Journal of Optimization of Soft Computing (JOSC)

Vol. 1, Issue 2, pp: (54-63), December-2023

Journal homepage: https://sanad.iau.ir/en/Journal/josc

Research paper

An optimal approach to detect anomalies in intrusion detection systems

Afsaneh Banitalebi Dehkordi[1*]

1. Department of Computer Engineering, Payame Noor University (PNU), P.O.Box 19395-4697, Tehran, Iran.

Article Info

Article History:
Received: 2024/3/3
Revised:
Accepted: 2024/4/3

Keywords:
Machine Learning,Security, Intrusion Detection System,Software Defined Network

***Corresponding Author's Email Address:** abanitalebi@pnu.ac.ir*

Abstract

Software Defined Networking (SDN) is considered as an innovate architecture of computer networks by using the central controller. Any modification in network data and its arrangement can be effortlessly executed in software via the controller in these networks. Consequently, the identification and timely response to Distributed Denial of Service (DDoS) attacks can be achieved, which is not the case in conventional networks.This paper uses the α-Entropy statistical method considering a threshold and machine learning techniques, K-Nearest Neighbor (KNN), Random Forest (RF) and Support Vector Machine (SVM) to increase the accuracy of detecting DDoS attacks. In this method, the results are evaluated by 10-fold cross validation. The used dataset is ISOT, CTU-13 and UNB ISCX. The results of evaluation with a precision of 99.84% and FPR value of 0.10% indicate the high efficiency of the proposed model in SDN networks.

1. Introduction

SDN is constructed as new network architecture to provide more flexibility in software control of network. In the SDN architecture, the data and control layers are separated, the intelligence of network is centralized and the network infrastructure is separate from the applications [1]. Figure 1 shows the overall architecture of the Open Flow network.

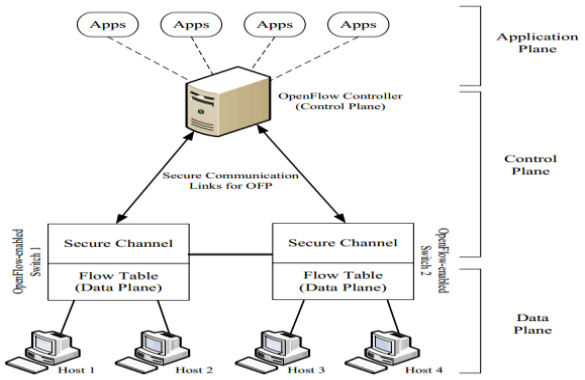

Figure 1. SDN architecture

In this architecture, the flow control is removed from the hardware level in network nodes and is taken centrally and separately by the controller.
The network operating system manages the SDN switches and gathers information from the APIs to create the data plane. [2].An essential situation in SDN networks, which distinguishes them from traditional networks, entails the ability of DDoS attacks to directly target the controller through the transmission of an excessive number of packet-in packets. Consequently, the controller must take into account certain rules for these packets in the flow table of switches and subsequently populate it with counterfeit flow entries[3]. In this research, a hybrid technique is presented to enhance the accuracy of detecting the DDoS attacks. Other sections of this article include the detection of DDoS attacks and their methods, the importance of security issues in SDN networks, the proposed method of this paper to detect DDOS attacks on SDN networks with a combination of statistical methods and machine learning including support vector machine, and a solution to obtain threshold

dynamic, introduction of the UNB-ISCX dataset used in this article, the evaluation of the proposed solution, the results of the statistical analysis and machine learning on the given dataset are evaluated, and finally concluding remarks.This paper is segmented into the subsequent divisions. We articulate certain techniques regarding the identification of DDoS in Division 2. Division 3 formulates a novel approach to identify DDoS assaults through utilization of the α-entropy methodology and classification algorithms. Division 4 discloses the collections of data. Division 5 encompasses the assessments of the performance of our proposed identification techniques on UNB-ISCX, CTU13, and ISOT datasets and elucidates the implementation surroundings. Division 6 recommends the experiments. The outcomes of this manuscript are expounded in division 7. Division 8 contrasts the techniques introduced in this study with some existing techniques. Ultimately, Division 9 presents a concise conclusion and future prospects. Numerous symbols are employed in this manuscript as depicted in Table 1.

Table 1. Major Notations used in this paper

Symbol	Explanation
DDoS	Distributed Denial Of Service
SDN	Software Defined Network
TF	Tolerance-Factor
TP	Time Period
RBF	Radial Basis Function
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
RF	Random Forest
GE	Generalized Information Entropy
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
TPR	True Positive Rate
FPR	False Positive Rate
AC	Accuracy
AR	Alarm Rate
F_i	Flow number i
T_1	Threshold

2. Related Work

Diverse methodologies were examined in the studies conducted on SDN networks to enhance the security measures implemented on these networks. The ensuing instances illustrate these approaches. In [4], Baczak et al. investigated the data mining methods and machine learning in detecting cyber space penetration. These methods were compared in different aspects such as precision, period of model training, time to classify an unknown sample by the trained model, and the impact of each one in detection of attacks. Yan et al. in [5] proposed a lightweight technique to detect the DDoS attacks. This method is based on hybrid fuzzy evaluation of decision-making model. For example, in case of the denial of service attacks, the network service may be slightly affected, but may still indicate normal status. In this case, 15% of DDoS attack occurs, rather than without any attack. This study uses a variety of features to identify whether an attack occurs or not. Using the formulas presented in the article, the denial of service attacks are detected. Stoyanova et al in [6] proposed a method of detecting a DDoS attack in VANET architecture. The test involves normal and DDoS attack traffics with IP addresses of spoofed resources. Based on traffic features such as the IP address of destination, and through the contribution of three windows with 30 packets, the entropy is computed. The attacks can be detected by calculating the entropy and defining a proper threshold. The provided solution is efficient and adds a low amount of code to the controller module, leading to little impact on the use of CPU.

In another research, Kia in [7] introduced a credible and lightweight solution to detect the denial of service attacks. The proposed method consists of calculating the entropy of the destination IP and the initial rate of flow and examining the characteristics of the packets, by allocating little time to detect attacks. The reduction method involves reducing the idle timer of flow, which is implemented in the case of an attack to contribute the switches to be less under-attack. The use of a single static threshold in this research increases the chance of FP and FN reports. Therefore, the need for a system, which uses optimal thresholds, is felt.

A new detecting method of denial of service attacks, involves calculating the entropy of destination IP to detect attacks in [8]. The provided solution is not only efficient, but also includes a low amount of code in implementation of the controller module, and does not increase the CPU load either in normal or in the event of an attack. In this research, the method of attack detection is based on entropy. One of the limitations of the mentioned method in this paper is that the entropy value cannot accurately detect the attack when the entire network and all hosts are under DDoS attack. A combination of statistical and machine learning methods is considered as the proposed method in this research to identify DDoS attacks. In the statistical section, three types of , Hartley entropy, Shannon entropy and collision entropy method are examined. In the machine learning section, the Support Vector Machine (SVM), K-Nearest

Neighbor (KNN) and Random Forest (RF) techniques are used to enhance the accuracy of DDoS attack detection.

3. Proposed Method

In order to detect DDoS attacks in SDN networks, the proposed method is the combination of α-entropy statistical method and the SVM, KNN and RF Models to increase the accuracy of attack detection. This method is a two-class problem and detects normal flows and attacks. The flowchart, as shown in Figure 2 represents the method. The approach illustrated in Figure 2 comprises various applications. These applications collaborate in order to identify DDoS attacks occurring within the Floodlight controller. Each individual section is introduced in the subsequent text.

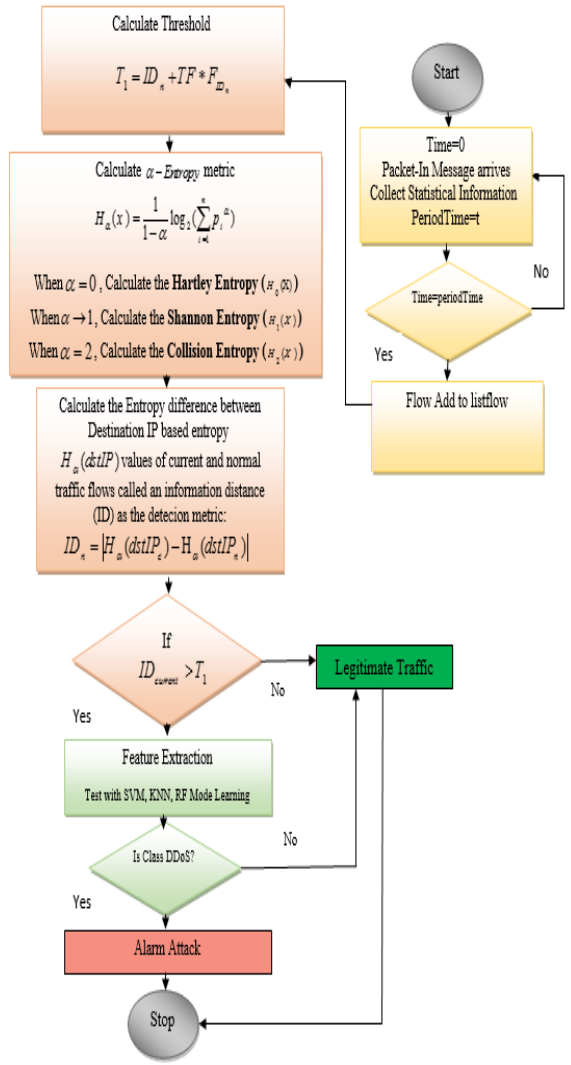

Figure 2. The Proposed Method

3.1. Proposed Statistical Method of α-Entropy

In 1948, Alfred Renyi introduced a more comprehensive definition of Shannon Entropy called generalized information entropy (GE) of the alpha grade, known as α-Entropy or Renyis' Entropy [9], as shown in equation 1.

$$H_\alpha(x) = \frac{1}{1-\alpha}\log_2(\sum_{i=1}^{n} {p_i}^{\alpha}) \quad (1)$$

Where $\alpha \geq 0$ and $\alpha \neq 1$.

- When $\alpha=0$, Hartley entropy of X is as equation 2:

$$H_0(X) = \log n = \log|X| \quad (2)$$

- When $\alpha \to 1$, the amount of Shannon entropy is as equation 3:

$$H_1(x) = -\sum_{i=1}^{n} p_i \log p_i \quad (3)$$

- When $\alpha=2$, Collision entropy is as equation 4:

$$H_2(X) = -\log \sum_{i=1}^{n} p_i^2 \quad (4)$$

In Figure 3, types of α-Entropy is shown

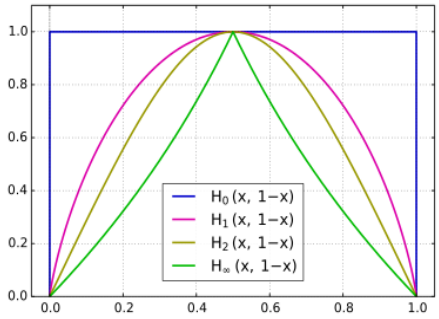

Figure 3. Renyi entropy of a random variable with two possible outcomes

This research addresses the detection of DDOS attacks and, attempts to select the best ID_n of α and increase the accuracy of attack detection by considering different values of α.This technique classifies the network flows in each time window and by calculating the ID_n that represents the difference value of the α-Entropy based on the destination IP address in two network traffic flow in normal and current modes, and it is recognized as detection metric. ID_n is defined in equation 5:

$$ID_n = |H_\alpha(dstIP_c) \_ H_\alpha(dstIP_n)| \quad (5)$$

Where, H_α indicates the α-Entropy in current and normal modes of network traffic flow.
Now, if this value is greater than threshold (T1) as shown in equation 6, attack is detected; otherwise, non-attack status is detected.

$$ID_n > T_1 \quad (6)$$

This technique is able to quickly detect DDoS attacks with low number of packets.

3.1.1. Calculation of Thresholds

In this study, a time-based computational method is used to compute the threshold. One of the aims of this threshold is to detect quickly DDoS attacks in time windows with small width. The relation of this threshold, T1 is presented in equation (7).

$$T_1 = ID_n + TF * F_{ID_n} \quad (7)$$

Where, ID_n value indicates the α-Entropy difference based on the destination IP address in two normal and current modes of network traffic flow as shown in equation 8. TF value is a fixed value. It belongs to the integer set, and is an experimental value used to adjust and improve the threshold dynamic values to detect DDoS attacks. F_{ID_n} represents the standard deviation of ID_n.

$$F_{ID_n} = \frac{1}{t}\sum_{i=1}^{t}(ID_n - \overline{ID}_n)^2 = \frac{1}{t}\sum_{i=1}^{t} ID_n{}^2 - \overline{ID}_n^2 \quad (8)$$

Where, F_{ID_n} represents the standard deviation of ID_n between normal flows.

3.2. Proposed Machine Learning Method

At the second phase of the proposed approach, a proportion of the dataset that was identified as an attack in the preceding phase, regardless of its accuracy, was forwarded to the machine learning component. The remaining dataset being detected correctly was filtered at this step. Different steps of machine learning are described below. Therefore, the pre-processing step and balancing normal data and attack are conducted. This research specifically focuses on the SVM, RF and KNN methods. The methodology used to detect DDoS attacks is analyzed. Then, the results of the SVM, RF and KNN algorithm are compared with other machine learning algorithms.

3.2.1. Extraction of characteristics

In this study, 11 features were considered for detecting the normal and attack packets. Table 2 introduced these features.

Table 2.Feature Extraction

Feature	Explanations
SumNeighborRelationshipsSrc	The number of neighbors connect to transmitter
CountUniDirectionalSrcSToD	The proportion of one-way connections that functioned as the origin for transmitting signals in relation to the overall number of connections to the desired node.
EntropyBytePerPacketSentSrc	Calculating the entropy
SumNeighborRelationdhipDst	The number of neighbors connect to recipient
CountUniDirectionalDstSToD	The proportion of one-way connections that served as the host for the recipient compared to the overall connections of the desired node.
CountSentSrc	The number of the flows which were the desired host of the transmitter.
CountReceiveSrc	The number of the flows which were the desired host of the recipient.
EntropyByte perPacketSentDst	Calculating the entropy of the flows which were the desired host of the recipient
CountSentDst	The number of flows which were the desired host of the transmitter
CountReceiveDst	The number of flows which were the desired host of the recipient
Packet In	This feature is the first packet which transmits a flow for any host in SDN.

These features can be extracted from the switches and controller of the SDN[10]. They are calculated for both ends of a flow and used for each flow in learning. Since separate rules are written in the flow table for two-way communications in software defined networks, the time and characteristics of the receiver and sender determine which flow is related to communication. After building all existing flows, the data between the two hosts in the form of an array which holds the data is given to the next step. These features can be extracted from the switches and controller of the SDN[11].

3.2.2. Support Vector Machine Model

SVM technique is developed by Vladimir Vapnik in 1961 and is based on the linear classification of data [12]. The SVM method uses structural risk minimization, while in other methods the empirical risk minimization is used. Some studies indicated that the SRM method had better performance than the ERM [13]. This paper uses the radial basis

function (rbf) kernel. This kernel function is considered as one of the best functions available in this field. Here, the main parameters of C and Gamma are shown. The parameter C indicates SVM penalty and gamma is presented as kernel coefficient. They are evaluated for the kernel function with the values of Table 3 by the corresponding Python code.

Table 3. C and Gamma values

C	Gamma
١	١
١٠	٠,١
١٠٠	٠,٠١
١٠٠٠	٠,٠٠١

As observed in Algorithm 1, the Python code in this algorithm is used to determine the best value of the C and gamma parameter in the set of determined data, and the best combination of C and gamma parameters in the SVM is obtained automatically by analyzing various combinations of them to find the best responses to the intended dataset, by using the GridSearchCV class in Python.

Algorithm 1.Find Best Parameter values in SVM model

```
From sklearn.model_selection import GridSearchCV
From sklearn.svm import SVC
param_grid = {'C': [1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01,
   0.001], 'kernel': ['rbf']}
grid_svm = GridSearchCV(SVC(), param_grid, refit = True)
```

The tested parameters values are in the param_grid and the optimal ones are in the grid_svm.

3.2.3. K-Nearest Neighbor Model

In this section, the KNN Model is expressed to present the best value of K in set of specified values using elbow technique, For this, the value of 1 to 25 for k was investigated and the error rate was shown by the elbow technique as in Figure 4.

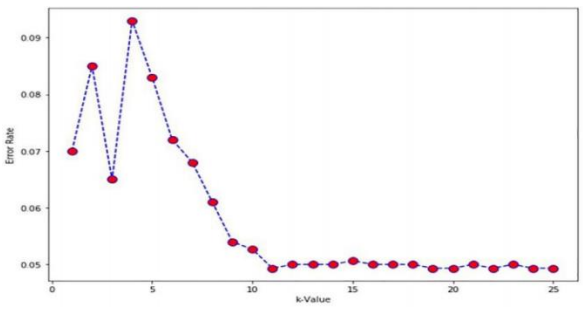

Figure 4. K-Value and error-rate

The results show that the optimum value of k=11 was selected by this technique because in this case the lowest error rate was obtained with value 0.049.In algorithm 2, best k-value is calculated.

Algorithm 2.Find Best K-value in KNN Model.

```
From sklearn.neighbors import KNeighborsClassifier
err_rate = [ ]
For k in range(1, 26):
   knn = KNeighborsClassifier(n_neighbors = k)
   knn.fit(X_train, y_train)
   k_opt = knn.predict(X_test)
   err_rate.append(np.mean(k_opt != y_test))
```

In this algorithm X_ train represents features, y_ train represents the value of output and y_test represents feature test.

3.2.4. Random Forest Model

With Random Forest Model, best response are fined with various decision trees. Four number of decision trees (n_ estimators) are investigated:

- n_ estimators = 10,100,500,1000

The best 'n_estimators' value is provided in Python code in algorithm 3.

Algorithm 3.Find the best decision trees

```
From sklearn.model_selection import GridSearchCV
From sklearn.ensemble import RandomForestClassifier
param_grid = {'n_estimators': [10,100,500,1000], 'criterion':
   ['gini']}
grid_rf = GridSearchCV(RandomForestClassifier(), param_grid,
   refit = True)
```

In this algorithm, the numbers of trees which is tested, holded in param_grid and the optimal response is in the grid_rf.

4. The Datasets

This study employs the UNB-ISCX dataset for the purpose of detecting Distributed Denial-of-Service (DDoS) attacks. This dataset is downloaded from the University of Canadian Institute of Cybersecurity website and is used to detect attacks. In the dataset presented in Table 4, a range of tools were employed to initiate different attacks. The resulting outcomes encompasses a period of network traffic, equating to a total size of 4.6 GB.

Table 4.Data statistics of the modified UNB-ISCX dataset

Application DDoS attack	Av duration	Av number of packets	Av number of flows	Av flow size (pkt)
DDoS improved GET(Goldeneye)	452s	6084	864	7
DDoS GET(ddossim)	138s	46081	22103	2
DDoS GET(hulk)	546s	8482	1085	8

The average rate of normal traffic in the UNB-ISCX dataset is 69 packets per second [14]. In addition to this dataset, the ISOT dataset is also used by the Internet of Things and Network Security Research Center at the University of Victoria. The captured ISOT dataset consists of a pcap file that is approximately 14.1 GB in size[15]. In this research, the normal part of this dataset is used. The details of the introduced attacks in the UNB-ISCX dataset are represented in Table 4.

5. Implementation Environment

The experiments were performed on an ASUS laptop equipped with an AMD (Bristl Ridge), FX-9830P CPU 2.8GHz processor and 12GB of RAM. The chosen operating system was Linux Ubuntu 14.04 LTS, which was executed on a Window 8.1 host machine. The network controller selected for the experiments was Floodlight [16], which relied on Mininet2.2.1 [17] for network simulation. This investigation illustrates the assessment findings of the proposed technique for identifying Distributed Denial of Service (DDoS) attacks. The training and assessment segment are chosen employing the K-

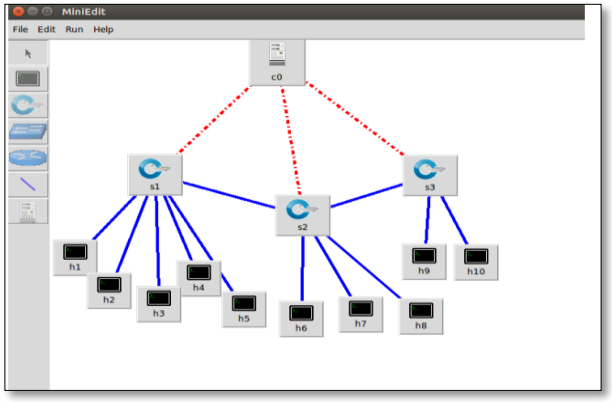

Figure 5.Simulated Topology

This topology is implemented using Mininet. It creates a network topology having 10 hosts connected with three OpenFlow-enabled switchs. All switches are interconnected with each other and are in turn connected with OpenFlow controller as shown. The controller must first collect the information needed from switches, hosts, and existing communications to detect the contraband attacks in order to be able to detect them. Distributed blocking attacks can be identified using different traffic features. The attributes of the headers are extracted using the toolbar. These features are collected every few seconds and the relevant features are calculated using this extracted traffic statistic. Figure 6 shows wireshark software that can be used to view and collect network information efficacy of the suggested resolution can be gauged through the utilization of the F-measure and precision parameters, as well as the accuracy parameter [18].

Fold approach, wherein the number of folds is regarded as ten in this particular investigation. The

6. Experiments

In this section, a clear implementation of the flow-based hybrid approach is presented. Figure 5 shows an overview of the implementation topology and simulation of the attack on this network.

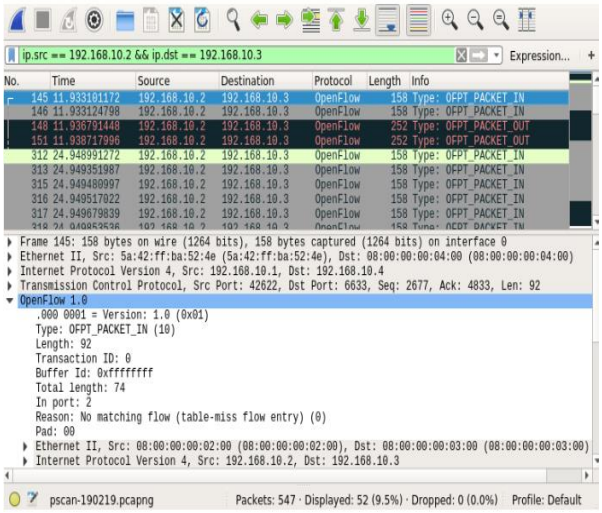

Figure 6.Flow analysis in Wireshark

In this Figure, using the OF filter can open packets and streams of SDN and analyze the statistical results.After the information extracted from the data set is sent to the controller, the Floodlight controller uses the written module to detect and detect the DDoS attack. Figure 7, part of this module, written in the Eclipse Neon software environment, is written in the Floodlight controller using the Java language.

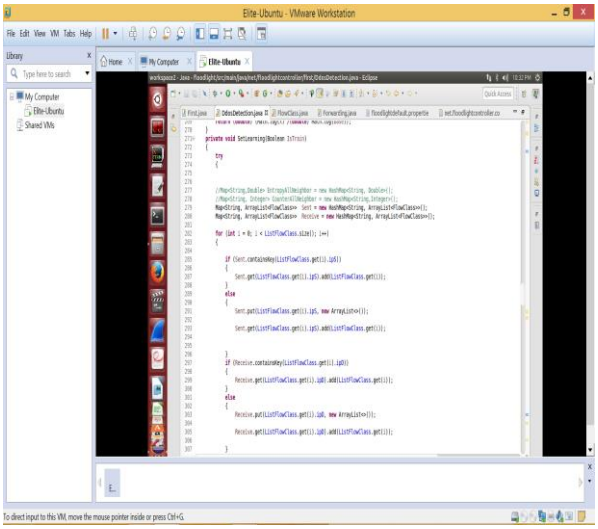

Figure 7. DDoS Detection Module in Floodlight

The implementation code written in Figure 7 shows part of the controller module for detecting attacks on the Floodlight controller. After the results of the statistical methods have been obtained, part of the results are sent to the machine learning section. This section deals with attack detection using classification algorithms in Software.

7. Results

Here, the results of proposed methods are presented for the intended dataset to identify DDoS attacks. For each dataset, the results are presented in two parts. First, the results are presented using the statistical method, and then the results of machine learning are demonstrated.

7.1. Results of Statistical Method

In this section, the results are evaluated in three stages for Hartley entropy, Shannon entropy and collision entropy, considering α=0, α=1 and α=2 values. The time periods are considered between 10 - 240 seconds, then among these periods of time, 15-45 seconds were selected as the best ones, because high level of attacks were detected during these periods, and also were detected earlier. Now, for each selected time, various w and m as tolerance-factors are evaluated. Since these variables are experimental parameters and play a decisive role in two threshold relations, the common values of -1, 0, and 1 are considered for three stages, and the detection results are obtained for each different state. In this table, for each stage of different tolerance-factors, the first mode in which TPR is 100 is considered, which is the case in all attacks in the dataset are correctly identified. Comparing this particular state with different tolerance-factors, the state where the FPR has the lowest value is selected as the chosen time period and its tolerance-factor value is set as the best value. The results of analysis for three Hartley, Shannon and collision entropies are presented in Tables 5, 6 and 7, respectively.

Table 5.Testing the accuracy of attack detection for UNB-ISCX dataset in $\alpha = 0$ For Hartly entropy

TF	TP	H_C	H_N	ID_n	T_1	TPR	FPR	AC
-١	١٥	٢,١٠	١,٨٨	٠,٢٢	1.76	81.45	11.34	**88.01**
	٢٠	٢,١٥	١,٩٤	٠,٢١	2.05	95.65	47.76	**56.15**
	٣٥	٢,٣٢	١,٩٨	٠,٣٤	2.12	100	89.37	**16.50**
	٤٥	٢,٥٧	٢,٢١	٠,٣٦	2.52	١٠٠	90.70	**15.96**
٠	١٥	٢,٢2	٢,١2	٠,١٠	٢,١١	84.18	47.75	**56.11**
	٢٠	٢.25	.٢17	٠,٠8	٢,٠٩	71.56	24.19	**75.63**
	٣٥	.٢30	.٢21	٠,٠٩	٢,١٧	86.04	53.52	**48.24**
	٤٥	٢,٣٩	٢,٢٤	٠,١٥	2.29	100	47.91	**55.01**
١	١٥	٢,١٥	٢,١١	٠,٠٤	2.10	٦٩,٦٤	١٤,٤٦	**٨٤,٣١**
	٢٠	٢,٢٩	٢,١٤	٠,١٥	2.19	٨٣,٥٣	١٧,٢٩	**٨٢,٧٥**
	٣٥	.٢31	٢,١٩	٠,١٢	2.13	٩٣,١٩	٤٩,٦٢	**٥٣,٦٥**
	٤٥	٢,٤٥	٢,٢٠	٠,٢٥	2.27	100	50.60	**52.48**

The less FPR value is obtained compared to other states by examining the results in the first mode for α=0 in Hartley entropy state and different tolerance-factors in different time periods in tolerance -factor = 1 when TPR = 100. Thus, the tolerance-factor = 1 in this mode, and accordingly the threshold value is calculated. In this case, by examining the results, it is observed that no good results are achieved, due to the α-Entropy value obtained for the current state and normal data. As observed, this value in the time period when both attacks and normal-flow are present is very close to the existing dataset, and the ID_n value is very small and will not distinguish the periods that the attacks are present. Therefore, the detection of DDoS attack traffic is difficult using α-Entropy and α=0 at first levels. Now, considering other values for α, we examine different values of α-Entropy. The results for α=1 andα=2 for Shannon and Collision entropy are presented in Tables 6 and 7.

Table 6.Testing the accuracy of attack detection for UNB-ISCX dataset in α=1 For Shannon entropy

TF	TP	H_C	H_N	ID_n	T_1	TPR	FPR	AC
-1	15	2.80	1.91	0.89	2.01	67.44	3.48	**93.55**
	20	2.87	1.95	0.92	2.56	86.95	6.68	**92.66**
	35	3.00	2.05	0.95	2.65	95.65	47.76	**56.15**
	45	3.06	2.15	0.91	2.78	100	38.83	**64.62**
0	15	2.94	1.99	0.95	2.60	71.56	24.19	**75.63**
	20	2.99	2.10	0.89	2.72	84.08	51.25	**50.90**
	35	3.04	2.17	0.87	2.29	98.09	51.88	**50.82**
	45	3.17	2.54	0.63	2.79	100	43.31	**59.87**
1	15	3.01	2.19	0.82	2.35	55.74	17.36	**80.99**
	20	3.14	2.11	1.03	2.57	65.69	23.23	**76.27**
	35	3.17	2.18	0.99	1.02	83.69	5.49	**93.45**
	45	3.21	2.13	1.08	2.19	100	19.10	**81.53**

Table 7. Testing the accuracy of attack detection for UNB-ISCX dataset in α=2 For Collision entropy

TF	TP	H_C	H_N	ID_n	T_1	TPR	FPR	AC
-1	15	3.05	1.90	1.15	2.01	70.00	3.17	**93.15**
	20	3.42	2.01	1.41	3.20	76.47	2.36	**95.13**
	35	4.06	2.16	1.90	3.26	81.01	26.55	**73.73**
	45	4.13	2.50	1.63	2.65	100	51.16	**51.65**
0	15	3.14	2.11	1.03	2.29	83.96	7.11	**92.05**
	20	3.23	2.22	1.01	2.76	87.87	25.79	**74.69**
	35	3.27	2.29	0.98	2.87	98.48	55.40	**46.49**
	45	3.32	2.30	1.02	2.76	100	35.57	**66.59**
1	15	2.85	2.01	0.84	2.67	66.44	15.46	**82.76**
	20	3.24	2.10	1.14	2.85	88.23	3.05	**96.00**
	35	3.35	2.27	1.08	3.02	91.17	23.54	**76.97**
	45	3.47	2.32	1.15	2.67	100	6.11	**94.42**

By examining the results in Tables 6 and 7 for different values of α=1and α=2, it is observed that the ID_n value in these two modes is higher than that of the state with α=0, indicating that with the increase of α the distance between the normal and attack traffic increases, makes the attack and normal traffic more distinguished than each other. The results are illustrated in Figures 8 -10.

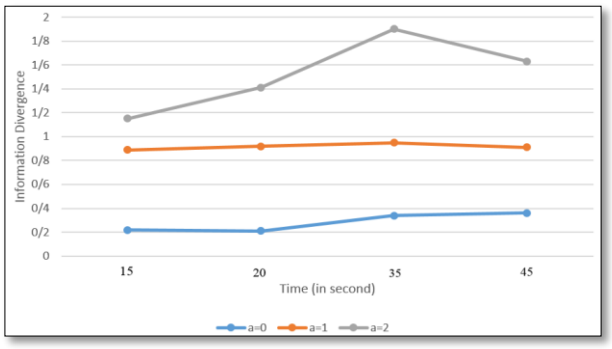

Figure 8. Comparison of ID_n for different modes of α in TF = -1

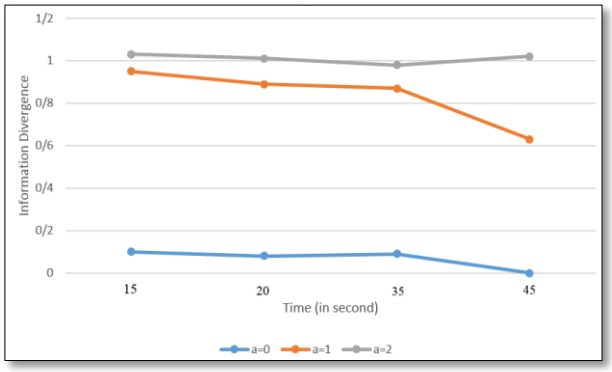

Figure 9. Comparison of IDn for different modes of α in TF = 0

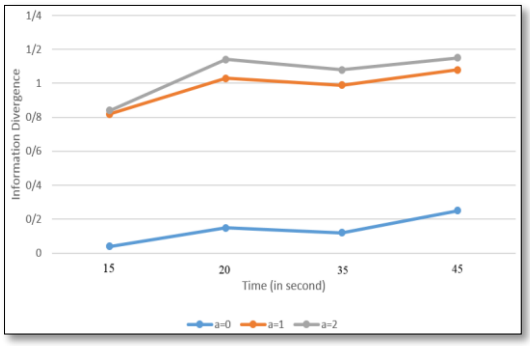

Figure 10. Comparison of ID_n for different modes of α in TF = 1

Based on the results in Figures 8-10, IDn in α=2 state is higher than the other states and is chosen as the selected α .Now, by examining the TPR and FPR values in this state, among different values, the TF =1 is chosen for the threshold relation because the FPR value is the least in this case. In general, by examining the obtained results in the statistical section, the selected state among the three modes is collision entropy with α=2 and TF = 1. In this case, FPR = 6.11% and TPR=100%. Unfortunately, these results indicate that the FPR value is high although TPR is high in this technique. Therefore, in order to eliminate this issue and refine the data and conduct pre-processing, the percentage of attacks alerted in this case are sent to the machine learning section and among different checked algorithms as shown in Table 8.

Model	Decisive Component	Optimized Parameter	TPR	FPR	AC	Precision
KNN	Euclidean distance	K=11	96.22	0.58	99.32	83.48
RF	Bagged Decision Tree	N_est=100	97.41	0.62	98.39	99.38
SVM	RBF Kernel	C=100.0 Gamma=1.0	99.68	0.10	99.84	99.66

As indicated in Table 8, the examined results for a variety of Models are discussed. Regarding the UNB-ISCX dataset, SVM algorithm is the best algorithm to detect DDoS attacks with a precision of 99.84% and a FPR value of 0.10%.

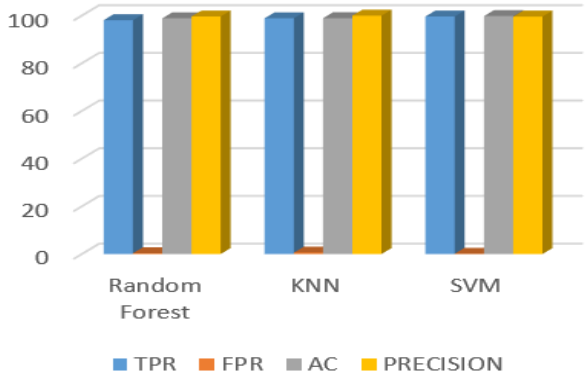

Figure 11. Results of Models for DDoS attacks Detection

As shown in Figure 11, the SVM algorithm has a better performance, compared to other discussed algorithms in machine learning method to detect DDoS attacks, which involves a higher accuracy.

8. Comparing the Method Presented In This Study To Other Methods

This section compares the proposed method in this study to other methods used in other studies. It should be noted that all these studies were conducted to detect DDoS attacks in the UNB-ISCX dataset being also explored in this study. The results on this dataset are summarized in Table 9 and then compared in the Figure 8.

Table 9. Comparing the accuracy of DDoS detection with different methods for the UNB-ISCX dataset

Detection Methods	Ref	AC	FPR
Computer Vision Technique	[19]	90.12	7.92
Cloud Computing	[20]	89.30	Not mentioned
K-Means+NBC	[21]	99	2.2
Neural Network	[22]	98	Not mentioned
Machine Learning Techniques	[23]	81.80	8.2
Ripper + C5.0	[24]	99	2
This Research		99.84	0.10

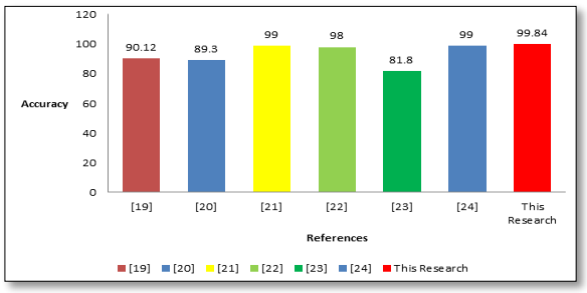

Figure12.Comparing the accuracy of the proposed method to other studies for the UNB-ISCX dataset

The results presented in Table 9 and Figure 12 demonstrate that the accuracy of the proposed method in this study surpasses that of comparable tasks.

9. Conclusion

In this paper, a novel approach was introduced to enhance the precision of identifying Distributed Denial of Service (DDoS) attacks on Software-Defined Networking (SDN) networks. This innovative technique encompasses a combination of statistical analysis and machine learning methodology.In the statistical portion, the three forms of α-Entropy, namely Hartly entropy, Shannon entropy, and Collision entropy, are derived from the examination of the destination IP. Subsequently, the IDn value is computed. By investigating three distinct values for α and examining the outcomes in 2=α for Collision entropy, a greater magnitude of true positive rate (TPR) and a reduced magnitude of false positive rate (FPR) are achieved. Consequently, α=2 is chosen from the scrutinized states. Subsequently, the TF parameter values are appraised to calibrate the desired threshold in this manuscript.A value is chosen that, for the initial instance, completely detected the regular traffic, and its True Positive Rate (TPR) should be 100 while simultaneously possessing the lowest False Positive Rate (FPR). Moreover, at this stage, a True Flag (TF) of 1 is opted for. Subsequently, the proportion of the dataset identified as an attack is transmitted to the machine learning phase in order to enhance the precision of detection.Among the techniques used in the machine learning section, SVM, RF, and KNN, the SVM technique, which has an accuracy rate of 99.84% and a false positive rate (FPR) of 0.10%, demonstrates the highest accuracy in detecting DDoS attacks.

References

[1] Yan, Q., et al., software defined networking (SDN) and Distributed DDenial of service (DDOS) Attacks in Cloud Computing Environments:A Survey ,Some Research Issue ,and Challenges. IEEE Communications Surveys and Tutorials, 2015.

[2] Lim, S., et al., A SDN-Oriented DDOS Blocking Scheme for Botnet-Based Attacks, in ICUFN 2014.

[3] Sahay, R., et al., Towards Autonomic DDoS Mitigation using SDN, in SENT 2015:NDSS Workshop on Security of Emerging Networking Technologies. 2015: United States.

[4] Buczak, A. and E. Guven, A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys and Tutorials, 2016. **18**(2): p. 1153-1176.

[5] YAN, Q., Q. GONG, and F. DENG, Detection of DDOS Attacks Against Wireless SDN Controllers Based on the Fuzzy Synthetic Evaluation Decision-making Model. Ad Hoc & Sensor Wireless Networks, 2016. **33**: p. 275-299.

[6] Todorova, M.S. and S.T. Todorova, DDoS Attack Detection in SDN-based VANET Architectures. 2016, AALBORG.

[7] Kia, M., Early Detection and Mitigation of DDoS In Software Defined Networks. 2015, Ryerson: Toronto,Ontario,Canada.

[8] Mousavi, S.M., Early Detection of DDoS Attacks in SDNController. 2014, Carleton: Ottawa,Ontario.

[9] Bolly, F. and I. Gentil, Phi-entropy inequalities for diffusion semigroups. 2018: Universit´e Paris-Dauphine, Ceremade, UMR CNRS 7534.

[10] Dhawan, M., et al., SPHINX:Detecting Security Attacks in Software Defined Networks, in Network and Distributed System Security Symposium. 2015.

[11] Hoque, N., H. Kashyap, and D. Bhattacharyya, Real-time DDoS attack detection using FPGA. Computer Communications, 2017.

[12] Yadav, A., et al., SDN Control Plan Security in Cloud Computing Against DDoS Attack. IJARIIE, 2016. **2**(3): p. 426-430.

[13] YANG, M. and R. WANG, DDoS detection based on wavelet kernel support vector machine. The Journal of China Universities of Posts and Telecommunications. **15**(3): p. 59-94.

[14] Hadian Jazi, H., H. Gonzalez, and N. Stakhanova, Detecting HTTP-based Application Layer DoS attacks on Web Servers in the presence of sampling. computer Networks, 2016.

[15] Bhamare, D., et al., Feasibility of Supervised Machine Learning for Cloud Security. IEEE, 2016.

[16] Asadollahi, S. and B. Goswami, Experimenting with scalability of floodlight controller in software defined networks, in International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT). 2017.

[17] Mininet ,An Instant Virtual Network on your Laptop (or other PC). Available from: http://mininet.org/.

[18] Cross-validation (statistics). Available: https://en.wikipedia.org/wiki/Cross-validatikon_(statistics).

[19] Tan, Z., et al., Detection of Denial-of-Service Attacks Based on Computer Vision Techniques. IEEE TRANSACTIONS ON COMPUTERS, 2013.

[20] Wang, B., et al., DDOS Attack Protection in the Era of Cloud Computing and Software -Defined Networking, in Network Protocols (ICNP), 2014 IEEE 22nd International Conference 2014.

[21] YASSIN, W., et al., ANOMALY-BASED INTRUSION DETECTION THROUGH KMEANS CLUSTERING AND NAIVES BAYES CLASSIFICATION, in 4th International Conference on Computing and Informatics, ICOCI. 2013.

[22] Saied, A., R. Overill, and T. Radzik, Detection of known and unknown DDOS attacks using Artifitial Neural Networks. Neurocomputing, 2015.

[23] Catania, C. and C. Garcia Garino, Towards Reducing Human Effort in Network Intrusion Detection, in The 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. 2013: Berlin, Germany.

[24] Fallahi, N., A. Sami, and M. Tajbakhsh, Automated Flow-based Rule Generation for Network Intrusion Detection Systems, in 24th Iranian Conference on Electrical Engineering (ICEE). 2016.