Flow Based Intrusion Detection System for Software Defined Networking using Hybrid Machine Learning Technique

K.Muthamil Sudar, P.Deepalakshmi

Abstract-Software Defined Networking and OpenFlow protocol have been recently emerged as dynamic and promising framework for future networks. Even though, programmable features and logically centralized controller leads to large number of security issues. To address the security problems, we have to impose Intrusion Detection System module to continuously keep track of the network traffic and to detect the malicious activities in the SDN environment. In this paper, we have implemented flow-based IDS with the help of hybrid machine learning technique. By collecting the flow information from the controller, we classify the traffic, extract the essential features and classify the attack using machine learning based classifier module. For classifier, we have developed hybrid machine learning model with the help of Modified K-Means and C4.5 algorithm. Our proposed work is compared with single machine learning classifier and our experimental results show that, proposed work can classify the normal and attack instances with accuracy of 97.66%.

Keywords – Software Defined Networking, SDN, Machine Learning, ML, Intrusion Detection System, IDS, flow-based, K-Means, C4.5, hybrid ML

I. INTRODUCTION

Traditional Networks are challenging and difficult to manage. Since traditional networks are vertically interspersed, it becomes more complicated to manage the dynamic requirements. In traditional networks, both control and data planes are packed inside the devices which reduce the flexibility to make dynamic changes in network environment. To impose and transform high level security policies into low level vendor specific policies, network operators have to perform manual configuration for each device which leads to increase in configuration complexity. To overcome these problems, a new era of networking is developed, called Software Defined Networking(SDN) [1]. It is an emerging paradigm which overcomes the complexity in traditional networks by breaking the logic of vertical integration of control plane and data plane. The control plane decides how to manage the traffic and the data plane forwards the traffic as per the decision made by the controller. This separation makes the network devices as a forwarding elements and the control logic will be taken care by centralized controller or Network Operating System (NOS).

This separation is successfully achieved by enabling properly defined programming interface between network devices and the controller. The two planes can make the communication through the OpenFlow (OF) protocol itemized by Open Networking Foundation(ONF) [2]. The forwarding devices in OpenFlow are called as OpenFlow switches which makes use of flow-based forwarding mechanism instead of destination-based. An OpenFlow switch contains more number of flow tables consisting of packet handling rules that determines how to process the packets by OF switches. This logically centralized controller offers more additional benefits such as simple and error free while modifying the network policies and the controller also gains the universal knowledge of the network which facilitates the dynamic changes in the networking services and functions. But the concept of SDN also introduces some security issues. Due to the centralized nature of controller, it is a major threat if an attacker takes control over the controller.

To address security issues, we need to impose a security module in the SDN framework called Intrusion Detection System (IDS) [3]. IDS help to keep track of the SDN environment, inspect the overall traffic and send alert when malicious activities happened in the network. An IDS is mainly classified into two types such as signature-based and anomaly-based. Signature-based IDS is not proficient in detecting unknown attacks. It generally makes use of patterns in the given data to detect the known attacks. Anomaly-based IDS helps to detect the unknown attacks by analyzing the deviation from normal behaviour in the network. Signature based IDS helps to detect the known attacks very easily and false alarm rate is also very low. Anomaly based IDS helps to detect the unknown attacks with high false alarm rate. In modern days, an increasing number of new attacks are triggered by the attackers. So, it is essential to develop anomaly based intrusion detection system to detect the attacks in the early stage and at the same time false alarm should be low.

Revised Manuscript Received on December 15, 2019.
***** Correspondence Author
K.Muthamil Sudar*, Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnankoil, India. Email: k.muthamilsudar@klu.ac.in
P.Deepalakshmi, Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnankoil, India. Email: deepa.kumar@klu.ac.in

Machine Learning (ML) [4] techniques work much better in providing promising solution compared to statistical methods. In this paper, we have proposed hybrid machine learning techniques to construct an efficient IDS by analysing the flow information. In general, IDS performs deep packet inspection to identify the malicious flows but it is very complex in high speed networks. So, to overcome this problem, we have implemented flow based IDS by adoptingK-Means clustering and C4.5 to construct the attack classification module.

II. RELATED WORK AND BACKGROUND

A. Related Work

In recent years, security issues in SDN environment have captured attention of many researchers. In[5], the authors proposed machine learning based classifier for real time IDS for high speed big data environments. They achieved high accuracy and efficiency by extracting nine important features and by using J48 classifier. They have used NSL-KDD dataset for evaluation of their proposed work. The authors of [6], proposed threshold-based mechanism to detect the malicious activities in home networks by revisiting the traffic. They concluded that the proposed algorithms can detect the attacks in home network with high accuracy and minimum CPU usage. In [7], the authors proposed the entropy-based mechanism to detect the Distributed Denial of Service (DDoS) attack in the SDN environment. They have used flow statistics information from the controller to compute the entropy value. They have implemented this detection module in SDN switches and concluded that their work reduces controller overload problem and also provides high prediction accuracy. The authors of [8], proposed a deep learning based approach to detect the DDoS attack in SDN environment. They have used stacked encoder (SAE) for feature extraction and classification and stated that proposed work can detect DDoS attack with an accuracy of 95.65%. In [9], the authors used fuzzy system to construct an efficient IDS for SDN environment. They have evaluated their model using KDD Cup 99 dataset and conveyed that threshold based methods can be used to detect the attacks in earlier stage and fuzzy logic to detect the attacks with high accuracy.

The authors of [10], proposed a flow-based statistical method for network anomaly detection in traditional networks. They have used neural classifier to detect the intrusions. They concluded that, flow-based technique reduces the amount of continuous monitoring data and neural classifier helps to classify the attacks with high accuracy. They have used DARPA'98 dataset to evaluate their proposed work. In [11], the authors integrated genetic algorithm and K-Nearest neighbour (KNN) method to construct an IDS classifier. They have used KDD cup'99 dataset for the evaluation and k-fold validation to overcome the over fitting problem in the dataset. The authors of [12], proposed a hybrid technique using Support Vector Machine (SVM) and enhanced SVM algorithm. They have concentrated more on feature selectionby using self-organized map, genetic algorithm. They have evaluated their work using DARPA'99 IDS dataset.

From the literature survey, we have identified that flow-based mechanism is one of the commonly used technique in both traditional and SDN environment. The prediction accuracy of every work gets differ based on the datasets they have selected. For our work, we have selected NSL-KDD dataset which contains reasonable amount of data with no redundancy. Statistical-based techniques help to detect the attacks earlier but with high false alarm rate. Machine learning based techniques help to detect the attack with high accuracy. Algorithms like Random forest, Naïve Bayes, Support Vector Machine (SVM) took more time to classify the attacks in the large datasets [5]. Naïve Bayes and SVM failed to provide the better decision making in large datasets. After analysing these drawbacks, we have chosen C4.5 and K-Means for our classification model. But hybrid model would work well for classification in terms of accuracy and false alarm rate because the input can be processed in two stages like pre-processing and classification [13]. Feature selection plays major role in machine learning based models. So, we have used flow based information to identify the essential features and nature of the traffic. Also we decided to use ML based techniques to classify the attacks based on the promising results of using ML techniques in the existing works.

B. Background:

1. Flow Handling Mechanism in SDN

In SDN architecture, controllers and data plane devices are the two important elements. OpenFlow is straightaway the famous design of data plane devices. An OpenFlow device may contain one or more flow tables. Commercially available switches have Ternary Content-Addressable Memory (TCAM), can hold approximately 8000 flow entries. High performance switches (e.g., EXchip NP-4) can hold 125,000 to 1,000,000 flow entries. Every entry of the flow has three parts such as matching rule, counters to keep track of matching packet statistics and actions to be taken on packet_match.

Whenever a new packet reaches the data plane device, table lookup process starts. If a rule is already available, data plane device performs an action without forwarding to the controller. If table_miss (When no rule is available for that packet) occurs, the switch forwards the packet_in message to the controller. Then the controller responds with packet_out message by processing packet headers to generate a new flow rule. Then the switch can perform same action for consecutive flows without forwarding to the controller. The flow will be removed from the data plane devices, if idle timeout occurs or no packet is matched. After that, flow information, collected during the life time of that particular flow will be sent to controller in the form of flow_removed message. In addition to that, for every second, controller will request the Open vSwitches(OVS) for flow statistics information. Upon receiving the request, OVS witches will respond with flow_stats message which contains flow entry information of the flow table. By using this statistics collection process, the proposed methodology extracts the flow features and used them for classification purpose. Since flow statistics information collection is an existing part of the SDN framework, there will not be any performance degradation.

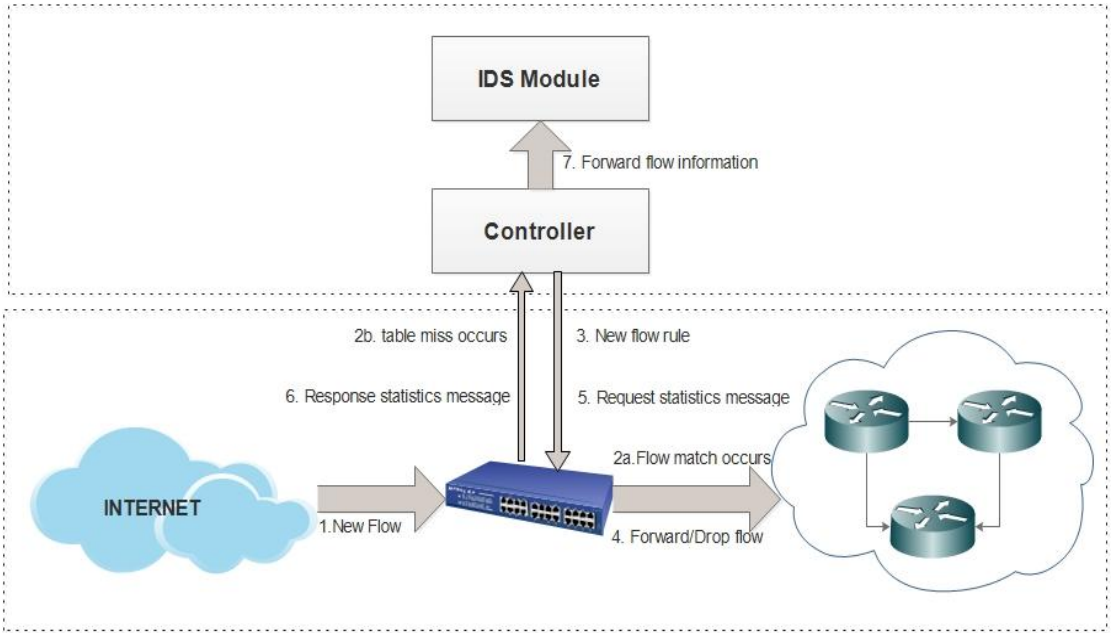

Fig. 1. Flow Based IDS

2. Why Hybrid Machine Learning?

Machine learning based techniques are commonly used in design of conventional IDS and substantially successful in both wired and wireless networks. In the same way, many researchers have implemented machine learning techniques in SDN-oriented environment also. These techniques are adopted to detect the malicious activities by analysing the various features in the network flows.

The ultimate goal of using this technique is to detect the attacks with high accuracy and low false alarm rate. This goal directs to the development of hybrid machine learning approaches. The concept of hybrid classifier is to integrate the several ML techniques to improve the performance of the classifier. A hybrid approach comprises of two functional phases. In the first phase, raw data is taken as input and produces the intervening results and this result is taken as input for second phase and generates the final results. In this context, clustering techniques are generally applied to raw input data to exclude the irregular data from each class. Then the results of these clustering techniques are taken as training examples to design the classifier. A hybrid machine learning approach is a combination of two machine learning techniques in that, first one focus on tuning the parameters in order to improve the performance in the second phase (i.e. Classifier).

III. PROPOSED WORK

The main goal of the proposed work is to detect the malicious activities in the SDN environment with high accuracy. Initially, the flow information is collected from OVS switches at regular intervals and by using that information essential features are extracted. After that by applying hybrid machine learning technique, we construct classifier module to detect attacks in the flow. In our proposed work, we have implemented K-Means clustering, Modified K-Means clustering, C4.5 decision tree and Modified K-Means+C4.5 (MKMC4) decision tree hybrid algorithm.

The IDS module consists of flow statistics collection module, traffic classification module, feature extraction module and hybrid machine learning testing and training phase to detect the attacks. From the controller, flow statistics are collected for every second. If a flow is inactive for more than two seconds, it is considered as idle. The message type indicates the reason for arrival of packets towards the controller. It may be due to table miss or flow rule installed in the flow table directing the packets towards the controller.

When a packet arrives towards the controller, feature extraction and traffic classification could happen by analysing header fields from the packet. For TCP and UDP traffic, source and destination IP, source and destination port, protocol type will have same values. Same is applicable for ICMP traffic also but with different port numbers. In addition to that, this module will eliminate the symmetric flow. If source IP address and source port number of one flow are similar to destination port number and IP address of another flow for TCP or UDP traffic respectively, then these flows are considered as symmetric flow. For ICMP symmetric flows, the two flows are request and response types. The main reason for eliminating symmetric flows is that attackers mainly spoof their IP addresses in order to restrict the responses from victims. So, this module installs the flow rules only for normal traffic and

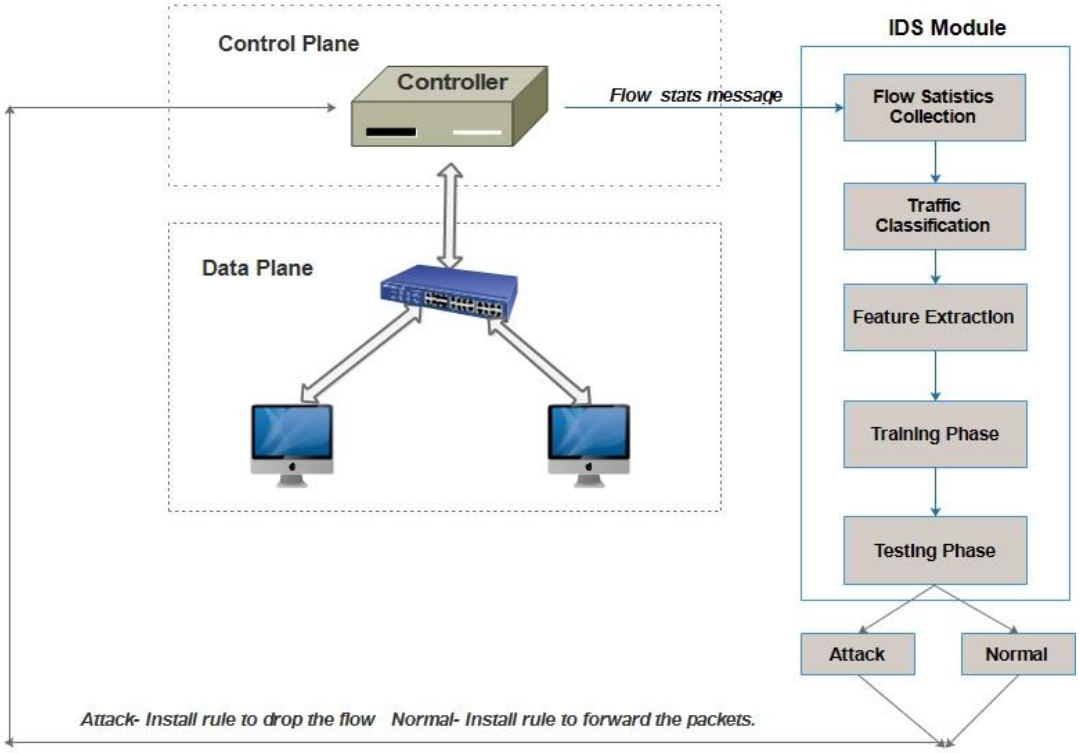

Fig. 2. System architecture diagram for proposed work

TABLE I: FEATURE SET AND ITS DESCRIPTION

Feature Name	Description
Protocol_type	Indicates nature of the protocol (TCP,ICMP,UDP,etc.)
Duration	Length of the connection (Number of seconds)
source_bytes	Number of bytes from source to destination
destination_bytes	Number of bytes from destination to source
Count	For the past two seconds, number of connections to the same host.
service_count	For the past two seconds, number of connections to the same service.

avoids the saturation in flow tables. For our proposed work, we have extracted six essential features such as protocol_type, duration, sorce_bytes, destination_bytes, count, service_count. The description for those features is given in the table 1. Then the machine learning based detection module will process the packets and classify it as normal or attack packets. Once the attack is detected, the OpenFlow protocol modifies the flow table immediately to drop the particular flow.

A. K-Means Algorithm:

Clustering techniques have been used as one of the famous machine learning technique. The main objective of the clustering is to form clusters based on the maximum uniformity among the data members into one cluster and minimum uniformity into separate cluster. In this context, K-Means clustering has been increasingly applied technique due to its simplicity, low complexity and possibly high in clustering large datasets.

The main objective of K- means clustering [14] in the implementation of IDS is to split and group the data into normal and attack instances. It partitions the input dataset into k-disjoint clusters based on their feature values, where k is the predefined positive parameter. Then the centroid value for each cluster will be computed by taking mean value of mathematical data. Each input data item will be committed towards the closest centroid by computing squared distance between centroids and the input data points. Then for each cluster, new centroid value will be computed by taking the mean value of input data items designated to each cluster. The steps involved in K-Means algorithm are:

1. Specify the K value (i.e) number of clusters.
2. Define the K cluster centroids. This can be defined by splitting all the input data items into K clusters, calculating their centroids and check whether all the centroids are unique.
3. For each data item, calculate the distance to centroid of all clusters. Reassigneach data item to the cluster with closest centroid value.
4. Update the centroid values for modified clusters.
5. Repeat step 3 until there is no change in centroid value.

To calculate the distance between two input data items, Euclidean distance function is used. It is defined as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$

But K-Means clustering technique has two drawbacks such as (i) initial selection of k and centroids, because different value of k results in different clusters.

It is really hard to obtain the optimal value of k, if the dataset distribution is unknown. Clustering outcomes mainly depends on selection of centroids only. So, k and centroid value plays a major role in clustering outcomes. (ii) The second drawback is that clustering could result in some empty clusters (i.e. degeneracy which are insignificant for classification. To overcome these problems ofK-Means technique, we have proposed Modified K-Means algorithm.

B. Modified K-Means Algorithm

In modified K-Means algorithm, number of clusters (k) is self-defined value instead of pre-defined constant value. Incorrect k value directs to poor clustering. The optimal solution is to try with different number of k values. But it is a complex process in large datasets. To overcome this problem, we have to choose the semi-optimal k value, based on the statistical properties of input data item. To calculate the correlation between two data items, our algorithm uses the Euclidean distance function (equation 1). The data should be normalized before calculating the Euclidean distance to avoid domination between the attributes. The steps involved in Modified K-Means algorithm are:

1. Specify the k value for clusters from the set {2,3,4,5…n}
2. Form the clusters based on the value of k.
3. Form the new clusters by removing the outliers from actual clusters.
4. Outliers, removed from actual clusters are assigned as new centroid, which may attract some data items from adjacent clusters to create a new cluster.
5. Some adjacent clusters may be linked to form a large cluster. The resulting clusters are multi-centred.
6. At the end of iteration, empty clusters will be simply removed.

The main aim is to remove the outliers from every cluster. Outlier is a data item that is away from maximum number of data in a particular cluster. The problem of degeneracy is eliminated by removing the empty clusters at end of iteration. The value of k in Modified K-Means algorithm is self-defined based on the data items and its properties.

C. C4.5 Algorithm

Decision tree algorithm [15] is one of the mostly used classification technique that repeatedly splits the input dataset into smaller subsets. To construct the decision tree, we have chosen C4.5 technique. The main reason for choosing C4.5 technique is thatit will overcome the over-fitting problem by applying entropy-based gain ratio as a splitting factor[16]. The construction of C4.5 decision tree is summarized as following steps:

1. Compute Information gain for each attribute.
2. Choose the attribute with highest gain ratio value.
3. Construct a branch for each possible value of selected attribute.
4. Split the input data items into subsets based on the selected attribute.
5. Repeat the steps 2, 3, 4 for every branch in the decision tree.

Entropy E can be computed by equation (2)

$$E = -\sum_{i=1}^{n} p_i \log_2 p_i \qquad (2)$$

where p_i shows the probability of class i

Information Gain (IG) for each attribute, '*a*' can be calculated by equation (3)

$$\text{IG(E,A)} = \text{Entropy(E)} - \sum_{a \in Values(A)} \frac{|E_a|}{|a|} * \text{Entropy}(E_a) \qquad (3)$$

where E represents entropy for overall input data items, A represents set of all instances for attribute A, E_a represents subset of E for which attribute A has value *a* and $|E_a|$ represents total number values in attribute 'a'.

D. Modified K-Means Plus C4.5 Technique (MKMC4)

We have taken the input data set (x_i,y_i), i=1,2,3..n, where x_i shows n-dimensional vector values and $y_i=\{0,1\}$ shows the label with "1" for attack and "0" for normal.The Modified K-Means+C4.5 technique has two phases: 1) training phase and 2) testing phase. In the training, steps 1-6 from Modified K-Means algorithm are applied to split the input dataset into k disjoint clusters. Then, by applying C4.5 decision tree, we train the data items in each cluster. Modified K-Means algorithm assures that every data item in the training set is available in only one cluster.

Algorithm I :Modified K-Means + C4.5 Algorithm

Cluster selection phase
Input: *Test dataset W_i , i=1,2,3…n.*
Output: *Closest cluster to test dataset W_i*
Start
For each i in W,
Calculate the Euclidean distance D (by applying equation 1)
Identify the closest cluster to W_i
Construct the C4.5 decision tree for the identified closest cluster.
End
Classification phase
Input: *Test dataset W_i , i=1,2…n.*
Output: *Classify data as attack or normal*
Start
Apply the test dataset W_i over the identified closest cluster by C4.5 decision tree.
Classify the data item as attack or normal.
Update the centroid of the cluster.
End

In the testing, we have two components: 1) cluster selection phase 2) classification phase. In the cluster selection phase, for every input data item, we calculate the Euclidean distance using equation (1) and identify the closest cluster. Then, we construct the decision tree for the identified cluster. In the classification phase, we apply the test data items over the identified closest cluster by C4.5 decision tree. Finally, we classify every data item in the test dataset as attack or normal. The pseudo code for this proposed hybrid algorithm is given below as Algorithm I.

IV. EXPERIMENTAL SETUP

For our experiment, we have used a network

emulator tool called Mininet [17].

It is a standard tool used to create SDN-oriented network environment. It can create a network of virtual controller, host, switches and links. For controller, we have chosen POX[18],a simple, fast and a python-based controller. We have used OpenFlow, a software based switch can handle the large amount of traffic loads. By using Mininet, we have created a network topology with single controller, three switches and 42 hosts. By using Scapy[19] tool, we have generated various kinds of traffic for both normal and attack.

A. Dataset

We have used NSL-KDD [20] dataset as a sample training set to train our machine learning module. The NSL-KDD dataset is improved version of KDD cup 1999 dataset which overcomes the redundant data problem in KDD'99 dataset. This dataset contains 125,973 data instances of both normal and various kinds of attack. Each data instance in the dataset has forty one features. All the attacks in the dataset come under any one of the four categories such as DoS (Denial of Service). U2R (User to Root), R2L(Remote to User), Probe. In DoS attack, an attacker restricts the normal users from accessing the service. Example: worm, apache2, Neptune, land back etc. In U2R attacks, an attacker tries to act as a root user by having local access to the victim machine. Example: xterm, rootkit, buffer-overflow, perl, ps etc. In R2L attacks, without having any account in a system, an attacker tries to gain access as a normal user. Example: xlock, sendmail, fpt-write, imap, phf etc. In Probe attacks, an attacker continuously monitors the targeted host to gain the information. Example: portsweep, saint, nmap, satan, ipsweep, etc.

TABLE II: NSL-KDD TRAINING DATASET INSTANCES

Class	Number of records	% of records
DoS	45927	36.46%
U2R	52	0.04%
R2L	995	0.79%
Probe	11656	9.25%
Normal	67343	53.46%
Total	125973	100%

TABLE III : NO OF RECORDS GENERATED USING SCAPY TOOL

Class	No. of records	Total Records
Normal	70656	99310
Attack (DoS, TCP, UDP, ICMP, SSh Brute force etc..)	28654	

For our work, we have chosen six features from the 41 features available in the NSL-KDD dataset. The main reason for selecting these six features is that in SDN environment also, we can extract these features from the network flow. Table II indicates the training dataset instances to train our model. Table III indicates the attack and normal instances generated with the help of Scapy tool.

B. Performance Evaluation

To evaluate our proposed work, we have used four performance metrics [21] such as Accuracy (A), Recall (R), Precision (P) and F-measure (F). Better IDS can detect the attacks with high accuracy and low false alarm rate. To calculate precision, recall and f-measure we have used n*nconfusion matrix, where n defines the number of classes. It is a combination of four different actual and predicted classes such as True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN). TP represents number of normal instances which are correctly classified as normal. TN represents number of attack instances correctly classified as attack. FP represents number of attack records classified as normal records. FN represents number of normal records classified as attack records. Accuracy (A) is defined as number of correctly classified records by total number of records. Precision (P) is defined as number of correctly classified instances by sum of TP and FP. Recall (R) is defined as number of correctly classified instances by sum of TP and FN. Equation 4,5,6 indicates calculation part of accuracy, precision and recall. The F-measure is mainly calculated to identify the balance between precision and recall in case of uneven class distribution. It is calculated by using equation 7.

$$\text{Accuracy (A)} = \frac{TP+TN}{TP+TN+FP+FN} * 100 \quad (4)$$

$$\text{Precision (P)} = \frac{TP}{TP+FP} * 100 \quad (5)$$

$$\text{Recall (R)} = \frac{TP}{TP+FN} * 100 \quad (6)$$

$$\text{F-measure} = 2*\frac{Precision*Recall}{Precision+recall} * 100 \quad (7)$$

To evaluate our proposed work, we have compared K-Means clustering, C4.5 decision tree, Modified K-Means, K-Means clustering + C4.5 and Modified K-Means clustering + C4.5 algorithm. Our experiment results show that hybrid machine learning algorithm provides better accuracy compared to single machine learning algorithm. To train our model, we have used 125,973 data instances from NSL-KDD dataset. To test our model, we have generated 99310 data instances of both attack and normal data by using Scapy tool. From the results, we have proved that the hybrid model works better compared to the single machine learning model. In Table IV, we have given the performance comparison of our proposed work with single machine learning algorithms. Our proposed model,Modified K-Means+C4.5 produces high accuracy of 97.66%. In Figure 3, we have shown the confusion matrix for two hybrid machine learning based classifiers. The Receiver Operating Characteristic (ROC) curve is the mapping of True Positive Rate (TPR) against False Positive Rate (FPR). For the perfect classification model, TPR = 1 and FPR = 0.For the worst model, TPR = 0 and FPR = 1. Average model is represented as a diagonal in the graph. All the points above the diagonal indicate the better classification results.

TABLE IV: PERFORMANCE EVALUATION OF PROPOSED WORK

	Performance Metrics in %			
Classifier	**Accuracy**	**Precision**	**Recall**	**F-measure**
K- Means	91.94	95.63	92.92	94.26
C4.5	93.96	97.09	94.34	95.69
Modified K-Means	93.29	96.33	94.15	95.23
K- Means + C4.5	96.73	96.97	98.41	97.68
MKMC4	**97.66**	**97.66**	**99.04**	**98.34**

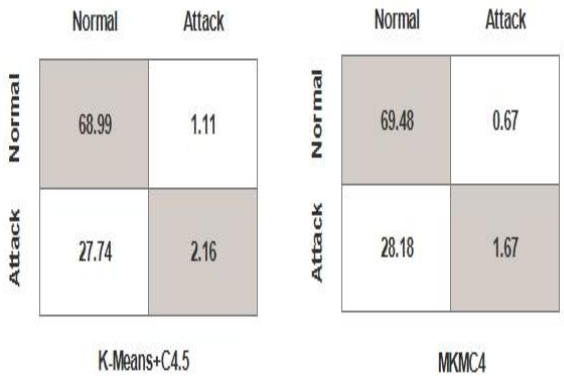

Fig. 3. Confusion Matrix

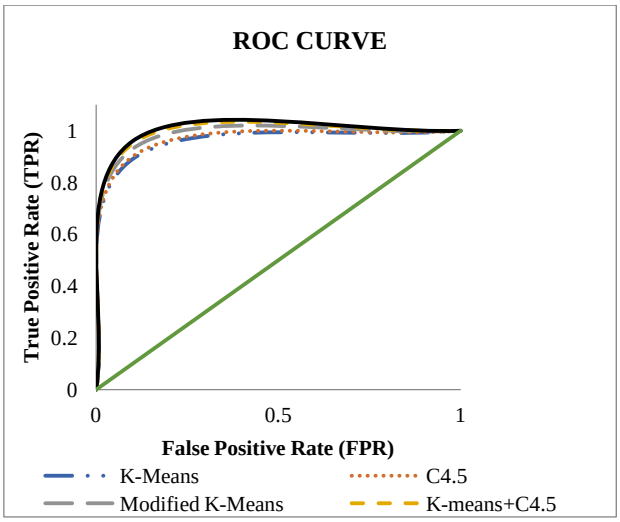

Fig. 4. ROC Curve

ROC curve for our proposed work shows the better results with all the points above the diagonal line which is shown in figure 4. Figure 5 shows the comparison of accuracy for our proposed work with single ML algorithms.

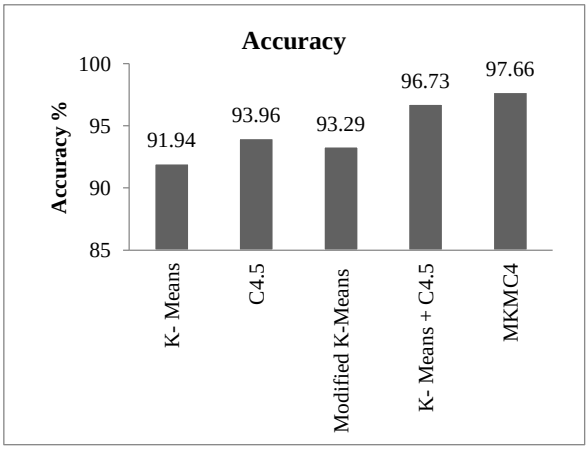

Fig. 5. Comparison of Accuracy.

V. CONCLUSION AND FUTURE WORK

In this paper, we have used a hybrid machine learning technique to construct efficient IDS. Our results shown that hybrid machine learning technique works well for classification of attacks in terms of accuracy and false alarm rate compared to the single machine learning technique. By periodically collecting the flow statistics information from the controller, we have implemented our proposed work. Since it is already part of the controller mechanism, there will not be any overload complexity in the SDN environment. From the collected flow statistics, we have performed the traffic classification in terms of protocol such as TCP, UDP, ICMP and feature extraction process. Then it is passed through our classifier model to detect the intrusion in SDN environment. By identifying drawbacks in K-Means clustering, we have developed Modified K-Means algorithm. By Combining C4.5 and Modified K-Means algorithm, our classifier achieved accuracy of 97.66%. In the near future, we aim to implement the detection mechanism in data plane switches for early detection of attacks and also to detect the insider attacks in the host system.

REFERENCES

[1]. Kreutz, D., Ramos, F. M., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, *103*(1), 14-76.

[2]. Specification, O. S. (2015). Version 1.5. 0. *Open Networking Foundation*.

[3]. Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1), 16-24.

[4]. Nguyen, T. T., & Armitage, G. J. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys and Tutorials*, *10*(1-4), 56-76.

[5]. Rathore, M. M., Ahmad, A., & Paul, A. (2016). Real time intrusion detection system for ultra-high-speed big data environments. *The Journal of Supercomputing*, *72*(9), 3489-3510.

[6]. Mehdi, S. A., Khalid, J., &Khayam, S. A. (2011, September). Revisiting traffic anomaly detection using software defined networking. In *International workshop on recent advances in intrusion detection* (pp. 161-180). Springer, Berlin, Heidelberg.

[7]. Wang, R., Jia, Z., & Ju, L. (2015, August). An entropy-based distributed DDoS detection mechanism in software-defined networking. In *2015 IEEE Trustcom/BigDataSE/ISPA* (Vol. 1, pp. 310-317). IEEE.

[8]. Niyaz, Q., Sun, W., &Javaid, A. Y. (2016). A deep learning based DDoS detection system in software-defined networking (SDN). *arXiv preprint arXiv:1611.07400*.

[9]. Shalini, S., &Vetriselvi, V. (2018). Intrusion Detection System for Software-Defined Networks Using Fuzzy System. In *Proceedings of the International Conference on Computing and Communication Systems* (pp. 603-620). Springer, Singapore.

[10]. Song, S., Ling, L., &Manikopoulo, C. N. (2006, December). Flow-based statistical aggregation schemes for network anomaly detection. In *2006 IEEE International Conference on Networking, Sensing and Control* (pp. 786-791). IEEE.

[11]. Canbay, Y., &Sagiroglu, S. (2015, December). A hybrid method for intrusion detection. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (pp. 156-161). IEEE.

[12]. Shon, T., & Moon, J. (2007). A hybrid machine learning approach to network anomaly detection. *Information Sciences*, *177*(18), 3799-3821.

[13]. Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *expert systems with applications*, *36*(10), 11994-12000.

[14]. Münz, G., Li, S., & Carle, G. (2007, September). Traffic anomaly detection using K-Means clustering. In *GI/ITG Workshop MMBnet* (pp. 13-14).

[15]. Quinlan, J. R. (1996). Learning decision tree classifiers. *ACM Computing Surveys (CSUR)*, *28*(1), 71-72. 19

[16]. Quinlan, J. R. (1996, August). Bagging, boosting, and C4. 5. In *AAAI/IAAI, Vol. 1* (pp. 725-730).

[17]. http://mininet.org/overview/

[18]. Kaur, S., Singh, J., &Ghumman, N. S. (2014, February). Network programmability using POX controller. In *ICCCS International Conference on*

Communication, Computing & Systems, IEEE (Vol. 138).

[19]. http://scapy.readthedocs.io/en/latest/

[20]. http://www.unb.ca/cic/datasets/nsl.html

[21]. Zhu, W., Zeng, N., & Wang, N. (2010). Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, *19*, 67.

AUTHORS PROFILE

Mr. K. Muthamil Sudar is working as an Assistant Professor in the Department of Computer Science and Engineering, School of Computing, Kalasalingam Academy of Research and Education, Krishnankoil, Tamilnadu. He is pursuing his PhD in the area of Security in Software Defined Networks.

Dr.P.Deepalakshmi is currently working as a Professor in Department of Computer Science and Engineering at Kalasalingam Academy of Research and Education. She is also serving as Dean, School of Computing. Her research interest includes Optimization Techniques, Network Routing, Distributed Computing and Network Security. She has published her research in many journals and conferences and also recognized as an eminent speaker in the field of Computer Science in the nearly region. She also takes care of KARE ACM student chapter as faculty mentor. Contact her at deepa.kumar@klu.ac.in