

# A Defense Mechanism for Identifying DDoS Attack in SDN Based Cloud

A.Lavanya

lavanyasamymsc@gmail.com

Assistant Professor, KG College of Arts and Science, Coimbatore

**Abstract.** The advancements in Software defined Networking (SDN) techniques, which have significantly increased the number of cyber security threats, cloud applications have experienced an astounding transformation. The performance of most organizations' systems is severely decreased by the damaging cyberattack known as distributed denial of service (DDoS). By effectively identifying the traffic data in the SDN-based cloud, prominent research has found that machine learning models are the most efficient options for detecting assaults. This work seeks to offer a machine learning classifier model that balances accuracy and complexity based on these interpretations. Through the use of a brand-new Hyper-heuristic Butterfly Optimisation Algorithm (HHBOA), an enhanced Support Vector Machine (SVM) classifier has been developed. Source-based IP filtering (SIPF), an effective filtering method, was used at first.

**Keywords:** Distributed denial-of-service, Hyper-heuristic Butterfly Optimization Algorithm, Software-defined networks, Support Vector Machines, Source-based IP filtering.

## 1 Introduction

The newest network architecture, known as SDN, is a dynamic, programmable, and adaptable structure that can be customized for any application [1]. Traditional cloud network design is not equipped to handle demands for high availability, fast processing speeds, rigorous management, and virtualization. As an alternative to such outdated architecture, SDN has materialized and offers great levels of flexibility and control to service providers and end users [2]. Additionally, the evolution of Network Functions Virtualization (NFV) has creatively enhanced the traffic monitoring and regulating features in a cloud network. By separating the virtualized functions from the physical equipment, the NFV enables cost-effective network design, deployment, and control [3]. Data, controller, and application layer planes make up the SDN, with the first two layers planes operating separately. Switches and routers are used in the data plane to forward packets, whereas Beacon, Floodlight, POX, and NOX controllers are smart regulator devices found in the controller plane [4]. For the edge of data transmission, Open-Flow is used. The SDN is dynamically optimized from a single location since these local control devices employ

the control logic and take on the role of the central authority. This characteristic increases the adaptability of the tee tire architecture [5].

In the suggested model, the SVM model configuration is optimized using a hybrid optimization technique called the Hyper-heuristic Butterfly Optimization technique (HHBOA). This results in a competent Support Vector Machines (SVM) based hybrid model. Using HHBOA, the best configuration of SVM is created by selecting the boundary parameter (or penalty), the type of kernel, and its constraints. To increase the accuracy of the detection, the SIPF is also introduced. Utilizing a DDoS evaluation dataset included in the SDN-based cloud model created using MATLAB, the proposed hybrid machine learning model of HHBOA-SVM and SIPF is assessed. The rest of this article presents the associated works, the SIPF and HHBOA-SVM - based models' designs, implementations, experimental findings, and conclusions.

## 2 Related Works

Studies that use machine learning techniques for DDoS recognition and eradication are becoming more prevalent, as evidenced in recent years. Recent works that use various machine learning classifiers for DDoS identification in SDN are discussed in this subdivision. For the purpose of identifying low rate-DDoS sessions, Wang et al. proposed the Hidden Markov Model with Renyi entropies (HMM-R) technique. The IP addresses of the data packets were employed in this method to compute the Renyi entropies after the data packets were collected and analysed from the data centre networks. In order to create a probability model that can identify low frequency DDoS attacks, these entropies are used as features for the classification process using HMM. With high detection rates and a detection time of 0.2227 seconds, this method produced 97.11% true positives and 1.81% false positives. When other DDoS attacks are used, however, this strategy performs poorly. Through mini-net and floodlight simulations using 6-tuple distinctive values from the control flow tables, Ye et al. [6] established a DDoS attack identification algorithm based on SVM. The multidimensional traffic data is processed next for highly accurate classification using the SVM classifier. With a low false positive rate of 1.26%, this SVM strategy achieved accuracy levels of 95.24% on average. However, the lack of adequate analysis of typical data flows may have also contributed to the low false positive rates. Likewise, there is a problem with the extremely low detection rate of ICMP flows.

The Advanced SVM-based DDoS attack detection model was first published by Myint Oo et al. [7]. Three issues with the traditional SVM are addressed by the suggested ASVM algorithm. The SDN design makes it difficult for SVM to manage numerous controllers. To address these problems, the ASVM was created, and five traffic features were used in its testing. With 50 seconds of training time and 55 seconds of testing time, the ASVM achieved 97% accuracy, 97% detection rate, and a 3% false alarm rate. The DDoS attacks in other SDN attack planes, however, cannot be mitigated by this way. Improved K-Nearest Neighbours (KNN) for DDoS attack recognition models with various attack intensities were developed by Dong and Sarem [8].

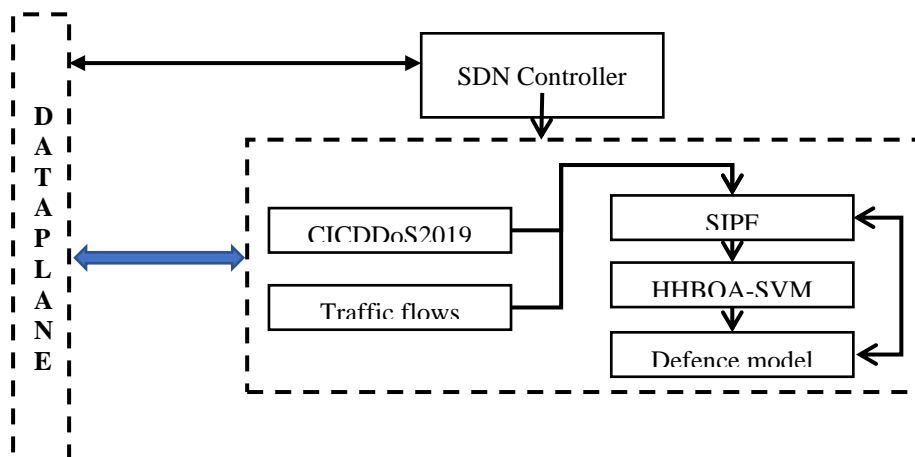
Through the use of an improved history-based IP filtering (eHIPF), Phan and Park [13] created a DDoS detection and prevention strategy. While an SVM classifier and the self-organizing map (SOM) algorithm are used to improve detection accuracy, the attack recognition rate increased. The model complexity and storage costs are significant, despite the fact that it obtained 99%

accuracy on CAIDA datasets. Using statistical and machine learning techniques, Dehkordi et al. created a DDoS detection algorithm. This model uses Bayes Network, J48, Random Tree, logistic regression, and REP Tree classification algorithms to detect DDoS attacks. It also includes a data collector and static and dynamic threshold-based entropy methods. For the UNB-ISCX and CTU-13 datasets, respectively, the trials were carried out over legitimate datasets, and the results showed that the REP Tree classifier with dynamic threshold based entropy approach achieved high accuracy of 99.12% and 99.85% with a false-positive rate of 0.1% and 0.04%. However, this statistical and machine learning (ML) based methodology only uses one controller in SDN to identify threats, which increases detection time.

The main conclusion drawn from the studies covered in this part is that machine learning algorithms still need to be improved in order to recognise different types of DDoS attacks. Additionally, it must guarantee that the classifier will avoid the over-fitting issue, simplify models, and use less resources. Additionally, filtering approaches can improve classifier accuracy while significantly raising storage costs. Taking into account these considerations, the suggested research study creates a powerful ML-based DDoS detection model using SIPF and HHBOA-SVM.

### 3 Methodology

To address the issues with model complexity and expensive storage, the suggested methodology introduces SIPF and HHBOA-SVM. To reduce the issue with inappropriate training data, this solution performs pre-processing on the traffic flow data. Following the completion of the pre-processing, the characteristics of valid and anomalous traffic are identified and screened using SIPF to increase the rates of attack detection. The hybrid classifier HHBOA-SVM divides the normal, other attack, and DDoS classes of data into normal, based on the findings of the filtering method. The SIPF and HHBOA-SVM based defence architecture is then activated to reduce DDoS attacks. The proposed methodology's design is depicted in Fig. 1.



**Fig.1.** Conceptual architecture design of the proposed methodology

DDoS assaults can currently be stopped using a variety of source-based IP filtering techniques. Because of the statistics used to analyse the nature of traffic packets when using the SIPF, the

storage costs appear to rise dramatically. For sensing the traffic data, the SIPF makes use of the source IP address and TTL value. This data can be utilised to track attack traffic flows because there is substantial correlation in regular traffic. This data is utilised to create a data table for each source IP in the state of normal traffic. In the event that the attacks are recognised, SIPF filters attack data packets based on their source IP, and their hopping would not coexist with regular traffic. However, the storage resources required and the associated expenses rise as a result of the filtering step. The SIPF has a cutting-edge counting-based bloom filter that lowers the indicators of the hopped originating IP address in order to cut down on these storage expenses.

Along with corresponding hop-counts, the information table between the source and its consumers is formed on the indicators of the source IP.  $S_0, S_1, \dots, S_{(n-1)}$ ;  $1n32$  signifies  $n$  packet segments, with the range of  $n$  segments represented as  $0$  to  $-1 N=2(32/n)$ . HC and OHC31 are symbols for the hop-count. The IP address and hop-count data are gathered for each server to create this database. To the extent that  $0j_{n-1}$  and  $0i_{N-1}$ , let  $x_{(j,i)}$  represent the mathematical evaluation of the data  $i$  of the  $j$ -th source IP fragment. The two components of  $x_{(j,i)}$  are the current statistics ( $p_{(j,i)}$ ) and the typical traffic statistics ( $q_{(j,i)}$ ). In most cases, the victim controller changes the value of  $q_{(j,i)}$  periodically based on  $p_{(j,i)}$ . The attack's processing will halt the evolution process.

The incoming packets  $S'$  will be checked for the DDoS attack based on its hop-count and IP address.  $S'$  Will be split into  $n$  segments as  $S' = \{S'_0, S'_1, \dots, S'_{n-1}\}$ . For each  $S'_j$ , the packet score is computed as

$$G(S'_j) = \begin{cases} p_{j,k} - q_{j,k} & \text{if } q_{j,k} < p_{j,k} \\ \infty & \text{if } q_{j,k} = 0 \\ 0 & \text{if } q_{j,k} > p_{j,k} \end{cases} \quad (1)$$

Here  $k$  denotes the statistics value of  $S'_j$ . Now, the attack packet and normal packets are distinguished by using a simple function

$$\|f(S')\| = \sum_{j=1}^{n-1} G(S'_j) \quad (2)$$

As previously noted, a threshold is established for the number of packets' related hops based on typical traffic. The calculation is

$$\delta = n \max_{1,j} |p_{j,i} - q_{j,i}| \quad (3)$$

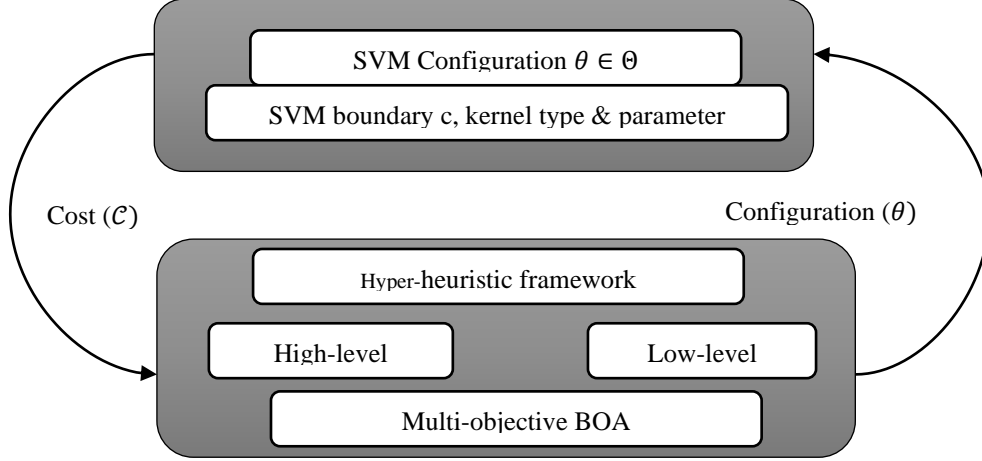
This threshold will be compared to Eq. (2), and if it holds, the packet is attacking.

$$\|f(S')\| \geq \delta \cdot w \quad (4)$$

The DDoS attack's intensity factor,  $w$ , is indicated below. The attack is deemed to be more potent when  $w$  is lower. It is calculated as the ratio of current statistics  $p_m$  and normal statistics  $q_m$  when the hop-count  $HC = m$  such that  $w = \frac{q_m}{p_m}$ . When  $w = 0$ , the packet is attacking packet while  $w = 1$  denotes the normal packet. it is a regular packet.

## HHBOA-SVM

The SVM and the hyper-heuristic BOA model make up the proposed HHBOA-SVM structure. The BOA is based on how butterflies naturally forage and mate. The suggested HHBOA-SVM working model is shown in Fig. 2.



**Fig.2.** Proposed HHBOA-SVM classifier

The suggested HHBOA carries out the methods of performance improvement through the structure selection for the SVM model. The regulation of the kernel type, its parameters, and the border parameter determine the best SVM configuration. When BOA is present, a fragrant fluid is emitted, attracting butterflies to other butterflies. Butterflies gravitate towards those that generate stronger scents, and the intensity of the butterfly stimulus will depend on the butterfly's goal function. The SVM configuration objective function, with each solution being mapped as butterflies, replaces this fragrance in the proposed HHBOA. Based on the accuracy parameter and the quantity of support vectors, the cost utility is modelled. The SVM's single, best-performing structure  $(\theta)$ , which is a one-dimensional group with the selected objective constraints, is the illustrative representation. The population of butterflies, or set of credible solutions, is where the system initialization begins. Each decision parameter, parameter  $B = \{b_1, b_2, \dots, b_n\}$ , is given an arbitrary value to allow for unrestricted adjustment of the population of butterflies.

$$b_i^u = l_i^u + Rand_i^u(0,1) \times (u_i^u - l_i^u),$$

$$u = 1,2, \dots, |U|; i = 1,2, \dots, d \quad (5)$$

Here,  $|U|$  denotes the population's size,  $Rand_i^p(0,1)$  denotes the arbitrary number *fori-th* decision parameter,  $l_i^u$  and  $u_i^u$  are the bottom and top boundaries, respectively.  $i=1, 2, \dots, d$  represents the decision parameter index,  $d$  characterises the aggregate sum of decision parameters,  $u$  is the probability, and  $d$  characterises the decision parameters' sum.

Based on the objective function  $F_i$ , which is provided as, fitness values are evaluated.

$$\text{Minimize } F_i = |f_1(b), f_2(b)| \quad (6)$$

Where the accuracy utility is denoted by  $f_1(b)$ , and the utility for the support vector quantity is denoted by  $f_2(b)$ .

The best butterfly (optimal configuration) is selected using both local and global search techniques. Moving the butterfly towards the ideal position ( $g^*$ ), based on fitness values, completes the global search. It is provided in HHBOA as follows:

$$b_i^{v+1} = b_i^v + (r^2 \times g^* - b_i^v) \times F_i \quad (7)$$

Here  $b_i^v$  and  $b_i^{v+1}$  here denote, respectively, the  $i$ -th butterfly's current location and its upcoming position. According to Eq. (6),  $g^*$  stands for the optimal current position,  $v$  for the current iteration,  $F_i$  for the fitness values, and  $r$  for the randomly chosen integer,  $r \in [0,1]$ .

The local search is described similarly as

$$b_i^{v+1} = b_i^v + (r^2 \times b_j^v - b_i^v) \times F_i \quad (8)$$

Where  $b_j^v$  and  $b_i^v$  are the positions of butterflies  $v$  and  $t$ , respectively

The low-level heuristics incorporate the set of problem-related standards created to provide reliable solutions for each issue case that has been given. A minimum of one solution is considered by the low-level heuristics, which either combine or modify them to form a different solution set. The solutions are constructed using various search-based tasks. To analyse and frame new sets, the HHBOA-based search process is used. They are archived in the non-ruled set of solutions after being created by low-level heuristics and being determined by the significant level approach. Based on the Pareto-front, the algorithm selects solutions from this stored location, and it presents the best structure as the final result. Algorithm 1 provides a summary of the proposed HHBOA-SVM.

#### **Algorithm 1: HHBOA-SVM**

Step 1: Begin

Step 2: Set population of butterflies  $B = \{b_1, b_2, \dots, b_n\}$

Step 3: Estimate the accuracy and number of support vectors

Step 4: Determine the value of probability parameter  $u \in [0,1]$

Step 5: Set iteration  $v = 0$  While  $q < q_{max}$ , do

    For each  $b$  in  $B$

        Evaluate the fitness  $F_i$  using Eq. (6)

    End for

Step 6: Analyse and determine the current best  $b$

Step 7: Select a threshold  $th$ ;  $th \in [0,1]$

Step 8: If  $th < p$  then

    Initiate global search through Eq. (7) Else

    Initiate local search through Eq. (8)

End if

Step 9: Apprise  $b$ ,  $u$  and  $v = v + 1$

Step 10:Apply low-level heuristics

Form new solutions

End while

Step 11:Apply high-level heuristics

Form new solutions, Estimate fitness values

Step 12:If  $th < p$  then

Initiate global search through Eq. (7) Else

Initiate local search through Eq. (8)

End if

Step 13:Apprise  $b, u$  and  $v = v + 1$

Step 14:Return final best  $b$

End

The defensive model is started to mitigate the attacks as soon as the SIPF and HHBOA-SVM detect DDoS attack packets. In the case of benign attacks, the regulator sends a modulation packet to the clients in order to start the drop process as well as updating the tables.

## 4 Performance Evaluation

In a MATLAB-simulated SDN-based cloud system, the suggested methodology's effectiveness is assessed. An Open-Flow switch, an attacker, a rudimentary SDN topology, a controller for the SDN, and server connections are initialised. The algorithms are trained on the CICDDoS2019 dataset, which is then used to evaluate the system. CICDDoS2019: For the year 2019, the CIC created this dataset. DDoS assaults of various kinds can be found in this dataset. Reflection and DDoS exploitation are the two main categories. These assaults include of MSSQL, LDAP, PortMap, and NetBIOS assaults. SYN flood is an example of a TCP abuse event, whereas UDP flood and UDP-Lag are examples of an assault. The first data dataset is the only one included in this evaluation out of the two sub-divisions that make up this dataset.

SIPF and HHBOA-SVM performance is compared to that of the eHIPF and SVM-SOM model. Since it has the highest accuracy of 99.8% in the literature, this approach was chosen for comparison. The eHIPF and SVM-SOM model is used in this evaluation and applied to the CICDDoS2019 dataset in a controlled setting. The performance is contrasted using the subsequent measures.

$$Accuracy = \frac{(TP+TN)}{(true\ values+false\ values)} \quad (9)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (10)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (11)$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

$$FPR = \frac{FP}{(FP+TN)} \quad (13)$$

$$FNR = \frac{FN}{(FN+TP)} \quad (14)$$

Kappa coefficient determines the ability of an algorithm to handle imbalance class instances, which can be computed as

$$K = \frac{Obsolute - Expect}{1 - Expect} \quad (15)$$

Here *Obsolute* = Accuracy And  $Expect = \frac{A+B}{(true\ values + false\ values)}$ .

The values of A and B can be obtained as  $A = \frac{(TP+FN)(TP+FP)}{(true\ values + false\ values)}$

and  $B = \frac{(FP+TN)(FN+TN)}{(true\ values + false\ values)}$ .

It ranges between (0, 1) and step size of 0.2. Thus, a total of six performance classes will be generated for the classifier as poorest ( $K \leq 0$ ), slim poor ( $0 < K \leq 0.2$ ), permissible ( $0.2 < K \leq 0.4$ ), medium ( $0.4 < K \leq 0.6$ ), substantial ( $0.6 < K \leq 0.8$ ) and near-ideal ( $0.8 < K \leq 1.0$ ). The comparative findings between SIPF and HHBOA-SVM for various DDoS attack types in the CICDoS2019 dataset are shown in Table 1.

**Table.1.** Performance evaluation over CICDDoS2019

Methods/ Metrics	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	FPR (%)	FNR (%)
Port Map	94.64	94.55	95.67	94.63	0.9	1.2
Net BIOS	88.97	94.5	97.86	94.78	2.67	1.67
LDAP	99.24	99.7	94.65	96.78	1.78	1.5
MS SQL	97.87	99.67	98.78	98.56	1.98	2.05
UDP	99.56	98.92	98.33	98.45	0.57	1.76
UDP-Lag	99.45	99.18	89.67	94.87	1.6	1.11
SYN	96.8	98.65	91.91	96.87	1.21	2.09

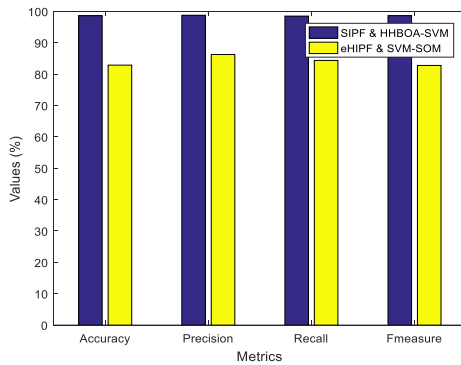
The results in Table 1 demonstrate unequivocally that the SIPF & HHBOA-SVM model's great performance for all DDoS attack types was obtained with high accuracy and low FPR and FNR values. This demonstrates how easily the suggested paradigm can be modified to handle different types of DDoS attacks in SDN. The two approaches used at CICDDoS2019 are evaluated in Table 2.



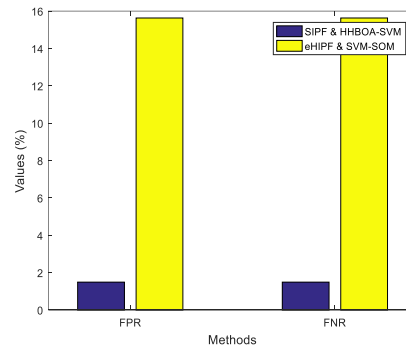
**Table 2.** Comparison of the performance of SIPF and HHBOA-SVM vs eHIPF and SVM-SOM

Method s/ Metrics	Accurac y (%)	Precisio n (%)	Recal l (%)	F-measure (%)	FP R (%)	FNR (%)	Kappa Coefficien t (%)	Processing time (s)
eHIPF& SVM- SOM	82.88	86.26	84.38	82.78	15.6	15.6	86.54	945.62
SIPF& HHBOA -SVM	98.64	98.78	98.51	98.63	1.49	1.52	97.25	505.78

The outcomes shown in Table 2 demonstrate that the proposed SIPF & HHBOA-SVM model has beaten the current eHIPF & SVM-SOM. The proposed model exhibits a 16% increase in accuracy, a 12% increase in precision, a 14% increase in recall, a 16% increase in F-measure, a 14% decrease in FPR, a 14% decrease in FNR, and a 10.7% increase in Kappa coefficient. Additionally, compared to the current model, the proposed model has taken much less time. This is mostly caused by fewer incorrect classification values and lower storage expenses. The comparison graphs for these ratings are shown below.



**Fig 3.** Performance comparisons



**Fig.4.** FPR and FNR

Fig. 3 compares the evaluation results of the suggested SIPF and HHBOA-SVM with the current eHIPF and SVM-SOM. It shows that the suggested strategy has obtained high performance measure values. This speed improvement is made possible by HHBOA's ideal SVM design in conjunction with better filtering, which has effectively mitigated DDoS attacks. The false value comparisons between the existing eHIPF & SVM-SOM and the suggested SIPF & HHBOA-SVM are shown in Fig.4 in terms of FPR and FNR. Low FPR and FNR were attained by the suggested model. Furthermore, the suggested approach features a well-adjusted trade-off between FNR and FPR, which has produced values for both metrics that are roughly identical. As a result, it can be concluded that the suggested model will work very well with SDN-based cloud architectures.

## 5 Conclusions

Through the creation of SIPF and HHBOA-SVM, an advanced IDS model was presented in this study article.. Additionally, with the ideal SVM design, it is possible to identify DDoS attacks with a high rate of detection while using fewer resources and less calculation time. In a simulated SDN-based cloud environment, the performances were evaluated using the CICDoS2019 dataset. The results supported the claim that the SIPF and HHBOA-SVM model enhanced DDoS attack detection with active recognition of various attack types. It drastically reduced the incorrect parameters and increased accuracy. The SIPF and HHBOA-SVM based solution in the SDN-based cloud is significantly capable of identifying DDoS attacks. Future research will look into filters that could enhance adaptive packet dropping, such adaptive filtering. In order to increase the detection rate, more research can be done on the packet\_in process of the SDN.

## References

- [1] Kirkpatrick, K. (2013). "Software-defined networking." *Communications of the ACM*, 56(9), 16-19.
- [2] Farhady, H., Lee, H., & Nakao, A. (2015). "Software-defined networking: A survey." *Computer Networks*, 81, 79-95
- [3] Jain, R., & Paul, S. (2013). "Network virtualization and software defined networking for cloud computing: a survey." *IEEE Communications Magazine*, 51(11), 24-31.
- [4] Yeganeh, S. H., Tootoonchian, A., & Ganjali, Y. (2013). "On scalability of software-defined networking." *IEEE Communications Magazine*, 51(2), 136-141.
- [5] Mohammed, A. R., Mohammed, S. A., & Shirmohammadi, S. (2019). "Machine learning and deep learning based traffic classification and prediction in software defined networking." In 2019 IEEE International Symposium on Measurements & Networking (M&N) (pp. 1-6). IEEE.
- [6] Ye, J., Cheng, X., Zhu, J., Feng, L., & Song, L. (2018). "A DDoS attack detection method based on SVM in software defined network." *Security and Communication Networks*, 2018.
- [7] Myint Oo, M., Kamolphiwong, S., Kamolphiwong, T., & Vasupongayya, S. (2019). "Advanced support vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN)." *Journal of Computer Networks and Communications*, 2019.
- [8] Dong, S., & Sarem, M. (2019). "DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks." *IEEE Access*, 8, 5039-5048.
- [9] Xu, Y., Sun, H., Xiang, F., & Sun, Z. (2019). "Efficient DDoS detection based on K-FKNN in software defined networks." *IEEE Access*, 7, 160536-160545.
- [10] Liu, Z., He, Y., Wang, W., & Zhang, B. (2019). "DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN." *China Communications*, 16(7), 144-155.
- [11] Ma, Z., & Li, B. (2020). "A DDoS attack detection method based on SVM and K-nearest neighbour in SDN environment." *International Journal of Computational Science and Engineering*, 23(3), 224-234.
- [12] Pérez-Díaz, J. A., Valdovinos, I. A., Choo, K. K. R., & Zhu, D. (2020). "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning." *IEEE Access*, 8, 155859-155872.
- [13] Phan, T. V., & Park, M. (2019). "Efficient distributed denial-of-service attack defense in SDN-based cloud." *IEEE Access*, 7, 18701-18714.
- [14] Arora, S., & Singh, S. (2019). "Butterfly optimization algorithm: a novel approach for global optimization." *Soft Computing*, 23(3), 715-734.

- [15] Lavanya A, (2023) "Enriched Model of Case Based Reasoning and Neutrosophic Intelligent System for DDoS Attack Defence in Software Defined Network based Cloud" International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 11 Issue: 4
- [16] J. K. Jaiswal, R. Samikannu and I. Paramasivam, "A Survey on Contemporary Security Threats in Big Data and Information System," 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), Tindivanam, India, 2017, pp. 263-268, doi: 10.1109/ICRTCCM.2017.33.
- [17] Omejevwe Efe-odenema and Jitendra Jaiswal (2020), 'Webpage Classification for Detecting Phishing Attack', 2020, <https://api.semanticscholar.org/CorpusID:235497818>