Real-time DDoS attack detection in Software Defined Networks using Machine Learning techniques

P. Visu
Associate Professor,
Department of Computer Science and Engineering,
Velammal Engineering College,
Chennai, Tamil Nadu, India.
E-Mail: pandu.visu@gmail.com

Neeba E. A
Assistant Professor,
Department of Information Technology,
Rajagiri School of Engineering & Technology,
Rajagiri Valley, Kakkanad, Kochi – 682039, Kerala.
E-Mail: neebarset@gmail.com

Koteeswaran. S
Associate Professor,
Department of Computer Science and Engineering, School of Computing,
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology,
Chennai-600062, Tamil Nadu, India.
E-Mail: s.koteeswaran@gmail.com

Abstract:

Software Defined Network (SDN) is a recently discovered network architecture, that provides decouples the control plan and data plan and it allows flexible and efficient management of the network by using software program. SDN control the entire network. The failure point in the new network is if Distributed denial of service attack makes the controller become unreachable. In order to overcome DDoS in Software Defined Networking (SDN), in this paper, we have proposed a novel hybrid algorithm for classification based on Decision tree (DT) and Support Vector Machines (SVM) Algorithms. The proposed mechanism is evaluated on SDN test bed, Our experimental result shows that our mechanism can effectively address the attack detection system using real-world network traffic. Our algorithm achieved high detection rate (DR) and reduces false positive (FP) for different types of network intrusions.

Keywords*Software-defined Networking, Distributed Denial of Service, Decision tree, Support Vector Machines, Feature extraction, Intrusion Detection System (IDS).*

1. INTRODUCTION

The traditional approaches, configuration and optimization methods may not be sufficient to deal with increasing network size, heterogeneity, and complexity of the current and future computer network [1-3]. The legacy networks don't offer programmability, flexibility in the networking functions. It does not provide any support to implement and test a new innovative idea without interrupting the ongoing services[3-5]. Traditional Network uses devoted appliances; most of the operation is implemented in devoted hardware. The traditional network has the following limitation.

- Many steps are needed to add or remove a single device from a traditional network. First, we have to manually configure all devices, then using a device-level management tools to update various configuration settings [6].
- Many organizations own networking devices from different vendors, for successful implementation the administrator must have knowledge of all present networking devices [7-8].
- Traditional network is too expensive [6-9]
- The applications running at layer seven (application layer in the OSI model) have discernability into resources available at the network layer [10].
- Troubleshooting the failures in networking devices is a tedious process.

Traditional network operation cannot be reprogrammed because of the underlying hardwired implementation of the routing rules. A new protocol development, design, and standardization take more years to develop. These observations caused the novel approach for future networks to support innovative ideas. In order to provide a better design for future networks, some new ideas like combining cloud with software-defined networking are introduced. SDN decouples the control plan and data plan and it allows flexible and efficient management of the network by using software program.

The controller creates the rule in the data plan and the device such as switches, routers in the data plan forwards the packet based on the rules created by the controller. These controllers provide an efficient platform to deploy new network applications and specific purposes such as firewall and intrusion detection system.

The control plan can send the command down to the switch operating at the hardware layer. This helps to make changes globally instead of device configurations. The switches in the

data plan forward the packet according to the flow rules created by the control plane that is controlled by the controller such as floodlight, NOX, POX, Open-Daylight, etc, hence the switches tasks are simplified as the need not do the control functions.

The data plan can be programmed by using an open interface such as OpenFlow. OpenFlow is a protocol that enables the implementation of SDN in both hardware and software. There are several advantages of SDN including intelligence, speed, easy network management, and multi-tenancy architecture. The SDN architecture has three main components including a controller, switches, and flow entries in the switches. The northbound API is given by the controller and the application need to deal with their correspondence to the controller through it. This API includes Procera, REST(Representational State Transfer) API, Java, and Python are some of the commonly used API's [11].

The southbound API is the correspondence between the controller and the system gadget. OpenFlow is the generally utilized convention for such correspondence. The switch that helps OpenFlow is comprised of three basic segments: Group table, stream table, and OpenFlow secure channel. The gathering table and stream table characterizes switches conduct for the various information streams originating from the physical or virtual interface. These tables contain a set of guidelines likewise called as streams in which stream of the information is characterized. The switch pursues the activity related to the switches. OpenFlow channel gives an interface to the controller so as to make new streams and oversee existing streams.

Each flow is consists of three fields: Rule, Action, and Statistics. The rule defines the header fields to match with the Ethernet frame flows. Once the rule is to match the traffic, the action associated with the rule should be performed. The actions can be forwarded to one or more ports, forward to the controller, drop the frame, and modify frame fields. Whenever the rule matches with incoming traffic the counters in the flow table have to be updated. These counters are maintained for flow entry, port, group, queue, and meter. Statistics maintained in the switch can be collected by the controller.

2. RELATED WORK

In the paper [11] they propose a crossover component, to be specific SDNScore, where switches are not just information forwarders. Rather, they can gather measurements and choose if DDoS assault is in real life. At that point, they organize with the controller and follow up on assault parcels in collaboration. SDNScore is a measurable and parcel based guard instrument

against DDoS assaults in SDN condition. Since it has a measurable scoring strategy, it can recognize referred to as well as obscure assaults involving parcels that are indistinguishable as far as TCP and IP layer properties. Likewise, it doesn't drop all bundles in a stream which incorporates both assault and lawful parcels, yet rather sift through assault bundles utilizing bundle based analysis.SDN switches and controller are furnished with some knowledge installed in proposed modules to empower this system. It involves both recognition and moderation capacities. SDNScore is successful against obscure assaults which utilize practically equivalent to parcels since it uses factual examination and makes a correlation with ostensible traffic. It decides most fitting qualities for current traffic and gives an extensive improvement in precision.

In this paper [12] they initially anticipate the measure of new demands for each Open Flow switch occasionally dependent on Taylor arrangement, and the solicitations will at that point be coordinated to the security entryway if the expectation esteem is past the edge. The solicitations that caused the sensational decline of entropy will be sifted through and principles will be made in security door by our calculation; the standards of these solicitations will be sent to the controller. The controller will send the principles to each change to make them direct the streams coordinating with the guidelines to the nectar pot. The recreation demonstrates the midpoints of both false positive and false negative are under 2%.When the security portal gets these parcels data, it will initially ascertain the sorts of entropy of them, and make out the guidelines dependent on the difference in entropy esteem. In the event that there is a sharp change among them, the security portal will announce an assault, and make rules for them. The rules are made based on the value that makes the entropy decrease dramatically. Simulation shows that our DDoS scheme is able to detect and defense against the DDoS attack well.

In this paper [13], the DDoS attack is detected based on the time duration of the attack. Here the authors also utilized the time attack pattern to avoid the attack in the future. The destination address is used for detecting the DDoS attack. A flow collector module is also established so as to collect the invalid packets (the packets having an invalid destination address). If the invalid packet at the flow collector is increased drastically, it will issue a notification to the controller. Then the controller will make a new rule to directly pass the invalid packets to the flow collector. The traffic pattern is noted to prevent future DDoS attack of the same pattern. The threshold value is fixed based on the number of invalid packets (R). If the

value of R is greater than the threshold value, then the flow collector will send the DDoS notification to the controller to create a new rule.

In paper [14], SDN contains queue for holding the packet requests. DDOS attack occurrence in SDN may high if a single queue is used since a single queue cannot hold off the spurious flow requests from attackers and the legitimate user's requests cannot access the queue. Hence 'MultiQ' which means multiple queues are employed in SDN, i.e., a queue for each switch in SDN and controller serves these queues. In this paper, 'MultiQ' in round-robin scheduling is used and is compared with 'Static' and 'SingleQ' schemes. In static the performance limiting factor [μ/k, where μ is the processing capacity and k is the number of switches] decreases as the number of switches increases whereas this would not be a problem in MultiQ and also the number of connections established even during the attack [N_s] is greater and so channel utilization by the legitimate users is also more in MultiQ than when compared to static scheme. The setup failure rate, R_f can be calculated from a Markovian process and R_f in SingleQ is more than in MultiQ. The performances of SDN with Static, single and MultiQ are analyzed by using an emulation tool.

In the paper, [15] Peer bolster system, was proposed under which, the switches treat each other as companions. At the point when a companion change comes up short on stream table assets, other friend switches bolster the focused on the switch by offering their inactive stream tablespaces to introduce the new guidelines (counting both assaulting principles and authentic standards), to moderate the assault. In this way, relief of assault is done, yet in addition, sending the greater part of the real traffic is additionally done effectively, limiting the QoS infringement enduring an onslaught. The traffic is designed to different changes by appropriating to the whole system, if the stream table is full, on observing the switch status. The guidelines to process the traffic must be introduced in the companion switches for diverting the traffic to friend switches. Additionally, the companion switches ought to be favored which are not full, closest to the over-burden switch and less occupied to redirect the traffic.

From the literature, it is inferred that the DDoS attack has been very frequent for the past three decades and after the introduction of SDN system, it is a very big issue. This is because all the networking devices in an area are connected to a single controller which means any attack on the controller can paralyze the entire network of that particular area. Hence to address the above problem, a novel DDoS detection system is proposed in this paper.

3. PROPOSED WORK

The proposed system is used to control the SDN to keep the network secure. When the switch receives the packet, it parses the packet and extracts the header fields and look for a matching entry in the flow table. If there are multiple matching flow rules, then based on priority the respective action in the rule gets executed. The Open Flow switch has counters for each flow and it is updated after executing the actions. Otherwise, if there is no matching entry found in the flow table then the switch sends packet arrival message to the controller by encapsulating the packet within the message. The proposed system is developed as an SDN application which runs on the application layer. The main objective of this application is to detect the DDoS attack before the first attack packet reaches the destination (victim). Also without any additional overhead, the control message sent from the switch to the controller can be utilized to collect the traffic. The classifier is used to detect the DDoS attack after capturing the incoming packet arrival messages. Once the attack is detected, it is mitigated with the help of features available in Open Flow protocol.

Figure 1 shows DDoS detection mechanism contain various stage i) Feature Extraction and Selection ii) Classification iii) Rule-based analyzer. Each and every packet arrival to the network will pass the various stages in DDoS detection engine for detection of DDoS attack.

a) Packet Flow Monitor

In Packet flow monitor which monitor the arrival packet and get the information from the arrival packet. Packet flow monitor which consist of two submodules 1. Traffic Analyzer 2. Feature Extraction and Selection. This module gets the flow information from the open flow enabled switches using an open flow control message. Both passive listening and active querying can be done to get the network information packet arrival FLOWMOD flow removed are the control message used to monitor traffic using passive listening each switch sends a bundle entry message to the controller each time it gets an information parcel that does not coordinate any of its stream table passages. The controller answers with flow mod message to introduce the new stream rule for the bundle and the resulting parcel in a similar stream rule. Both of these messages include some level of flow information (e.g., source IP and port destination IP and port), routing configuration (e.g., VLAN tag, ingress interface egress interface), and the flow start time the packet arrival message is encapsulated

with the actual packet sent from the hosts in SDN environment. The packet header fields are extracted to get the traffic features.

b) Flow Classifier

The proposed classifier used to identify the new incoming packet arrival message from the switch to the controller is triggered by the normal or abnormal flow. The field extracted from the message are formed as individual attributed and then these attributes are grouped together as a single instance. These instances are given as training example to the classifier. The proposed classifier is trained with the sample data and formed as a model. Once the new packet arrival message received, the packet fields are passed to the proposed classifier and it identifies the new instance as normal or attacks. When the classifier returns the result as normal, the packet arrival message is sent to the controller and continue the process. Else the packet is attacked it should not pass to the controller. It moves to detector action.

c) Rule-based priority analyzer

In priority, analyzer gets an input from the flow classifier while in the classifier it classifies the flow into a normal flow or abnormal flow. If the flow is abnormal it forwarded to priority analyzer. In priority analyzer, abnormal flow in check based on the rule. Install high priority flow rule for the attack traffic so that benign traffic rules are subsided. Drop the attack traffic and updated in training data.

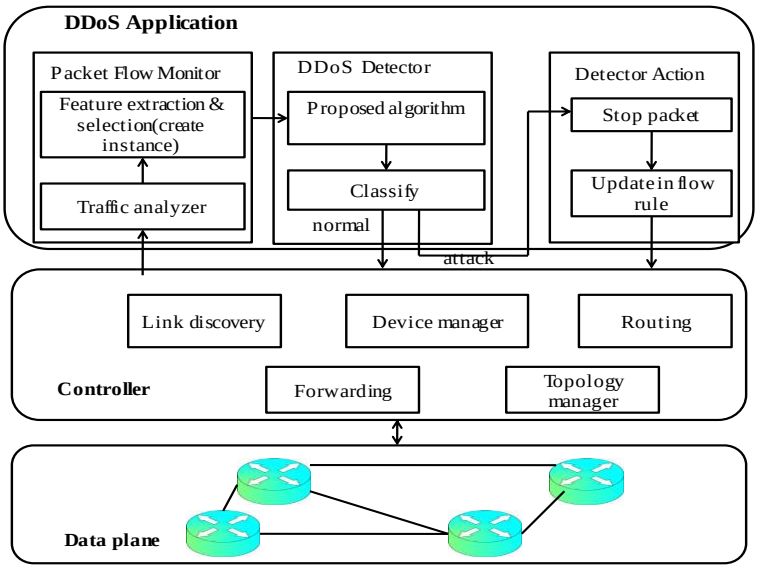

Figure. 1 System Architecture

Decision tree Algorithm

Decision Tree is a classification method which combines multiple tree predictors. Each tree depends on the value of randomly chosen vector which is distributed among all trees. The main key advantage of the decision tree is it can measure the importance score of each feature. DT Algorithm can handle high dimensional data and use a large number of trees in the ensemble. Hence the random selection of variables will minimize the correlation between the trees in the ensembles. So the error rates are reduced and computationally much lighter compared with other classifiers.

Support Vector Machine

Among the machine learning methods, SVM has maximum generalization capability compare to other machine learning algorithms. Generalization refers to the ability of the training machine classify the unknown test sample accurately with the experience attained from the training dataset. This generalization helps to identify the correct class for the attack instances. SVM always find a global optimum solution rather than stopping with local optimum. It also reduces the false positive rate and improves accuracy. This false positive rate helps to reduce the analysis of normal packet flows that have been classified as attack and focus on actual attack instances. SVM helps to trace back the malicious traffic. This helps to identify the source of the attack and block it. Also, it increases the possibility of prevention of DDoS. SVM has only a few parameters to be adjusted (cost and gamma parameters). SVM cannot give additional information about the detection type of attack. The complexity of this algorithm is very high.

Hybrid algorithms:

To overcome such issues, in this work we have proposed two independent hybrid algorithms respectively for decision tree and SVM classifiers to improve the classification accuracy and reduce the False positive rate. The hybrid classifier tries to address the desirable properties of both SVM and DT based classifiers. The SVM trained model is loaded with the rule set file and the input parameters are created as instance format. The instance is a collection of attributes after the nominal to binary attribute conversion. SVM classifier works based on the hyperplane decision boundary framed from the training data.

Algorithms

Pseudo code for Decision tree Algorithm

Input: training data

Output: Decision tree

1. In tree create a node t.
2. Label t with the most common value of Target in D.
3. In D if all values are + return single node t, with the + label.
4. In D if all values are - return single node t, with the - label.
5. Return single node if the attribute is empty
6. Let A* be the attribute from Attributes that best classifies examples in D.
7. Assign t the decision attribute A*.
8. For each possible value "a" in A* do:
9. Add a new tree branch below t, corresponding to the test A* = "a".
10. Let D a be the subset of D that has value "a" for A*.
11. If D is empty: Then attach a leaf node with a label of the most common value of Target in D.
12. Else attach the node returned by generating a decision tree.
13. Return t.

Pseudo code for SVM Algorithm

Input: instance, gamma,cost
Output: traffic Classification
1 begin
2 initialize training points as(X_i, Y_i)
3 nominal_to_binary(categorical)
4 decomposed the problem into a binary problem
5 compute SVM solution for w and b
$w=\sum\alpha_i y_i x_i$
$B=y_k-w^t x_k$
6 compute the classification result
$F(x)=\sum\alpha_i y_i x^t_i x+b$
7 determine the support vector
8 use hyperplane to classify the test data
9 end

4. EXPERIMENTAL SETUP AND RESULT ANALYSIS

The application is developed using MININET emulator openVswitch and POX controller. POX is a Python-based SDN controller. It offers both virtual and physical open flow switches openVswitch function as software virtual switch in a virtualization environment. The network created in MININET is interfaced with POX controller. DDoS attack module is developed using python and also packet generation is done using a python script. The packet arrival message process is modified to implement the DDoS engine the packet arrival message includes header information and then the actual packet received from the host. The features identified from the packet arrival message passed to the flow classifier model, in that model we developed a novel hybrid classifier to classify the data. Hybrid classifier model was trained with normal and attack traffic patterns include NTP DDoS flooding, reverse TCP flooding. Based on the result the further action is taken in case of attack the flow rule has to be created with action set to drop or forward to the controller for further analysis. If the result is normal then the normal packet arrives flow can be continued. Figure 2 shows the link states between the host and switches whereas Figure 3 shows the screenshot of the attack launched over the particular host. Figure 4 shows the attack traffic detected using the proposed method.

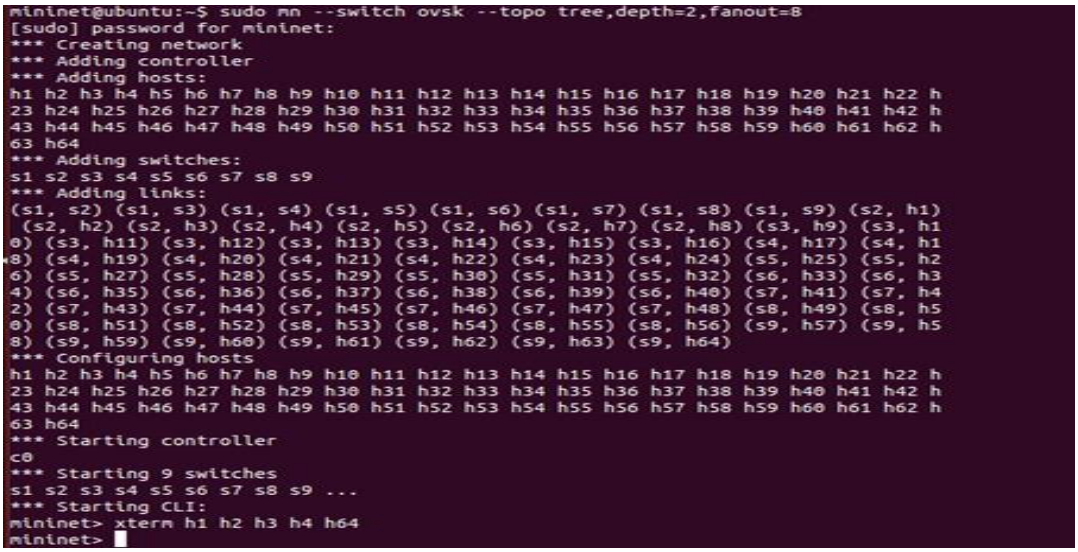

Figure.2 Link Generation Between Switches And Hosts

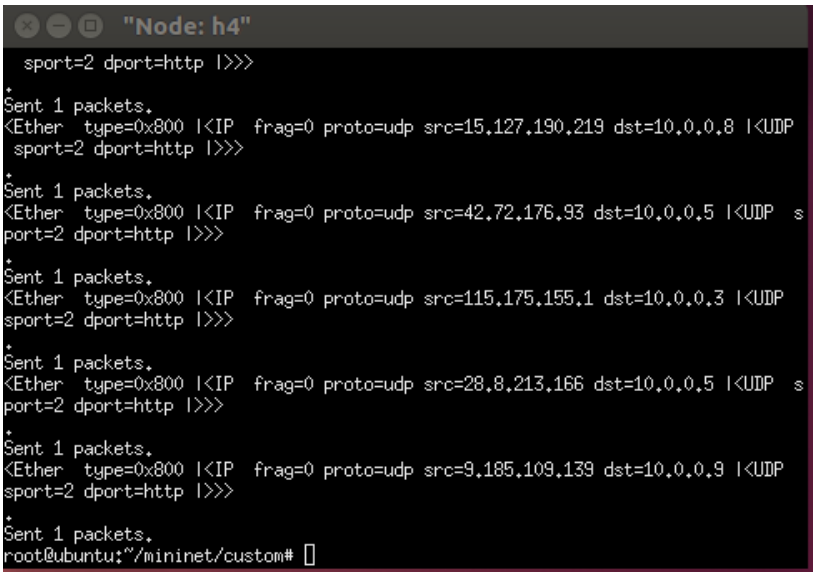

Figure.3 Launch Attack

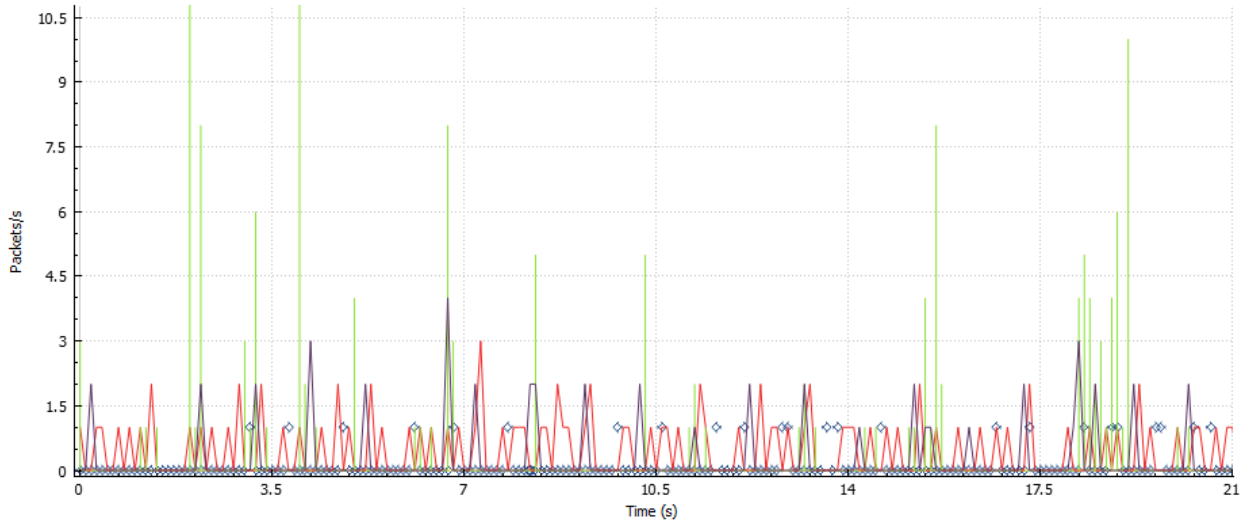

Figure.4 shows the graph for detecting DDoS attack using a new novel proposed classification algorithm and rule defined.

5. CONCLUSION

The design of this research is standard because traffic collection detection and mitigation are decoupled and it can be easily modified the proposed system detects the possibilities of attack with first control packet sent from switch to the controller the system performance is

evaluated in terms of bandwidth utilization flow rule creation time and packet forwarding delay the simulation result shows that the attack is detected effectively with false positive rate.

References

1. Chung, C. J., Khatkar, P., Xing, T., Lee, J., & Huang, D. (2013). NICE: Network intrusion detection and countermeasure selection in virtual network systems. *Dependable and Secure Computing, IEEE Transactions on*, *10*(4), 198-211.
2. Wang, B., Zheng, Y., Lou, W., &Hou, Y. T. (2015). DDoS attack protection in the era of cloud computing and Software-Defined Networking. Computer Networks, 81, 308-319
3. Yu, S., Tian, Y., Guo, S., & Wu, D. O. (2014). Can we beat DDoS attacks in clouds?. *Parallel and Distributed Systems, IEEE Transactions on*, *25*(9), 2245-2254.
4. Yau, D. K., Lui, J., Liang, F., & Yam, Y. (2005). Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles.*IEEE/ACM Transactions on Networking (TON)*, *13*(1), 29-42.
5. Netti, K., &Radhika, Y. (2015, December). A novel method for minimizing loss of accuracy in Naive Bayes classifier. In *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*(pp. 1-4). IEEE.
6. Schmidt, B., Kountanis, D., & Al-Fuqaha, A. (2014, December). An artificial immune system inspired algorithm for flow-based internet traffic classification. In *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on* (pp. 664-667). IEEE.
7. Benferhat, S., &Tabia, K. (2005, November). On the combination of naive Bayes and decision trees for intrusion detection. In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on* (Vol. 1, pp. 211-216). IEEE.
8. Ashraf, J., &Latif, S. (2014, November). Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques. In*Software Engineering Conference (NSEC), 2014 National* (pp. 55-60). IEEE.

9. Alfisahrin, S. D. N. N., &Mantoro, T. (2013, December). Data Mining Techniques for Optimization of Liver Disease Classification. In *Advanced Computer Science Applications and Technologies (ACSAT), 2013 International Conference on* (pp. 379-384). IEEE.
10. Kokila, R. T., ThamaraiSelvi, S., &Govindarajan, K. (2014, December). DDoS detection and analysis in SDN-based environment using support vector machine classifier. In *Advanced Computing (ICoAC), 2014 Sixth International Conference on* (pp. 205-210). IEEE.
11. Kiibra Kalkan, Giirkan Giir and Fatih Alagoz," SDNScore: A Statistical Defense Mechanism Against DDoS Attacks in SDN Environment", Proc. of IEEE Symposium on Computers and Communications (ISCC),2017, pp. 669-675.
12. Xueli Huang, Xiaojiang Du and Bin Song," An Effective DDoS Defense Scheme for SDN" proc. of IEEE International Conference on Communication,2017, pp. 1-6.
13. N. I Gde Dharma, M. Fiqri Muthohar, J. D. Alvin Prayuda, K. Priagung, Deokjai Choi, "Time-based DDoS Detection and Mitigation for SDN Controller", 17th Asia-Pacific Network Operations and Management Symposium(APNOMS), 2015, pp.550-553.
14. Sungheon Lim, Seungnam Yang, Younghwa Kim, Sunhee Yang, Hyogon Kim (2015)," Controller scheduling for continued SDN operation under DDoS attacks", IEEE, Volume-51, pp.1259-1261.
15. Bin Yuan, Deqing Zou, Shui Yu, Hai Jin, and Jinan Shen," Defending Against Flow Table Overloading Attack in Software Defined Networks", Proceedings of IEEE Transactions on Services Computing, 2017, volume: PP, Issues: 99.