DESINGING INTRUSION DETECTION SYSTEM IN SOFTWARE DEFINED NETWORKS USING HYBRID GWO-AE-RF MODEL

Ancy Sherin Jose

Research Scholar, Division of Computer Science and Engineering, School of Engineering, Cochin University of Science and Technology,
Kochi 682022, Kerala, India
ancysherin2021research@gmail.com

Latha R Nair

Associate Professor, Division of Computer Science and Engineering, School of Engineering, Cochin University of Science and Technology,
Kochi 682022, Kerala, India
latharnair@cusat.ac.in

Varghese Paul

Professor, Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology
Kochi 682039, Kerala, India
vp.itcusat@gmail.com

Abstract

Software Defined Networks is a promising networking solution which mitigates the limitations of traditional networks. The presence of logically centralized controller enables global view of the network in SDN. The provision for configuration and management of networking devices with high-level programming languages helps for adding proactive attack detection and mitigation strategies. However, SDN is prone to several evolving network attacks. Malicious traffic from botnets disrupts the network services and causes financial and reputational damages to individuals as well as enterprises. Intrusion detection systems aim to safeguard the network from vulnerabilities by detecting them instantaneously. Machine Learning based intrusion detection systems are used in traditional networks and are found very effective. This paper aims to build an Intrusion Detection System for Software Defined Networks leveraging machine learning and deep learning techniques. In order to build a model with reduced space and time complexity, feature selection and dimensionality reduction techniques were used. The feature selection using Grey Wolf Optimizer and dimensionality reduction with Autoencoder (GWO -AE) are incorporated in the study. The work is evaluated on the latest public SDN dataset – InSDN. Multiclass classification using Random Forest classifier with the reduced feature space obtained from GWO - AE gave weighted F1 score of 98.95%.

***Keywords*: Multiclass classification; Grey Wolf Optimizer; Autoencoder; Random Forest**

1. Introduction

Software Defined Networks (SDN) is a revolutionary networking paradigm which decouples the data plane from control plane. This loosely coupled model has the network brain embedded in the controller which serves as the Network Operating System. In SDN, the logically centralized controller helps for the global view of network and serves as the single point for configuration and management of networking devices [Kreutz *et al.*, (2015)]. The limited responsibility of networking devices in SDN is to forward the traffic according to the flow-based decisions inserted by the controller in the SDN switch. The decoupled architecture hence enriches SDN with many capabilities. The individual configurations of devices and programming the devices with vendor specific languages is no longer needed in SDN [Hu *et al.*, (2014)] [Kim and Feamster, (2013)]. Further the capability of building flow-based solutions on the top of SDN controller allow for the development of applications in order to cater various business requirements. This facility can be utilized for building Network Intrusion Detection Systems (NIDS) in SDN environments. The capability of SDN architecture to add proactive and reactive flow rules in networking devices can mitigate the network level intrusions instantaneously when such cases are detected.

Even though SDN mitigates many limitations faced by traditional networks, the entire network can be paralyzed by various attacks in SDN environment. The attack vectors of SDN are different from traditional networks [Elsayed *et al.*, (2020)]. The attacks on the data plane devices can compromise the network switch and can gain access to launch further attacks against controller. The virtualized switches in SDN are easy to be compromised by the attacker who can utilize the understanding of flow table rules to host sophisticated attacks. The communication channel between the SDN controller and the switches can be congested by launching flooding attacks. Further attacks against the controller can paralyze the entire SDN network. Also, the application plane of SDN architecture can be exploited for executing malicious code by attackers, an example is Smart City application [Bawany and Shamsi, (2016)]. Attacks against all the components of SDN architecture namely data plane devices, the controller, communication link between the controller and data plane devices and the application plane altogether pose serious security threats. Hence it is critical to safeguard the SDN from these network vulnerabilities.

Many solutions have been proposed by researchers and industry to protect SDN networks. These include architecture level changes like configuring distributed controllers to avoid single point failures as well as machine learning based solutions for intrusion detection [Nunes *et al.*, (2014)] [Jose *et al.*, (2021)]. Machine learning based Network Intrusion Detection Systems (NIDS) are found very effective in traditional networks. These intrusion detection systems can be either Signature Based Intrusion Detection Systems or Anomaly Based Intrusion Detection Systems. Signature based systems work by learning from the known attack pattern or signatures. These systems are capable of detecting the attacks whose pattern is already known. However, these systems fail to detect the presence of newly evolved attacks which are also called zero-day attacks. Anomaly based Intrusion detection systems are capable of detecting such attacks, as they work by detecting the traffic which deviates from the normal traffic.

This work aims to build machine learning based intrusion detection system over Software Defined Networks for classifying the network traffic into normal or a specific type of attack. Multiclass classification with tree based classifier – Random Forest is attempted in this work. Most discriminating features have been selected with a metaheuristic wrapper based feature selection technique namely Grey Wolf Optimization. Autoencoder based dimensionality reduction is also attempted to provide the compressed representation of feature space. This work uses the recently published InSDN dataset that is created in SDN environment. The arrangement of the paper is as follows. Section 2 describes about the contributions of the study. Section 3 briefly describes the background concepts. Section 4 consolidates the literature in the domain. Section 5 presents the proposed method. Section 6 discusses the results of the study. Section 7 is the conclusion.

2. Contributions of the study

(1) In this work, we attempt to build Network Intrusion Detection System (NIDS) for SDN utilizing machine learning and deep learning techniques. Instead of using traditional network based datasets, this work uses the recently published SDN dataset - InSDN. Multiclass classification to classify seven different attacks along with normal traffic is attempted in the study.
(2) InSDN dataset is an imbalanced dataset with 4 majority classes and 4 minority classes. Several sampling techniques like SMOTE, SMOTE-ENN were attempted to handle the class imbalance. It was derived that random oversampling of minority classes followed by random under sampling of majority classes were providing improved classification performance.
(3) In this work feature selection with metaheuristic Grey Wolf Optimization technique was leveraged for obtaining the best feature subset. Dimensionality reduction with autoencoder was further attempted on the feature subset to obtain the compressed representation. Multiclass classification with Random Forest classifier was performed which classified the traffic into seven attack classes along with normal traffic. The hybrid GWO – AE – RF achieves weighted F1 score of 98.95% with the reduced feature space.
(4) Evaluation of multiclass classification with features selected through three techniques – feature selection through Correlation Based Feature Selection (CBFS), Tree Based Feature Selection (TBFS) and Autoencoder (AE) based dimensionality reduction (AE) were also performed. It was analyzed that the hybrid GWO – AE – RF model obtained superior performance.

3. Background Concepts

3.1. *Software Defined Networks*

SDN offers a dynamic network control architecture where the controller is separated from the forwarding devices [Myint *et al.*, (2019)]. SDN architecture presented in Fig.1. constitutes of three layers namely – infrastructure layer or data plane, control layer and an application layer. Network devices like switches and routers form the infrastructure layer. In SDN environments, switches only forward the traffic according to the flow table rules installed in them. The controller which is the network brain resides in the control layer and governs the forwarding decisions. These decisions are configured in the switches with southbound protocol. The commonly used southbound protocol is OpenFlow which is created by Open Networking Foundation (ONF) [ONF Software-Defined Networking, (2012)]. The power of SDN resides in the centralized controller intelligence. The communication between controller and infrastructure layer happens through secure communication channel with TLS/SSL security [Agborubere and Sanchez Velazquez, (2017)]. Various network monitoring applications can be built on the top of the controller in the application layer. SDN allows building network applications with high level programming languages [ONF Software-Defined Networking, (2012)]. These applications can be built on the top of the controller or they can communicate with the controller with northbound interfaces. Controller plays an important role in interconnecting the network applications with the infrastructure layer. SDN architecture enables simplified network management and introduces vendor independent nature in infrastructure layer with network programmability [Javeed *et al.*, (2021)]. It allows for extending many networks in the data plane like Internet of Things (IOT), vehicular networks etc. [Al-Rubaye *et al.*, (2019)] [Narayanadoss *et al.*, (2019)] [Yang *et al.*, (2022)].

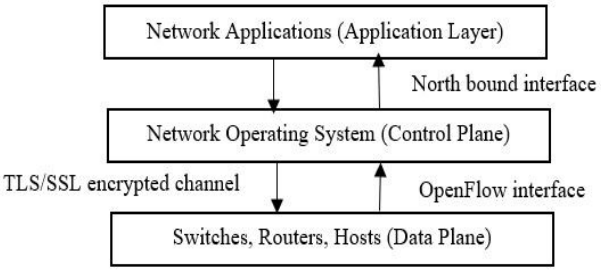

Network Applications (Application Layer)
North bound interface
Network Operating System (Control Plane)
TLS/SSL encrypted channel
OpenFlow interface
Switches, Routers, Hosts (Data Plane)

Fig.1. SDN Architecture

The communication between the controller and switches is an inevitable part of SDN. SDN follows flow-based architecture. The IP packets which have the same five tuples {source IP address, destination IP address, source port, destination port, protocol} constitute the flow [Xiao and Peng, (2016)]. The switches in SDN consists of a pipeline of flow tables. The flow table architecture is depicted in Fig. 2.

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie

Fig. 2. Flow table Structure

Flow table rules dictate on the actions that has to be carried out for forwarding the packets. The flow rule consists of the match fields which actually filters the packet based on certain match conditions. Header fields of flow entry is depicted in Table 1.

Ingress Port	Ether Source	Ether Destination	Ether Type	Vlan Id	Vlan Priority	IP Src	Ip Dst	IP Protocol	IP ToS bits	Port Src	Port Dst

Table 1. Header Fields of Flow Entry [Braga *et al.* (2010)]

Each flow rule is associated with an action which includes forwarding the packet to the particular port, flooding the packet or dropping the packet [Nunes *et al.*, (2014)]. The default flow rule of an SDN switch is a table miss entry, assigned with a priority of zero which forwards the packet to the controller as PACKET_IN message. The controller makes the forwarding decisions and sends the PACKET_OUT message which incorporates the actions to be performed on the packet. The controller sends flow modification messages to the switches for adding, deleting or modifying the flow rules [Alamri and Thayananthan, (2020)]. Controller can be often burdened with the overwhelmed PACKET_IN messages coming to it, which in turn increases the communication delay.

3.2. *Metaheuristic Feature Selection with GWO*

The aim of feature selection is to select the relevant subset of features from the dataset based on certain selection criterion for reducing the time and space complexity of the machine learning model resulting in reduced training time [Harris *et al.*, (2020)]. Feature selection techniques are classified as filter techniques, wrapper techniques and embedded techniques. Filter based techniques focus on the general characteristics of the data [Xu *et al.*, (2010)]. These techniques use ranking methods as the key criteria for selecting the relevant features. Ranking methods place a threshold to score the variables and those variables whose score is less than the threshold is eliminated [Chandrashekar and Sahin, (2014)]. Information gain, Correlation based techniques are examples of filter methods.

In wrapper methods, predictor performance is used to evaluate the variable subset. Here predictor performance is the objective function for evaluation. The search strategy of the wrapper based techniques are categorized into three which are exponential, sequential and randomized selection strategy [Agrawal P *et al.*, (2021)] [Jovic *et al.*, (2015)]. In the exponential search method, the number of features evaluated increases exponentially with the size of features. Due to the high computational cost these techniques are not practically possible. Exhaustive search, branch and bound methods are the examples of this technique [Agrawal P *et al.*, (2021)]. Sequential algorithms start with the complete set of features or empty set of features and will sequentially add or remove the features to maximize the objective function. But after including a feature or removing a feature, it cannot be changed which can result in selecting the local optima. Forward feature selection, backward feature selection, linear forward selection are some examples of sequential methods. Randomized methods evaluate different subsets which are generated by searching the search space randomly, there by not getting trapped in local optima. They are population based methods like simulated annealing, metaheuristic methods etc. In Embedded techniques feature selection is part of the training process of the classifier.

GWO is the population-based metaheuristic method which is developed based on the hunting or chasing characteristics of wolves. This bioinspired technique was developed by Mirjalili *et al.* in 2014 [Mirjalili *et al.*, (2014)]. Alpha, beta, delta and omega forms the four categories of wolves in the grey wolf community. Among these four categories, alpha wolves are the leaders of the group, while beta wolves assist alpha in making decisions. Beta wolves are considered as the eligible candidates to alpha wolves when alpha candidates retire or dies during hunting. Delta wolves are the elder wolves which protects the boundaries. Omega wolves are the last category, who has to obey the other three categories of dominant wolves. The hunting nature of wolves can be modelled mathematically, where each wolf is searching for the solution in the search space. The wolves surround the prey in order to chase it. The encircling behavior of the wolves are mathematically represented from "Eq. (1)" to "Eq. (4)" [Mirjalili *et al.*, (2014)].

$$\vec{D} = \left|\vec{C}\ \vec{X}_p\,(t) - \vec{X}\,(t)\right| \qquad (1)$$

$$\vec{X}\,(t+1) = \vec{X}_p(t) + \vec{A}\ \vec{D} \qquad (2)$$

where $\vec{X}$ is the grey wolf position vector, and $\vec{X}_p$ indicates the prey position, $\vec{A}$, $\vec{C}$ are the coefficient vectors, and *t* represents the iteration number.
Vectors $\vec{A}$ and $\vec{C}$ are calculated with "Eq. (3)" and "Eq. (4)".

$$\vec{A} = 2a.\vec{r}_1 - a \qquad (3)$$
$$\vec{C} = 2\vec{r}_2 \qquad (4)$$

Here *a* is diminished from 2 to 0 linearly, $\vec{r_1}, \vec{r_2}$ are random vectors in [0,1]. Alpha guides the chase. The position of the wolves is estimated with respect to the fitness function which is defined based on the problem. First best solution is represented as α, and the second best solution is represented by β, and the third by δ based on the fitness value. In the search to find the best solution, wolves update their positions which are represented mathematically in equations below.

$$\vec{X}\,(t+1) = \frac{\vec{X_1}+\vec{X_2}+\vec{X_3}}{3} \qquad (5)$$

$$\vec{X_1} = \vec{X_\alpha} - \vec{A_1}\,.(\vec{D_\alpha}\,) \qquad (6)$$

$$\vec{X_2} = \vec{X_\beta} - \vec{A_2}\,.(\vec{D_\beta}\,) \qquad (7)$$

$$\vec{X_3} = \vec{X_\delta} - \vec{A_3}\,.(\vec{D_\delta}\,) \qquad (8)$$

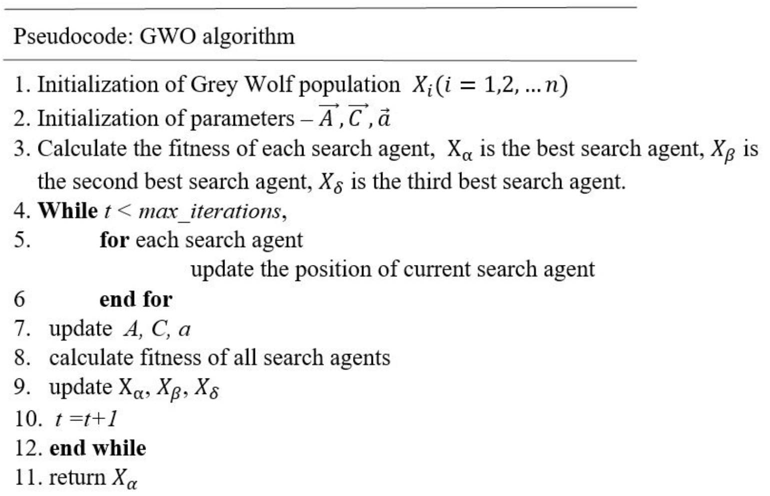
$X_\alpha, X_\beta, X_\delta$ are the three best solutions in the given iteration *t*. $D_\alpha, D_\beta, D_\delta$ are represented in "Eq. (9)" to "Eq. (11)".

$$\left|\vec{D_\alpha} = \vec{C_1}.\vec{X_\alpha} - \vec{X}\right| \qquad (9)$$

$$\left|\vec{D_\beta} = \vec{C_2}.\vec{X_\beta} - \vec{X}\,\right| \qquad (10)$$

$$\left|\vec{D_\delta} = \vec{C_3}.\vec{X_\delta} - \vec{X}\,\right| \qquad (11)$$

The pseudo code for the algorithm is as below [Mirjalili *et al.*, (2014)].



Pseudocode: GWO algorithm

```
Initialization of Grey Wolf population $X_i (i = 1,2, ... n)$
Initialization of parameters – $\vec{A}, \vec{C}, \vec{a}$
Calculate the fitness of each search agent, $X_\alpha$ is the best search agent, $X_\beta$ is
  the second best search agent, $X_\delta$ is the third best search agent.
While $t < max\_iterations$,
        for each search agent
                update the position of current search agent
        end for
update $A, C, a$
calculate fitness of all search agents
update $X_\alpha, X_\beta, X_\delta$
$t = t+1$
end while
return $X_\alpha$
```

3.3 Autoencoder based Dimensionality Reduction

Autoencoder is a feedforward multilayer neural network which reconstructs the input after the training. It is also called a replicator neural network. Two nonlinear mapping in Autoencoder are encoder and decoder network. If $X = \{x_1, x_2, \ldots x_n\}$ represents the training dataset with n samples, and x_i is a d-dimensional feature vector, the aim of encoder is to map the input vector x_i to a compressed representation or lower dimensional representation c_i in the latent space such that $c_i = f_\theta(x_i) = s(Wx_i + b)$ where W is a $d' \times d$, and d' represents number of hidden units, b is a bias vector, and θ is a mapping parameter such that $\theta = \{W, b\}$ and s represents an activation function [Farahnakian and Heikkonen, (2018)] [Schreyer *et al.*, (2017)]. The commonly used activation functions are rectified linear function, softmax, tanh, sigmoid, linear etc. Sigmoid activation function is as given in "Eq. (12)".

$$s(t) = \frac{1}{1+exp^{-t}} \qquad (12)$$

The decoder$g_\theta(.)$ then maps c_i to reconstructed d dimensional vector $\hat{x}_i$, $\hat{x}_i = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4 \ldots \hat{x}_n\}$ of the original input space, such that $\hat{x}_i = g_{\theta'}(x_i) = s(W'c_i + b')$ where $W' = d' \times d$, b is a bias vector and $\theta' = \{W', b'\}$. Reconstruction error refers to the difference between original input and the reconstructed input. Autoencoder tries to achieve $x_i \approx \hat{x_\imath}$, for the reason that it is trained to learn a set of optimal encoder decoder model parameters θ^*, that minimizes difference between the input x_i and the reconstructed input $\hat{x}_i = g_\theta(f_\theta(x_i))$, as conscientiously as possible. In essence, autoencoder is trying to learn the identity function of the original data distribution, by mapping the original data set X to the low dimensional feature space c_i which is capable of recovering $\hat{x}_i$ from the projected low dimensional space [Mirsky *et al.*, (2018)]. Thus, the compressed representation $\hat{c}_i = \{\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4 \ldots \hat{c}_n\}$ is considered as the new training set. The classifier training is then accomplished with the newly obtained lower dimensional training set. The objective of an autoencoder [Schreyer *et al.*, (2017)] is to develop a model that optimize

$$\arg\min_\theta ||X - g_\theta(f_\theta(X)|| \qquad (13)$$

for all the entries in the dataset. The loss function L_θ, which is the cross entropy loss used in the multiclass classification is given in "Eq. (14)".

$$L_\theta(x_i, \hat{x}_i) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} x_j^i \ln(\widehat{x_j^i}) + (1 - x_j^i)\ln(1 - \widehat{x_j^i}) \qquad (14)$$

where x_i represents the entries in the dataset and $\hat{x}_i$ represents the reconstructed input. There are several works that used autoencoders in intrusion detection domain [Al-Qatf *et al.*, (2018)] [Javaid *et al.*, (2016)] [Min *et al.*, (2021)] [Zhou and Paffenroth, (2017)].

3.3. *Random Forest Classifier*

Random Forest classifier was developed by Breiman in 2001 is an ensemble technique which combines many decision trees. The variance of a single tree is reduced by averaging the prediction of multiple trees and hence improved performance is achieved. Training is done with the random samples taken from the original dataset which is random bootstrapping (Bagging). In order to reduce the correlation among the trees grown, random subset of predictors is used for the split condition [Hastie, *et al.*, (2001)] [Pham *et al.*, (2017)]. The trees grown in the Random Forest are built independent of each other. The predictive accuracy of the bagged model is measured with Out OF Bag error (OOB error). In each bootstrap iteration, certain samples are kept aside and not used for fitting the model, which are called OOB samples and they are used for evaluating the performance of the model in that iteration. For N such bootstrap samples, N performance measures can be derived. The OOB error of the ensemble is the average of the N performances [Hu *et al.*, (2020)].

4. Related Works

This section describes the recent works in literature for detecting intrusions in SDN environments. Elsayed et al proposed hybrid CNN and LSTM model on InSDN dataset for detecting attacks in SDN environment [Abdallah *et al.*, (2021)]. In the work, they used L2 regularization and dropout method to overcome the overfitting problems. This work achieved an overall accuracy of 96.32%. Matthew Banton et al attempted CNN based classification on UNSW-NB15 dataset [Banton *et al.*, (2020)]. SMOTE and ADASYN based sampling techniques were used to balance the data. The work used 28 features from the dataset in order to suit for the SDN environment. Principal Component Analysis (PCA) with 97% variance was used for dimensionality reduction. This work achieved a detection accuracy of 93.3%. Myo Myint *et al.* used Advanced SVM based multiclass classification on the emulated SDN dataset. In this work, they used volume based and asymmetry based features for detecting DDoS attacks. With the five features, the system was able to achieve a detection accuracy of 97% [Myint *et al.*, (2019)]. Stacked Auto Encoder Multilayer Perceptron was used by Nisha Ahuja et al. for classifying TCP, UDP and ICMP flooding attacks over SDN DDoS dataset. SAE-MLP model performed even better than CNN-LSTM model in classification with an accuracy of 99.75% [Ahuja *et al.*, (2021)].

Georgi *et al.* attempted machine learning based classification with five classifiers namely – Support Vector Machine, Decision Tree, Bagged Tree, Random Forest and k Nearest Neighbors for classifying the normal traffic from three classes of attack which are DoS, HTTP credential brute force and SSH brute force. The dataset was created in the emulated environment and statistics information were collected every second from the switches. The work was conducted with 9 aggregated features collected from the SDN network. This work achieved F1 score of 98%. [Ajaeiya *et al.*, (2017)]. Malicious behavior in SDN network traffic was detected utilizing machine learning classifiers in the work by Alzahrani *et al.* [Alzahrani and Alenazi, (2021)]. This work classified four types of attacks namely DDoS, probe, R2L and U2R. Decision Tree, Random Forest and XGBoost based classification was performed with only five features of NSLKDD dataset and this work accomplished a detection accuracy of 95.95%.

Ensemble classification with kNN, SVM and Extreme Learning Machine with max vote strategy was attempted by Vimala and Dhas [Vimala and Dhas, (2018)]. The dataset was created from real time traffic collected with Wireshark tool. Ensemble classification achieved better accuracy, sensitivity and specificity when compared to the individual machine learning models. Malik *et al.* attempted Cuda enabled deep learning based architecture with LSTM (Long Short Term Memory) and CNN (Convolutional Neural Networks) on CICIDS2017 dataset to classify to three attacks namely port scan, cross site scripting (XSS) and botnet attacks [Malik *et al.*, (2020)]. The work accomplished a detection accuracy of 98.6%. Even though they had attempted LSTM-DNN and LSTM–GRU models, LSTM-CNN model outperformed the rest.

Khairi *et al.* employed classification of normal flows from conflict flows by utilizing machine learning classifiers namely Decision Tree, Support Vector Machine, Extremely Fast Decision Tree (EFDT) and hybrid DT-SVM models [Khairi *et al.*, (2021)]. The work was done on Mininet emulated environment with RYU controller. EFDT and hybrid DT-SVM classifiers achieved higher classification rate of 99.49% and 99.27% respectively. Sankara Babu *et al.* have attempted Grey Wolf Optimizer and Autoencoder techniques for feature selection and dimensionality reduction respectively. The researchers have built a model using Recurrent Neural Network for medical disease prediction [Babu *et al.*, (2018)].

5. Proposed Hybrid Model

The proposed multistage NIDS model aims to detect the network anomalies using machine learning based multiclass classification. The system performs intrusion detection in an offline manner. Feature selection based on binary Grey Wolf Optimizer method was employed in the study.

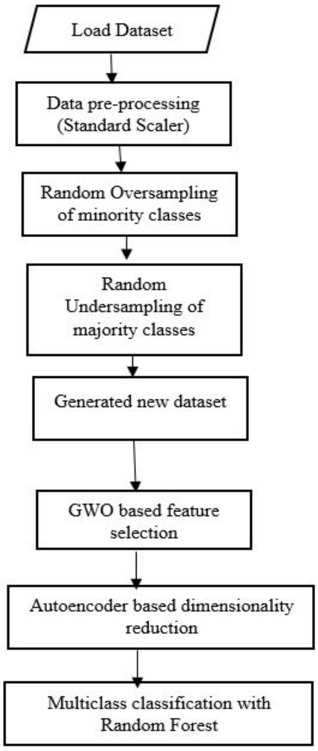

Load Dataset
Data pre-processing (Standard Scaler)
Random Oversampling of minority classes
Random Undersampling of majority classes
Generated new dataset
GWO based feature selection
Autoencoder based dimensionality reduction
Multiclass classification with Random Forest

Fig. 3. System Process Flow

The GWO selects the relevant feature subset. The feature subset produced by GWO was further processed by Autoencoder based dimensionality reduction technique. The architecture of the multistage model is depicted in Fig. 3.

5.1. *Dataset*

In this work, the recently published InSDN dataset was used for multiclass classification. The dataset was created in SDN environment and attack traffic is injected from internal and external networks. Seven different types of attacks are present in the dataset which are namely DoS, DDoS, probe, botnet, exploitation, password-guessing, and web attacks. The normal traffic was created by launching applications like HTTPS, HTTP, DNS, Email, FTP and SSH. 80 traffic features were extracted from the SDN packet capture using CICFlowmeter [Lashkari *et al.*, (2017)]. The traffic wise number of instances in the dataset is tabulated in Table 2. The dataset is an imbalanced dataset where there are four minority classes. The presence of this imbalance has to be handled while building this intrusion detection system, as machine learning classifiers tend to ignore the minority classes. At the same time, attackers can inject minority class attacks into the network which can compromise the entire system.

Class	No of instances
DDoS	121942
Probe	98129
Normal	68424
DoS	53616
BFA	1405
Web attack	192
Botnet	164
U2R	17

Table 2. Class wise Instance Count

Even though there were 80 features available in the dataset, based on the studies of Mahmoud Said ElSayed *et al.* [Elsayed *et al.*, (2020)] and Krishnan *et al.* [Krishnan *et al.*, (2019)], only 48 SDN specific features were used for the work and they are listed in Table 3.

S. No	Feature Name	S. No	Feature Name
1	Protocol	25	Fwd Header Len
2	Flow Duration	26	Bwd Header Len
3	Tot Fwd Pkts	27	Fwd Pkts/s
4	Tot Bwd Pkts	28	Bwd Pkts/s
5	TotLen Fwd Pkts	29	Pkt Len Min
6	TotLen Bwd Pkts	30	Fwd Header Len
7	Fwd Pkt Len Max	31	Pkt Len Max
8	Fwd Pkt Len Min	32	Pkt Len Mean
9	Fwd Pkt Len Mean	33	Pkt Len Std
10	Fwd Pkt Len Std	34	Pkt Len Var
11	Flow Byts/s	35	Pkt Size Avg
12	Flow Pkts/s	36	Active Mean
13	Flow IAT Mean	37	Active Std
14	Flow IAT Std	38	Active Max
15	Flow IAT Max	39	Active Min
16	Flow IAT Min	40	Pkt Len Max
17	Fwd IAT Tot	41	Pkt Len Mean
18	Fwd IAT Mean	42	Pkt Len Std
19	Fwd IAT Std	43	Pkt Len Var
20	Bwd IAT Tot	44	Pkt Size Avg
21	Bwd IAT Mean	45	Active Mean
22	Bwd IAT Std	46	Active Std
23	Bwd IAT Max	47	Active Max
24	Bwd IAT Min	48	Active Min

Table 3. Features of InSDN dataset

Further the number of features were reduced based Grey Wolf Optimization wrapper technique to derive at the minimum number of features that can classify the traffic into normal and that of malicious. Autoencoder based dimensionality reduction was attempted on the resulted subset. The visualization of the dataset with tSNE in the two dimensional space is depicted in Fig. 4.

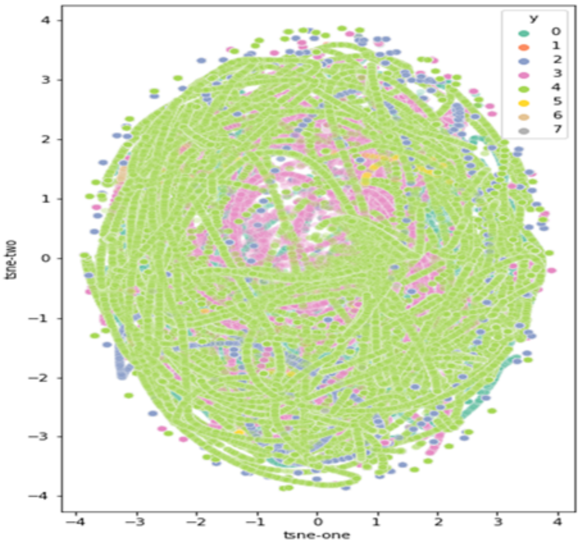


Fig. 4. Visualization of dataset with tSNE in 2D space

Visualization of the dataset indicates the nonlinear nature of the dataset. Hence tree based machine learning classifier – Random Forest classifier was employed for multiclass classification. Various other tree based classifiers like Decision Tree, Extra Tree and XGBoost classifiers were also experimented along with the proposed

GWO – AE hybrid technique. Random Forest classifier was selected due to the superior classification performance on the InSDN dataset.

5.2. *System specifications*

Hardware and software specifications used in the current study are listed in Table 4.

<table>
<tr><th colspan="3">Hardware Specification</th></tr>
<tr><td colspan="3">Intel (R) Cor (TM) i7-7500U CPU@ 2.70GHz Multicore (4 core) processor 64-bit, 12 GB RAM</td></tr>
<tr><td colspan="3"><b>Software Specification</b></td></tr>
<tr><td>Software</td><td>Name</td><td>Version</td></tr>
<tr><td>Operating System</td><td>Ubuntu</td><td>16.04.7 LTS</td></tr>
<tr><td>Machine Learning based classification</td><td>Python
scikit learn</td><td>Python 3.7.2
0.24.1</td></tr>
<tr><td>Dataset</td><td>InSDN Dataset (Elsayed <i>et al.</i>, 2020)</td><td>Available:
https://aseados.ucd.ie/datasets/SDN/</td></tr>
</table>

Table 4. Hardware and software specifications

5.3. *Data Pre-processing*

Data pre-processing is a necessary step that has to be performed before attempting machine learning based classification. The following three steps were performed as the part of data pre-processing for InSDN dataset.

(1) The dataset consisted of three different CSV files which were merged to create a single CSV file. 48 SDN specific features were selected for performing the classification. Label encoding technique was attempted to convert the string labels into numerical representation of classes. One hot encoding was performed for mapping the categorical feature ‘protocol’ into numerical values.

(2) Further standard scaling (z-score normalization) technique was attempted as various traffic features were of different range. For converting the range of values between 0 and 1, z-score normalization technique as given in “Eq. (15)” was performed.

$$z = \frac{x_{ij} - \mu}{\sigma} \qquad (15)$$

where x_{ij} represents the value for the i^{th} observation for j^{th} column, μ represents the average of the values in the column, σ represents the standard deviation of the observations.

(3) The dataset was divided into training set and test set with 70:30 ratio. In order to handle the class imbalance problem, data sampling techniques were attempted on the training set. The test set was excluded from data sampling in order to avoid the overfitting problem. Random oversampling technique was used for increasing the number of instances of the minority classes namely Web attack, U2R, botnet and password-guessing. This was followed by random under sampling technique which were applied to majority classes normal, DDoS, DoS and probe classes. In our study, the combination of random under sampling and random oversampling techniques were giving better results than Synthetic Minority Oversampling (SMOTE) technique and combination of SMOTE and Edited Nearest Neighbours under sampling (ENN) technique - SMOTE-ENN.

6. Results and Discussion

6.1. *Feature Selection & Dimensionality Reduction*

Feature selection with metaheuristic Grey Wolf Optimization wrapper technique was performed to select the important features there by reducing the feature space which helped in producing a less complex model. With autoencoder, compressed representation of the feature space was further achieved. The selected features with both techniques are tabulated in Table 5.

Feature selection	Number of features selected
GWO Feature Selection	35
Autoencoder Dimensionality Reduction	9 vector elements

Table 5. Summary of features selected

From the obtained feature importance, it is noticed that attribute ‘protocol’ was found to be an important feature due to its high discriminating power. Also, attacker tends to launch attack with minimum number of packets, in order to save the resources at their end. The legitimate flows contain at least 5 packets, and any flow which contains less than 5 packets are considered as abnormal [Gkountis, *et al.*, (2017)] [Peng *et al.*, (2003)]. Hence packet length and packet size are important features while detecting intrusions. As attackers usually spoof their IP address, there is high chance of asymmetry of flows in forward and backward direction, hence total number of packets in forward direction and the same in backward direction are key features as they help in detecting orphan flows which are created during an attack.

6.2. *Multiclass classification performance*

In this work, Random Forest machine learning classifier was used for evaluating multiclass classification performance. The evaluation of the multiclass classification was done with six different performance metrics which are precision, recall and F1-score, Mathews Correlation Coefficient (MCC), ROC curves and PR Curves. For evaluating the performance of multiclass classification, macro, micro and weighted metrics were considered. Macro metrics weights majority and minority classes equally by computing the metrics for each class independently and then taking the average. Micro metrics computation is performed by aggregating the contributions of all classes equally to compute the average metrics. The better performance of the majority classes influences the micro metrics and in the case of multiclass classification problem with imbalanced data, the results get skewed to the majority classes [Bagui and Li, (2021)]. The weighted metrics considers the size of the classes into account. The weighted metrics will be high, in the cases where the majority classes are well classified but at the same time, minority classes are not classified correctly. In such cases, weighted metrics fails to reflect the bad performance of minority classes.

The assessment metrics are defined as given below.
a. Precision is given by $\frac{TP}{TP+FP}$, which represents the positive cases of attack which are detected from the total predicted positive cases.

b. Recall refers to the total positive cases of attack that are detected correctly and is given by $\frac{TP}{TP+FN}$, where TP refers to True Positive values, FP refers to False Positive values, FN refers to False Negative Values retrieved from confusion matrix.

c. F1 score combines both precision and recall is the harmonic mean between precision and recall. F1 score is given by $\frac{2*Recall\ \times Precision}{Recall+Precision}$

d. Mathews Correlation Coefficient (MCC) – MCC is the performance metrics used especially in case of imbalanced classes. It is more useful than F-score, as proportion of the classes in the confusion matrix is considered in the calculation [Davide and Giuseppe, (2020)]. It is expressed as in “Eq. (16)” and “Eq. (17)” [Li *et al.*, (2012)].

$$\mathrm{MCC} = \frac{\mathrm{TP_i} \times \mathrm{TN_i} - \mathrm{FP_i} \times \mathrm{FN_i}}{\sqrt{(\mathrm{TP_i}+\mathrm{FN_i})(\mathrm{TP_i}+\mathrm{FP_i})(\mathrm{TN_i}+\mathrm{FP_i})(\mathrm{TN_i}+\mathrm{FN_i})}} \qquad (16)$$

$$\mathrm{MCC_{avg}} = \frac{\sum \mathrm{MCC_i}}{8} \qquad (17)$$

e. Receiver Operator Characteristics (ROC) curves provide the summary of efficiency of classifiers on a range of TPRs (True Positive Rate) and FPRs (False Positive Rate) and is the illustrative method of measuring classifier accuracy. TPR is also called as sensitivity or recall. FPR is given by $\frac{FP}{FP+TN}$. ROC curve determines the percentage of observations which are properly classified for a certain FPR by evaluating the trained models at different error values. Area under ROC curve (AUC) metric is a commonly used metric for evaluating the classifiers facing class imbalance problem [Zavrak and Iskefiyeli, (2020)]. It is used as a measure to evaluate the classifier performance, and considers the two-dimensional area under ROC curve. AUC is impassive by prior probabilities and chosen threshold, it provides a single number to compare classifiers. AUC values always range between 0 and 1.

The formula to find AUROC [Othman *et al.*, (2018)] is given in "Eq. (18)".

$$\mathrm{AUROC} = \int_0^1 (\frac{\mathrm{TP}}{\mathrm{P}})\, \mathrm{d}\, (\frac{\mathrm{FP}}{\mathrm{N}}) \qquad (18)$$

f) Precision Recall Curve (PR Curve) explains the trade-off between precision and recall at various threshold by plotting precision against recall at various thresholds. PR curve depicts how the performance of classifier changes with respect to different thresholds. The optimal threshold generally appears at the top right portion of the curve [Blevins *et al.*, (2021)]. Area Under Precision Recall curve (AUPR) is the two-dimensional area under PR curve and provides a single number to evaluate the classifiers. The higher AUPR indicates better classification performance. The AUPR is given as in "Eq. (19)" [Othman *et al.*, (2018)].

$$\mathrm{AUPR} = \int_0^1 (\frac{\mathrm{TP}}{\mathrm{TP+FP}})\, \mathrm{d}\, (\frac{\mathrm{TP}}{\mathrm{P}}) \qquad (19)$$

After necessary pre-processing, GWO selected features were submitted to autoencoder resulting in nine vector elements. The nine vectors were fed to the Random Forest classifier. The performance evaluation of the classifier was done with five fold cross validation. The test set results obtained are tabulated in Table 6. The micro, macro and weighted metrics show that nine vector based GWO-AE-RF hybrid model obtained competitive results.

Accuracy	**Micro Averaged Precision**	**Macro Averaged Precision**	**Weighted Average Precision**	**Micro Averaged Recall**	**Macro Averaged Recall**	**Weighted Average Recall**	**Micro Averaged F1 score**	**Macro Averaged F1 score**	**Weighted Average F1 score**	**MCC**
98.9%	98.95	93.2	98.95	98.95	87.54	98.95	98.95	89.73	98.95	98.56

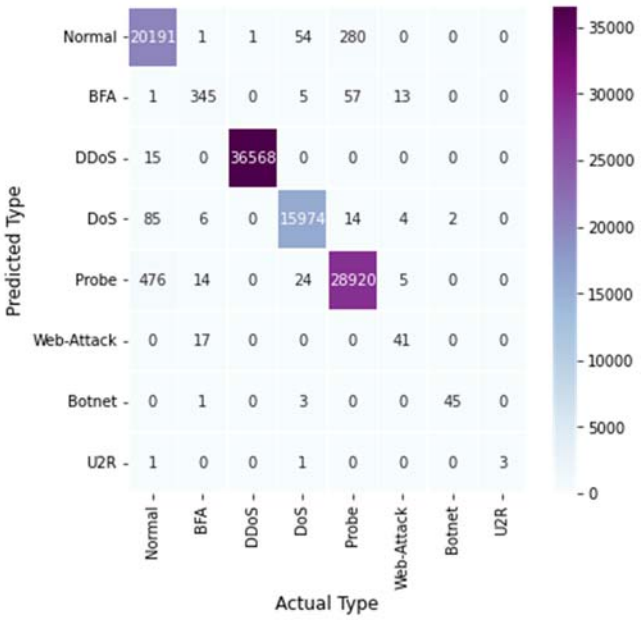
Table 6. Performance of GWO – AE – RF model

The class wise performance of the model was also evaluated which tabulated in Table 7. The model performed very well with the reduced dimensionality space.

GWO –AE - RF model	Normal	Brute Force	DDoS	DoS	Probe	Web Attack	Botnet	u2R
Precision	97	90	100	99	99	65	96	100
Recall	98	82	100	99	98	71	92	60
F1 score	98	86	100	99	99	68	94	75

Table 7. Class wise performance of GWO – AE – RF model

The confusion matrix, ROC curves and precision curves of the hybrid GWO – AE – RF model depicted in Fig. 5 Fig. 6. and Fig. 7. respectively demonstrate the high performance obtained by the model.



	Normal	BFA	DDoS	DoS	Probe	Web-Attack	Botnet	U2R
Normal	20191	1	1	54	280	0	0	0
BFA	1	345	0	5	57	13	0	0
DDoS	15	0	36568	0	0	0	0	0
DoS	85	6	0	15974	14	4	2	0
Probe	476	14	0	24	28920	5	0	0
Web-Attack	0	17	0	0	0	41	0	0
Botnet	0	1	0	3	0	0	45	0
U2R	1	0	0	1	0	0	0	3

Fig.5. Confusion Matrix of GWO-AE-RF model

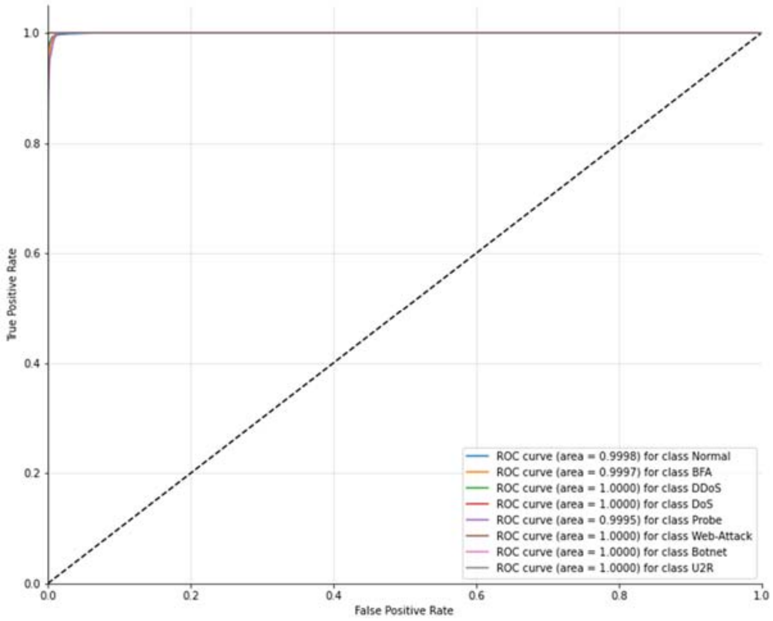


Fig.6. ROC curve for GWO – AE – RF model

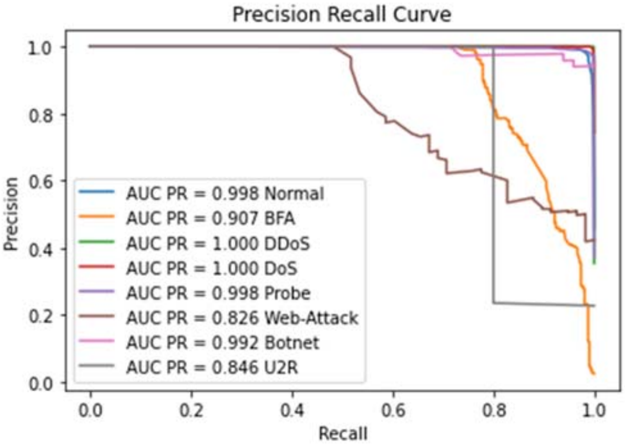


Fig.7. Precision Curve for GWO – AE – RF Model

6.3. *Comparison of results with other feature selection techniques*

The results of the hybrid GWO – AE – RF model was compared with the other feature selection techniques Correlation based feature selection (CBFS) and Tree based feature selection (TBFS). Also, the dimensionality reduction with Autoencoder with the complete features were also analyzed. Area under precision recall curve was used for comparison as the metrics is found effective for evaluating imbalanced classes. AU-PR curve of the RF classifier with the four techniques have been tabulated in Table 8.

Technique	Normal	Brute Force	DDoS	DoS	Probe	Web Attack	Botnet	u2R
CBFS – RF AUPR	0.999	0.871	1.00	1.00	0.998	0.81	0.991	0.805
TBFS – RF AUPR	0.998	0.889	1.00	1.00	0.998	0.801	0.987	1.00
AE – RF AUPR	0.995	0.737	1.00	0.997	0.998	0.702	0.992	0.401
GWO – AE – RF AUPR	0.998	0.907	1.00	1.00	0.998	0.826	0.992	0.846

Table 8. Comparison of AU-PR curves for various feature selection techniques

Results shows the high performance of the hybrid GWO – AE – RF model in detecting the minority classes.

7. Conclusion

In this paper, various feature selection and feature extraction techniques were studied and experimented with the latest InSDN dataset. The dataset was an imbalanced dataset with four majority classes and four minority classes. In order to obtain a model that works with reduced feature space, a hybrid GWO selected Autoencoder reduced Random Forest classifier based hybrid model was developed. The GWO offered feature subset was reduced to 9 vector elements by Autoencoder. The hybrid GWO – AE – RF model achieved weighted F1 score of 98.95%. The model was found superior than the reduced feature subsets obtained from Correlation based feature selection, Tree based feature importance and Autoencoder based dimensionality reduction. Hyperparameter optimization is an effective way to increase the classifier performance. Metaheuristic hyperparameter optimization techniques are found to be effective in improving the multiclass classification will be the focus of our future works. The latest research works also prove that deep learning models achieve better performance than the machine learning counter parts. Metaheuristic optimization of deep learning model hyperparameters for improved multiclass classification in attack detection will be the focus of our future works.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Abdallah, M., An Le Khac, N., Jahromi, H., & Delia Jurcut, A. (2021, August). A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs. In The 16th International Conference on Availability, Reliability and Security (pp. 1-7).

[2] Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). IEEE Access, 9, 26766-26791.

[3] Agborubere, B., & Sanchez-Velazquez, E. (2017, June). Openflow communications and tls security in software-defined networks. In 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 560-566). IEEE.

[4] Ahuja, N., Singal, G., & Mukhopadhyay, D. (2021, January). DLSDN: Deep learning for DDOS attack detection in software defined networking. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 683-688). IEEE.

[5] Ajaeiya, G. A., Adalian, N., Elhajj, I. H., Kayssi, A., & Chehab, A. (2017, July). Flow-based intrusion detection system for SDN. In 2017 IEEE Symposium on Computers and Communications (ISCC) (pp. 787-793). IEEE.

[6] Alamri, H. A., & Thayananthan, V. (2020). Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks. IEEE Access, 8, 194269-194288.

[7] Al-Rubaye, S., Kadhum, E., Ni, Q., & Anpalagan, A. (2017). Industrial internet of things driven by SDN platform for smart grid resiliency. IEEE Internet of Things Journal, 6(1), 267-277.

[8] Al-Qatf, M., Lasheng, Y., Al-Habib, M., & Al-Sabahi, K. (2018). Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. Ieee Access, 6, 52843-52856.

[9] Alzahrani, A. O., & Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. Future Internet, 13(5), 111.

[10] Babu, S. B., Suneetha, A., Babu, G. C., Kumar, Y. J. N., & Karuna, G. (2018). Medical disease prediction using grey wolf optimization and auto encoder based recurrent neural network. Periodicals of Engineering and Natural Sciences (PEN), 6(1), 229-240.

[11] Bagui, S., & Li, K. (2021). Resampling imbalanced data for network intrusion detection datasets. Journal of Big Data, 8(1), 1-41.

[12] Band, S. S., Janizadeh, S., Chandra Pal, S., Saha, A., Chakrabortty, R., Melesse, A. M., & Mosavi, A. (2020). Flash flood susceptibility modeling using new approaches of hybrid and ensemble tree-based machine learning algorithms. Remote Sensing, 12(21), 3568.

[13] Bao, T. H. Q., Le, L. T., Thinh, T. N., & Pham, C. K. (2022). A High-Performance FPGA-Based Feature Engineering Architecture for Intrusion Detection System in SDN Networks. In International Conference on Intelligence of Things (pp. 259-268). Springer, Cham.

[14] Bawany, N. Z., & Shamsi, J. A. (2016, May). Application layer DDoS attack defense framework for smart city using SDN. In The Third International Conference on Computer Science, Computer Engineering, and Social Media (CSCESM2016) (p. 1).

[15] Blevins, D. H., Moriano, P., Bridges, R. A., Verma, M. E., Iannacone, M. D., & Hollifield, S. C. (2021). Time-based can intrusion detection benchmark. arXiv preprint arXiv:2101.05781.

[16] Braga, R., Mota, E., & Passito, A. (2010, October). Lightweight DDoS flooding attack detection using NOX/OpenFlow. In IEEE Local Computer Network Conference (pp. 408-415). IEEE.

[17] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). Classification and regression trees. Routledge.

[18] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16-28.

[19] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics, 21(1), 1-13.

[20] Elsayed, M. S., Le-Khac, N. A., & Jurcut, A. D. (2020). InSDN: A novel SDN intrusion dataset. IEEE Access, 8, 165263-165284.

[21] Farahnakian, F., & Heikkonen, J. (2018, February). A deep auto-encoder based approach for intrusion detection system. In 2018 20th International Conference on Advanced Communication Technology (ICACT) (pp. 178-183). IEEE.

[22] "Flowtbag (2015) Program to Calculate Flow Statistics from a Given Capture File [online]
https://github.com/danielarndt/flowtbag (Accessed 1 May 2022). "

[23] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine learning, 63(1), 3-42.

[24] Gkountis, C., Taha, M., Lloret, J., & Kambourakis, G. (2017, September). Lightweight algorithm for protecting SDN controller against DDoS attacks. In 2017 10th IFIP Wireless and Mobile Networking Conference (WMNC) (pp. 1-6). IEEE.

[25] Goetz, M., Weber, C., Bloecher, J., Stieltjes, B., Meinzer, H. P., & Maier-Hein, K. (2014). Extremely randomized trees based brain tumor segmentation. Proceeding of BRATS challenge-MICCAI, 006-011.

[26] Gouveia, A., & Correia, M. (2020). Network intrusion detection with XGBoost. In Recent Advances in Security, Privacy, and Trust for Internet of Things (IoT) and Cyber-Physical Systems (CPS) (pp. 137-166). Chapman and Hall/CRC.

[27] Hall, M. A. (1999). Correlation-based feature selection for machine learning (Doctoral dissertation, The University of Waikato).

[28] Harris, A., Mintaria, A. E., Stiawan, D., bin Idris, M. Y., & Budiarto, R. (2020, October). Improving the anomaly detection by combining pso search methods and j48 algorithm. In 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI) (pp. 119-126). IEEE.

[29] Harris, A., Mintaria, A. E., Stiawan, D., bin Idris, M. Y., & Budiarto, R. (2020, October). Improving the anomaly detection by combining pso search methods and j48 algorithm. In 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI) (pp. 119-126). IEEE.

[30] Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature selection for intrusion detection using random forest. Journal of information security, 7(3), 129-140.

[31] Hastie, T., Tibshirani, R., & Friedman, J. (2001). Prototype methods and nearest-neighbors. In The elements of statistical learning (pp. 459-483). Springer, New York, NY.

[32] Henckaerts, R., Côté, M. P., Antonio, K., & Verbelen, R. (2021). Boosting insights in insurance tariff plans with tree-based machine learning methods. North American Actuarial Journal, 25(2), 255-285.

[33] Henriques, J., Caldeira, F., Cruz, T., & Simões, P. (2020). Combining k-means and xgboost models for anomaly detection using log datasets. Electronics, 9(7), 1164.

[34] Hu, L., Liu, B., Ji, J., & Li, Y. (2020). Tree-Based Machine Learning to Identify and Understand Major Determinants for Stroke at the Neighborhood Level. Journal of the American Heart Association, 9(22), e016745.

[35] Hu, F., Hao, Q., & Bao, K. (2014). A survey on software-defined network and openflow: From concept to implementation. IEEE Communications Surveys & Tutorials, 16(4), 2181-2206.

[36] Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016, May). A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS) (pp. 21-26).

[37] Javeed, D., Gao, T., & Khan, M. T. (2021). SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT. Electronics, 10(8), 918.

[38] Johnson, R., & Zhang, T. (2013). Learning nonlinear functions using regularized greedy forest. IEEE transactions on pattern analysis and machine intelligence, 36(5), 942-954.

[39] Jose, A. S., Nair, L. R., & Paul, V. (2021). Towards Detecting Flooding DDOS Attacks Over Software Defined Networks Using Machine Learning Techniques. REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS, 11(4), 3837-3865.

[40] Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 1200-1205). Ieee.

[41] Kazemitabar, J., Amini, A., Bloniarz, A., & Talwalkar, A. S. (2017). Variable importance using decision trees. Advances in neural information processing systems, 30.

[42] SYED ARIFFIN, S. H., Latiff, A., Muazzah, N., Skudai, M. C., & Naji, F. (2021). Detection and Classification of Conflict Flows in SDN Using Machine Learning Algorithms.

[43] Kim, H., & Feamster, N. (2013). Improving network management with software defined networking. IEEE Communications Magazine, 51(2), 114-119.

[44] Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2014). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14-76.

[45] Krishnan, P., Duttagupta, S., & Achuthan, K. (2019). VARMAN: Multi-plane security framework for software defined networks. Computer Communications, 148, 215-239.

[46] Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., & Ghorbani, A. A. (2017, February). Characterization of tor traffic using time based features. In ICISSp (pp. 253-262).

[47] Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., & Dai, K. (2012). An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert systems with applications, 39(1), 424-430.

[48] Malik, J., Akhunzada, A., Bibi, I., Imran, M., Musaddiq, A., & Kim, S. W. (2020). Hybrid deep learning: An efficient reconnaissance and surveillance detection mechanism in SDN. IEEE Access, 8, 134695-134706.

[49] Min, B., Yoo, J., Kim, S., Shin, D., & Shin, D. (2021). Network anomaly detection using memory-augmented deep autoencoder. IEEE Access, 9, 104695-104706.

[50] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.

[51] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089.

[52] 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.

[53] Narayanadoss, A. R., Truong-Huu, T., Mohan, P. M., & Gurusamy, M. (2019, April). Crossfire attack detection using deep learning in software defined ITS networks. In 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring) (pp. 1-6). IEEE.

[54] Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications surveys & tutorials, 16(3), 1617-1634.

[55] "ONF Software-defined Networking: The New Norm for Networks (2012) [online] http://opennetworking.wpengine.com/wp-content/uploads/2011/09/wp-sdn-new/norm.pdf [Accessed 24 May 2022]."

[56] Othman, S. M., Ba-Alwi, F. M., Alsohybe, N. T., & Al-Hashida, A. Y. (2018). Intrusion detection model using machine learning algorithm on Big Data environment. Journal of big data, 5(1), 1-12.

[57] Peng, T., Leckie, C., & Ramamohanarao, K. (2003, May). Protection from distributed denial of service attacks using history-based IP filtering. In IEEE International Conference on Communications, 2003. ICC'03. (Vol. 1, pp. 482-486). IEEE.

[58] Pham, B. T., Khosravi, K., & Prakash, I. (2017). Application and comparison of decision tree-based machine learning methods in landside susceptibility assessment at Pauri Garhwal Area, Uttarakhand, India. Environmental Processes, 4(3), 711-730.

[59] Schreyer, M., Sattarov, T., Borth, D., Dengel, A., & Reimer, B. (2017). Detection of anomalies in large scale accounting data using deep autoencoder networks. arXiv preprint arXiv:1709.05254.

[60] Siddiqi, M. A., & Pak, W. (2020). Optimizing filter-based feature selection method flow for intrusion detection system. Electronics, 9(12), 2114.

[61] Vimala, S., & Dhas, J. (2018). SDN based DDoS attack detection system by exploiting ensemble classification for cloud computing. International Journal of Intelligent Engineering and Systems, 11(6), 282-291.

[62] Aouedi, O., Piamrat, K., & Parrein, B. (2022). Intelligent Traffic Management in Next-Generation Networks. Future internet, 14(2), 44.

[63] Xu, Z., King, I., Lyu, M. R. T., & Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. IEEE Transactions on Neural networks, 21(7), 1033-1047.

[64] Yang, L., Moubayed, A., & Shami, A. (2021). MTH-IDS: a multitiered hybrid intrusion detection system for Internet of vehicles. IEEE Internet of Things Journal, 9(1), 616-632.

[65] Yarveicy, H., & Ghiasi, M. M. (2017). Modeling of gas hydrate phase equilibria: Extremely randomized trees and LSSVM approaches. Journal of Molecular Liquids, 243, 533-541.

[66] Zavrak, S., & İskefiyeli, M. (2020). Anomaly-based intrusion detection from network flow features using variational autoencoder. IEEE Access, 8, 108346-108358.

[67] Zheng, H., Yuan, J., & Chen, L. (2017). Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. Energies, 10(8), 1168.

[68] Zhou, C., & Paffenroth, R. C. (2017, August). Anomaly detection with robust deep autoencoders. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 665-674).

Authors Profile

Ancy Sherin Jose, is pursuing research under division of Computer Science Engineering, Cochin university of Science and Technology. She is a B. Tech, M. Tech holder in Computer Science. Her research areas are SDN, Network Security, Machine Learning, Deep Learning and Big Data Analytics. She has published papers in the network security domain.

Dr. Latha R. Nair, is working as Associate Professor in the division of Computer Engineering, Cochin University of Science and Technology. She is a B. Tech, M. Tech and Ph.D. holder in Computer Science. She has published a number of papers in the areas of machine intelligence and natural language processing. She has done extensive research in Malayalam language computing. Her areas of interest are machine intelligence, natural language processing and image processing.

Dr. Varghese Paul, is working as Post Graduate Professor in Computer Science and Engineering Department, in Rajagiri School of Engineering and Technology. He is a BSc. MTech, Ph.D. holder in Computer Science. His research areas are Data security using Cryptography, Data Compression, Data Mining, Image Processing and E-Governance. He is the developer of TDMRC Coding System for character representation and encryption system using this coding system. He has got many research publications in international as well as national journals. He is a certified Software Test Manager, Ministry of Information Technology, Government of India. Also, member of Information System Audit and Control Association USA and Indian Society for Technical Education, India.