IEEE Transactions on Information Forensics and Security, vol. 12, no. 8, pp. 1830–1843, Aug 2017

[6] X. Shu, D. Yao, and E. Bertino, "Privacy-preserving detection of sensitive data exposure," IEEE Transactions on Information Forensics and Security, vol. 10, no. 5, pp. 1092–1103, May 2015.

[7] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, and B. Yang, "Predicting network attack patterns in sdn using machine learning approach," in 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 167–172, Nov 2016.