Article, Review

Optimization of Intrusion Detection Systems Determined by Ameliorated HNADAM-SGD Algorithm

Shyla Shyla[1,*] , Vishal Bhatnagar[2], Vikram Bali[3] and Shivani Bali[4]

[1] Research Scholar, NSUT East Campus Formerly Ambedkar Institute of Advanced Communication and Technologies, GGSIPU, New Delhi, India, *Correspondence; shylasinghit@gmail.com; ORCID; 0000-0002-8279-4195,

[2] Professor, NSUT East Campus Formerly Ambedkar Institute of Advanced Communication Technologies and Research, New Delhi, India;vishallbhatnagar@yahoo.com ORCID; 0000-0002-2681-0103

[3] Professor and Head-CSE, Department of Computer Science and Engineering, JSS Academy of Technical Education, Noida, Uttar Pradesh, India; vikramgcet@gmail.com; ORCID;0000-0002-2809-8455

[4] Professor and Area Chair-Business Analytics, Jaipuria Institute for Management, Noida, Uttar Pradesh, India; lbsshivani@gmail.com; ORCID; 0000-0001-5618-1857

Abstract: A single Information security is of pivotal concern for consistently streaming information over the widespread internetwork. The bottleneck flow of incoming and outgoing data traffic introduces the issue of malicious activities taken place by intruders, hackers and attackers in the form of authenticity desecration, gridlocking data traffic, vandalizing data and crashing the established network. The issue of emerging suspicious activities is managed by the domain of Intrusion Detection Systems (IDS). The IDS consistently monitors the network for identification of suspicious activities and generates alarm and indication in presence of malicious threats and worms. The performance of IDS is improved by using different signature based machine learning algorithms. In this paper, the performance of IDS model is determined using hybridization of nestrov-accelerated adaptive moment estimation –stochastic gradient descent (HNADAM-SDG) algorithm. The performance of the algorithm is compared with other classification algorithms as logistic regression, ridge classifier and ensemble algorithm by adapting feature selection and optimization techniques.

Keywords: Intrusion Detection System (IDS); HNADAM-SDG(Hybrid Nestrov-Accelerated Adaptive Moment Estimation –Stochastic Gradient Descent); Network-based Intrusion Detection System (NIDS); Host-based Intrusion Detection System (HIDS); Signature-based Intrusion Detection System (SIDS); Anomaly-based Intrusion Detection System (AIDS); Algorithms; Machine Learning.

1.Introduction

The extensive practice of technology over widespread network introduces the importance of information security. The illegitimate activities of intruders is kept under consistent monitoring provided by different security e-equipment as firewall, anti-malware system, anomaly detection systems, endpoint systems and intrusion detection systems (IDS). The significant amount of businesses, organizations and different sectors gained access to technical, wireless and cloud resources. The availability of on demand resources allows users to store the voluminous amount of data and information over the service providers platform where the platforms are open sourced software, open clouds and integrated development environments. The extent of exposure to virtual resources give arise to evolving problem of virtual security threats. (Mohd, Singh and Bhadauria., 2021) found that the security threats can arise within the network or outside the network by multiple intruders in the form of cyber-attacks, malware and worms. IDS are intrusion detection systems that track malicious activities by monitoring incoming and outgoing network traffic, the alarm or alerts are generated on the basis of identification of unusual activities within the network. The alarm is generated if the intruder attempts to hamper the virtual security by gaining unauthorized access to private network, wide area network, personal network, personal computer and large scale computer hubs by passing information to administration of security breach. The IDS system aims to catch the suspicious activity of intruders or hackers before they forges the information by crashing the network system. The IDS are defined as network based intrusion detection system (NIDS), host based intrusion detection system (HIDS), Signature based intrusion detection system (SIDS) and anomaly based intrusion detection system (AIDS). The one of the most popularly used IDS is SNORT which usually works on Unix or Linux based operating system with lightweight NIDS (Pokuri., 2021). The IDS system installed over any network is used for detection of malicious activities and to keep the track of different types of cyber-attacks imposed over the network. The recorded attack patterns helps to identify future attack occurrences in any organization to change and adopt the enhanced security systems. It is able to track bugs or network configurations. The IDS sensory devices is the other methodology to make system more secure and sustainable by using alarm filtering technique to differentiate between malicious and normal activity patterns.

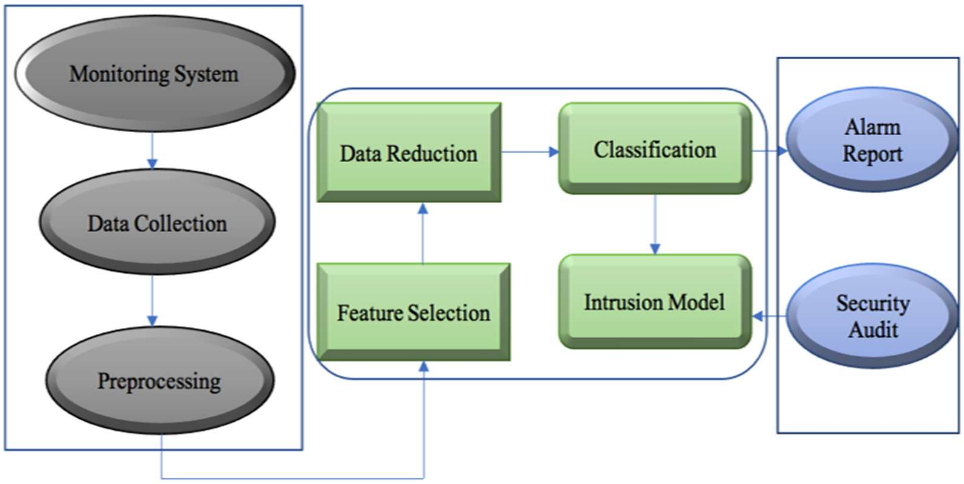


Figure 1: Schematic Representation of Intrusion Detection system

The figure 1 shows the schematic representation of intrusion detection system. This is the illustration for designing IDS model where hypertuning and optimization is the technique. The methodology to design a machine learning algorithm with good classification and prediction result involves data collection, data preprocessing, feature selection, data reduction and classification. The IDS works as a monitoring system that constantly tracks incoming and outgoing network traffic, the network attack and malicious activity is captured in the form of data which is then used for understanding attack patterns and for changing the security model on the basis of updated attack pattern.

The cogency of paper is defined as, the paper is focused on finding the hyperparameter optimization methodology for UNSW-NB15 dataset to gain the higher accuracy by minimizing the generalization error. The proposed HNADAM-SDG algorithm trains the network with only a subset of hyperparameters configuration where the subset of configuration is randomly opted. The hyperparameters are defined before the model training to provide the flexibility to fit the model on the basis of dataset. The compulsions of gradient descent algorithm is accelerated by optimization of hyperparameters. The performance of the algorithm is compared with other classification algorithms as logistic regression, ridge classifier and ensemble learning algorithm by adapting feature selection and optimization techniques. The methodology is used to achieve the following objectives.

- To find the combination of hyperparameters to maximize the performance of model by diminishing generalization error and computational cost.

- To find a hyperparameter response space which depends on methodology, hyperparameters, dataset and metrics.

- To deploy a methodology by sampling of candidate parameters using cross-validation scheme.

The paper is organized as follows section 2 represents related work. The detail description of dataset has been explained in section 3 and then the detailed explanation of the methodology used in the proposed algorithm has been explained in section 4. In section 5 the proposed algorithm has been explained in detail, in section 6 the results observed from the proposed algorithm and the discussion of the result has been done and in section 7 the conclusion of the proposed work has been explained.

2. Related Work

The In the recent research trends it is been observed that the method of feature selection is modified to enhance the classification performance of the designed model. It has been found by (Iman and Ahmad, 2020) that based on selected features and data reduction the classification performance of the model enhances. The authors performed test on three different set of features to validate its hypothesis of enhancing performance of model on the basis of different features and performance of classification model becomes faster when size of data is small. The three feature section methods used are IG + Correlation, Multimodal fusion, GA + LR. The accuracy and precision is computed in two different scenario and performance measured without data reduction, It is observed that in scenario 2 the performance of three feature set is increased by 0.02% and in scenario 1 the performance is increased by 0.03%.

These authors (Drewek-Ossowicka, Pietrołaj and Rumiński., 2021) made a survey in the area of usage of neural network for IDS. The survey includes the study of different types of dataset that are extensively used for designing IDS as KDD Cup 1999, NSL-KDD, UNSW-NB-15 and Kyoto2006+. The neural network technology is often combined with hybrid models for better performance. The different datasets studied shows some drawbacks of being redundant and older.
The challenges faced by IDS is during occurrences of newly generated cyber-attacks. The existing cyber-attacks are easily detected by model on the basis of attack pattern.

A hybrid approach has been introduced by (Pokuri., 2021) for analysing IDS using Naïve Bayes and improved BAT algorithm by analysing selected features. The author validates the hypothesis that the feature section methodology improves the performance of the anomaly detection model. In this the features are ranked on the basis of their weight values using IG algorithm, the features marked with same ranks or falls in the category of same weights are grouped together to form sets which is then applied using BA algorithm.

The processing of subsets results in feature optimization, the optimized features are applied over random forest algorithm in the form of different sized feature sets as 15, 20, 35 and found that random forest, J48 results into better classification performance.

IDS for advanced metering infrastructure (AMI) in grid systems has been done by (Yao et al., 2021) as they are the target due to its two way communication ability over internetwork. To overcome the problem of considering global and temporal characteristics of malicious information the long short-term memory (LSTM) networks based convolution neural network (CNN) algorithm is used which is fused with cross layer features over AMI IDS. The authors proposed an LSTM-CNN feature-fusion based cross layer IDS to track the normal and suspicious behaviour of data. The KDD Cup 99 and NSL-KDD dataset is used with proposed algorithm to test and train the model. The proposed model shows overall accuracy of 99.79%.

An IDS model based on data integrity attacks as DI-EIDS to achieve low false alarm rate and high sensing rate has been proposed by (Benisha & Ratna., 2020). The proposed methodology is classified as sampling and feature selection based on attack patterns tracked by IDS. The black forest classifier (BFC) is used for training the model, grey wolf optimization (GWO) and deviation forest (d-forest) is used for optimization of ratio and barriers removal for sampling selection. The research gap is that the training procedure takes longer time than usual and it is used only for the detection of data integrity based attacks. The overall accuracy for DI-EIDS algorithm is 0.947 with overall precision, recall and F-measure as 0.627, 0.671 and 0.625. The prediction measures in comparison with NB, ELM, SVM algorithm the proposed algorithm achieves high detection rate with low false alarm rate.

A new hybridization techniques is proposed by authors (Samriya and Kumar., 2020) to provide security to cloud computing environment against, phishing, fake identity and data absconding detection. The proposed algorithm upgrades fitness value automatically by modifying clusters using fuzzy based ANN. The spider monkey optimization method is used for dataset and dimensionality reduction. The NSL-KDD dataset is reduced and optimized by FCM-SMO cluster classifier. It is found that the performance of proposed algorithm outstands existing algorithms as ANN, FCM+SVM, ANN+FCM. The overall accuracy of hybridization algorithm is ranging between 0.80 – 0.85 percent with precision, sensitivity, F-measure and specificity as 0.85-1, 0.85-0,95, 0.67-0.75 and 0.85-0.88 percent.

2.1. *Research Gap*

Performance of the machine learning algorithm is dependent on the selection of hyperparameters.

Hyperparameters optimization algorithm performance depends on various factors as, number of hidden layers, number of units per-layer, activation function, dropout amount, regulizer learning rate and weight decay.

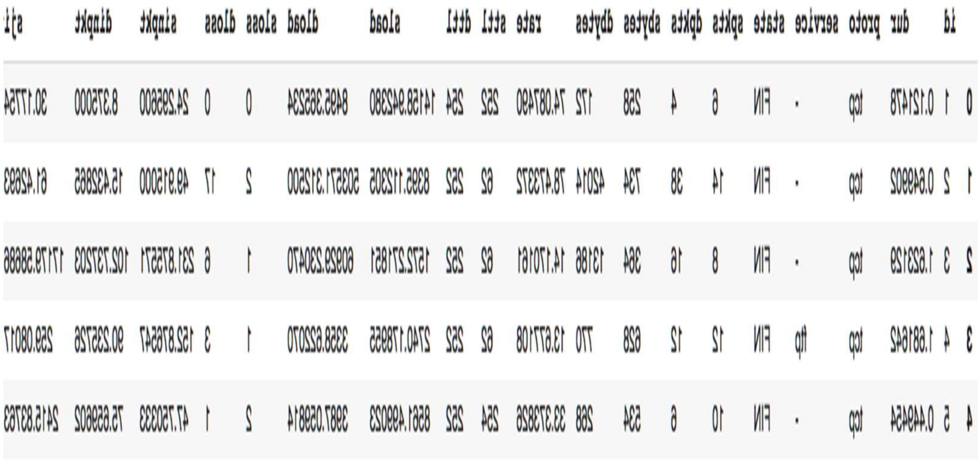
The non-optimal setting of hyperparameters will drastically affect the algorithm performance varies from extremely low learning rate to very large learning rate.

The hyper-tuning approach varies depending on type of dataset, nature of dataset and its size as there is no well-defined formula to find hyperparameters.

3.Dataset Description

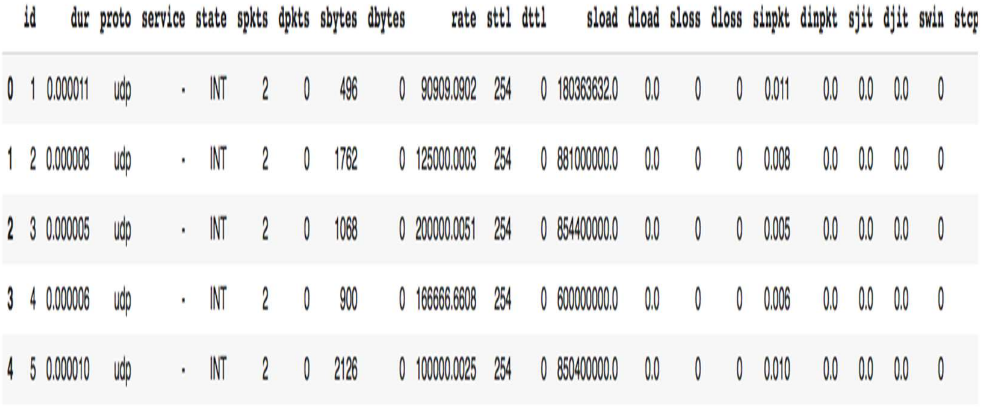
The UNSW-NB15 dataset was created by PerfectStorm tool in the Australian Centre for Cyber Security Cyber Range Lab for tracking of synthetic and normal contemporary attack patterns. In this 100 Gb of Pcap Raw traffic files is used by Tcpdump tool. The dataset is enclosing the attack patterns for nine type of attacks as Backdoors, Fuzzers, Analytical, DoS, Exploits, generic, reconnaissance, worms, shellcode, phishing. The HANADAM-SDG algorithm is used to analyze and train 49 features developed with class labels in the dataset.

The total of two million and 540,044 records are used which is segregated in four CSV extended files as, UNSW-NB15$_1$.csv, UNSW-NB15$_2$.csv, UNSW-NB$_3$.csv and UNSW-NB$_{4.csv}$. The list of event file and truth table file is named as UNSW-NB15Ground_Truth.csv and UNSW-NB15List_Event.csv.



	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	sinpkt	dinpkt	sjit
0	1	0.121478	tcp	-	FIN	6	4	258	172	74.087490	252	254	14158.942380	8495.365234	0	0	24.295600	8.375000	30.17754
1	2	0.649902	tcp	-	FIN	14	38	734	42014	78.473372	62	252	8395.112305	503571.312500	2	17	49.915000	15.432865	61.42693
2	3	1.623129	tcp	-	FIN	8	16	364	13186	14.170161	62	252	1572.271851	60929.230470	1	6	231.875571	102.737203	17179.58686
3	4	1.681642	tcp	ftp	FIN	12	12	628	770	13.677108	62	252	2740.178955	3358.622070	1	3	152.876547	90.235726	259.08017
4	5	0.449454	tcp	-	FIN	10	6	534	268	33.373826	254	252	8561.499023	3987.059814	2	1	47.750333	75.659602	2415.83763

Figure 2: Test Data



	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	sinpkt	dinpkt	sjit	djit	swin	stcp
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	254	0	180363632.0	0.0	0	0	0.011	0.0	0.0	0.0	0	
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	254	0	881000000.0	0.0	0	0	0.008	0.0	0.0	0.0	0	
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	254	0	854400000.0	0.0	0	0	0.005	0.0	0.0	0.0	0	
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	254	0	600000000.0	0.0	0	0	0.006	0.0	0.0	0.0	0	
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	254	0	850400000.0	0.0	0	0	0.010	0.0	0.0	0.0	0	

Figure 3: Train Data

The figure 2 and figure 3 shows the test and train dataset as UNSW-NB15$_1$.csv, UNSW-NB15$_2$.csv. The total number of training and testing records includes 175,341 and 82,332 from different types of normal and attack patterns. The dataset is freely accessible for research and academic purpose in perpetuity. The access to dataset for commercial purpose should be agreed on copyright terms registered authentically with authors.

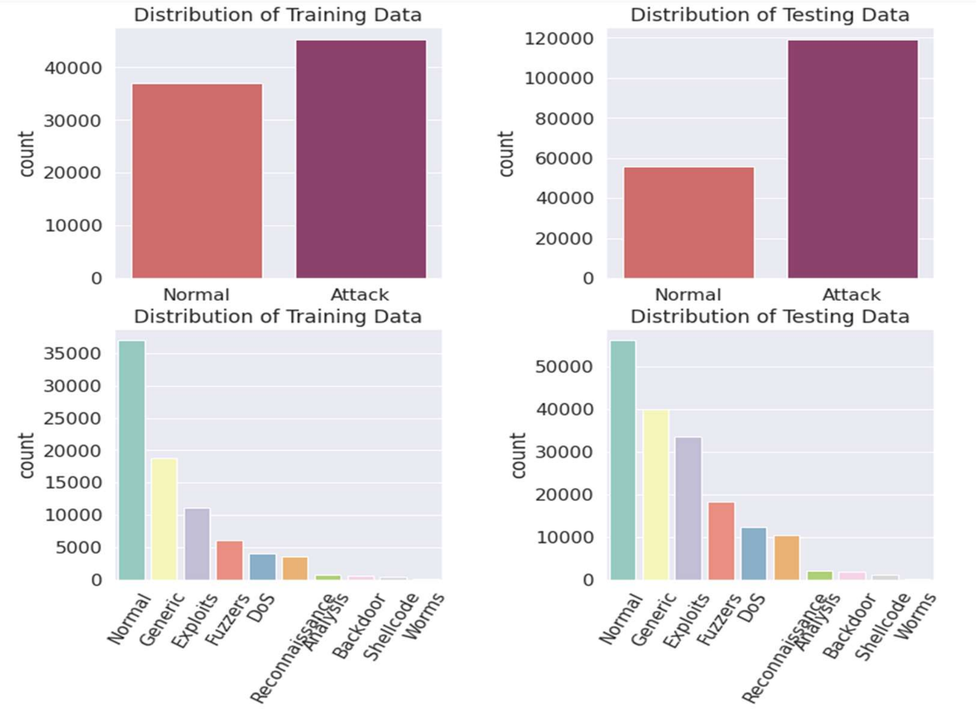


Figure 4: Dataset Description

The figure.4 shows the distribution of training dataset and testing dataset on the basis of different types of normal and attack behavior's. The x-axis in all the bar graphs shows captured normal, attack patterns and represents nine different attacks as Normal, Generic, Exploits, Fuzzers, Denial of

Services, Reconnaissance, Backdoor, shellcode and worms and y-axis represents count of the number of records.

4. Traditional Regression Analysis

Regression is the methodology to find the functional relation between two or multiple variables. The fitness of straight line over the variables defines the linear regression of that relation which is simply defined by straight line.

The fitted line equation, which shows the average relationship between two variables and predict the value P of dependent variable for the given value of Q variable in equation (1), (Benisha and Ratna., 2020)

$$X=p+qY, \tag{1}$$

where X and Y are dependent and independent variable, p is bias and Error Rate is X actual points-X. The error is difference of points from line.

For example, If we have value for two variables say P and Q, and it is needed to fit a line over the top of the points P and Q and if it is required to check how these points are fitted to line then we compute the error value as in equation (2), (Benisha and Ratna., 2020)

$$\text{Error Value}= X_{\text{actual}} - X, \tag{2}$$

The error is the distance between the fitted line and the point. Here for second entry and third entry we get the error of 1 and -1.

To remove the negative entries so that the overall result should not be zero or wrong we will square the error values. The line is the best fit if the magnitudes of deviations are low for an individual case, for large values of deviation the method of "least squares" is used to minimize the squared deviation in equation (3), (Benisha and Ratna., 2020).

Thus for minimization,

$$\sum(y-Y)^2, \tag{3}$$

where y is the variable value and Y is the line.

The coefficients of least square regression line equation a and b is given by equation (4,5), (Benisha and Ratna., 2020).

$$\sum y = Ma + b\sum x, \tag{4}$$
$$\sum xy = a\sum x + b\sum x^2, \tag{5}$$

doi:10.20944/preprints202112.0323.v1

Here M includes total number of cases, These equation (4) and (5) are “normal equations”, for simplification,

$$b = \frac{\sum xy - \frac{(\sum x)\,(\sum y}{M}}{\sum x^2 - \frac{(\sum x)^2}{M}}, \tag{6}$$

$$a = \frac{\sum y}{M} - b\,\frac{\sum x}{M}, \tag{7}$$

if line passes through (x,y)

$$b = \frac{\frac{\sum xy}{M} - \overline{x\bar{y}}}{\frac{\sum x^2}{M} - \overline{x^2}}, \tag{8}$$

Here denominator is variance of variable x and the numerator is defined as the covariance of variable x and y. This shows the line of least squared deviation and shows the fitness of regression line on x using equation (6,7,8,9), (Benisha and Ratna., 2020).

$$a = \bar{y} - b\bar{x}, \tag{9}$$

The equation (8) relation between variance and covariance is written as equation (10), (Benisha and Ratna., 2020).

$$b = \frac{cov(x,y)}{var\ x}, \tag{10}$$

The value of x variable is predicted by using given value for y variable, which is represented as the regression of x on y and the equation becomes,

x =a+by, the xy with b shows the slope of regression line for x on y, Similarly the yx with b shows the slope for regression for y on x. The equation b_{xy} is given as equation (11,12,13), (Yao et al., 2021)

$$b_{xy} = \frac{cov(x,y)}{var\ y}, \tag{11}$$

The regression analysis for grouped data follows the same procedure as for simple linear regression for two variables apart from the criteria that for all the data items falls in one group is approximated to have a value equal to mid-point value of specified group where data is organized in two-way matrix as equation (12).

$$b_{yx} = \frac{\frac{\sum fxy}{M} - \frac{\sum fx}{M} . \frac{\sum fy}{M}}{\frac{\sum fx^2}{M} - (\frac{\sum fx}{M})^2}, \tag{12}$$

Here numerator is the frequency and the count of those items which have their values in the specified group for the term variable x and y.

$$= \frac{\frac{\sum fd_x d_y}{M} - \frac{\sum fd_x}{M}\ \frac{\sum fd_y}{M}}{\frac{\sum fd_x^2}{M} - (\frac{\sum fd_x}{M})}, \tag{13}$$

Linear Regression is a supervised learning based algorithm which target prediction values considering independent variables, this defines the relationship between multiple variables and forecasting.

The relationship between multiple variables and forecasting. The relationship between dependent and independent variable differs the regression model. The dependent variable (p) is predicted on the basis of given independent variable (q) to define a linear relationship between (p-input) and (q-output)

The hypothesis function for linear regression is defined in equation (14), (Yao et al., 2021):

$$q = \propto_1 + \propto_2 \times p, \tag{14}$$

Here p is a univariate input training data and q is data labels. The regression training model should fit a line to predict the data labels for a particular value of one input variable.

The model fits the best when the accurate value of $\propto_1$ and $\propto_2$ is computed, it predicts the value of q on the basis of input value p.

The best fit values of $\propto_1$ and $\propto_2$ is computed or updated by using cost function.

To achieve the best fit the regression model objects to predict data labels (q) in such a way that the difference among true value and predicted value is minimum. The cost function is used for updating the best fit values of $\propto_1$ and $\propto_2$ to minimize the error rate using equation (15,16,17), (Yao et al., 2021).

$$\text{mini}\frac{1}{n} \sum_{k=1}^{n}(predition_{k}-q_{k})^2, \tag{15}$$

$$M=\frac{1}{n}\sum_{k=1}^{n}(prediction_{k}-q_{k})^2, \tag{16}$$

$$\text{RMSEV}=\sqrt{\frac{\sum_{k=1}^{n}(q_{prediction,k}-q_{k})^2}{k}}, \tag{17}$$

Here the root mean square of difference of (q) predicted value and (q) real value is cost function (M). The cost function is to calculate the difference in true value and predicted value. Root Mean Squared Error Value is defined as the square root of mean of error value square.

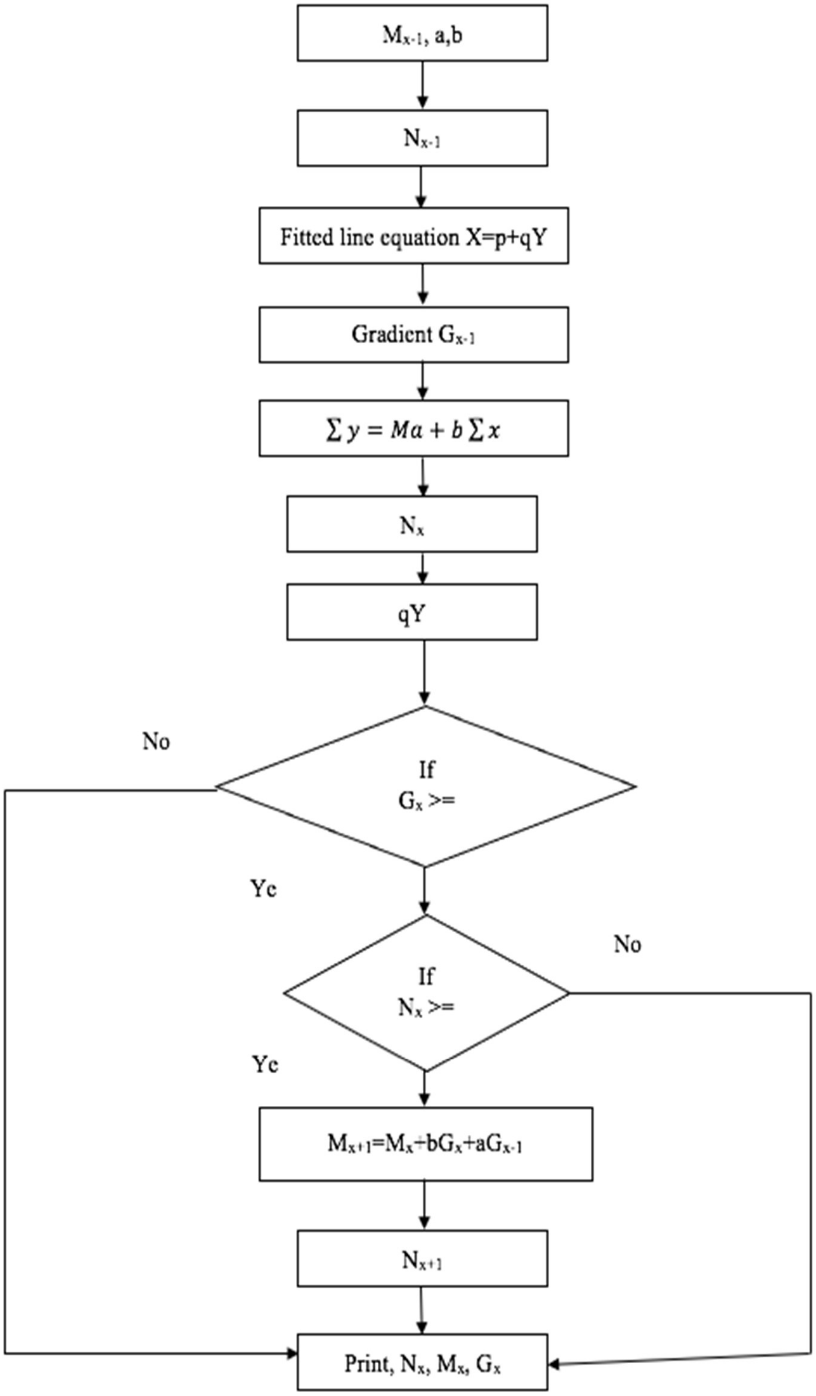


Figure 5: HNADAM-SGD flow chart.

The figure 5 represents the HNADAM-SDG flow chart having tendency to lower itself during convergence and is represented using variable momentum and step size as equation (18), (Yao et al., 2021).

$$M_{ij}(k+1) = M_{ij}(k) + \Delta M_{ij}(k) + \boldsymbol{\beta}\Delta\, M_{ij}(k-1), \tag{18}$$

Here N is the error function, M_{ij} weight and ΔM_{ij} is the continuous change in the value of weight for each loop. The objective is to minimize the gradient of N (error function).

4. Proposed Hyperparameter Optimization Algorithm

Linear regression is used to measure the relationship between population of variables based on collected paired samples. The correlation of variables is determined to further compute the mathematical relationship so as to predict the value of one variable based on other variable and to observe the change in the value of variable on the basis of other variable.

Let the study variable is ‘p’ and explanatory variable is ‘q’, the items for variable ‘p’ and ‘q’ is paired as, (q_1,p_1), (q_2,p_2),……, (q_m,p_m), the simplest form of relation is a linear relationship which is defined as $p = x+yq$, the line is required to be fit along the points. To make the line best fit the values of a and b is to be maintained accurately.

Least square estimation is made for the best fitness of variables. The exact relation between variables is found by approximating the relationship along the line, so the line equation can be rewrite as, $\boldsymbol{\overline{p} = x + yq}$, Here $\boldsymbol{\overline{p}}$ is the predicted or estimated value of p. The exact relationship among variable is defined as p= x+yq=error, the error here is the difference between predicted value and actual value for (q_1,p_1), (q_2,p_2),……, (q_m,p_m), the error is computed as $(p_j - q - xq_j)$ for I = 1, 2,…., n. It is required to find the value of a and b for which the difference is minimum for best fit.

In least square estimation the summation of squared residuals are minimum. The differentiation is separately obtained for a and b having derivative be zero. The a and b estimates are defined in equation (19,20,21), (Yao et al., 2021).

$$\boldsymbol{\widehat{a} = \overline{y} - \widehat{b}\overline{x},} \tag{19}$$

$$\boldsymbol{\widehat{y} = \frac{\sum_{j=1}^{m}(a_j-\overline{a})(b_i-\overline{b})}{\sum_{i=1}^{n}(a_i-\overline{a})^2} = \frac{SSXY}{SSX}}, \tag{20}$$

Least square estimation (LSE) r for b is,

$$r = \boldsymbol{\widehat{y}\sqrt{\frac{\sum_{b=1}^{m}(q_i-\overline{q})^2}{\sum_{j=1}^{m}(p_i-\overline{p})^2}} = \widehat{y}\sqrt{\frac{SSX}{SSY}}}\ , \tag{21}$$

Here r and $\boldsymbol{\widehat{y}}$ is same, Coefficient of determination is used when there is no linear relationship between the variables, the residual quantities as in equation (22), (Yao et al., 2021).

$$\boldsymbol{p_i - x - yq_j}\ \text{, where (j=1,2,……,m)} \tag{22}$$

The good fitted linear model have small magnitude of the residuals , The variability of Y using X shows how the change in values of Y affect the prediction of values, the variance is computed as in equation (23,24,25,26,27,28,29,30), (Yao et al., 2021).

$$\frac{1}{m}\sum_{j=1}^{m}(\boldsymbol{p_i}-\overline{\boldsymbol{p}})^{2}, \text{ this is partitioned as,} \tag{23}$$

$$\frac{1}{m}\sum_{j=1}^{m}(\boldsymbol{p_j}-\overline{\boldsymbol{p}})^{2}=\frac{1}{m}\sum_{j=1}^{m}(\boldsymbol{p_j}-\widehat{\boldsymbol{p}_j}+\widehat{\boldsymbol{p}_j}-\overline{\boldsymbol{p}})^{2}=\frac{1}{m}\sum_{j=1}^{m}(\boldsymbol{p_j}-\widehat{\boldsymbol{p}}_j)^{2}+\frac{1}{m}\sum_{j=1}^{m}(\widehat{\boldsymbol{p}_j}-\overline{\boldsymbol{p}})^{2}, \tag{24}$$

$$\sum_{j=1}^{m}(\boldsymbol{p_j}-\overline{\boldsymbol{p}})^{2}=\sum_{j=1}^{m}(\boldsymbol{p_j}-\widehat{\boldsymbol{p}_j})^{2}+\sum_{j=1}^{m}(\boldsymbol{p_j}-\overline{\boldsymbol{p}})^{2}, \tag{25}$$

$$\frac{1}{m}\sum_{j=1}^{m}(\boldsymbol{p_j}-\widehat{\boldsymbol{p}_j})(\widehat{\boldsymbol{p}_j}-\overline{\boldsymbol{p}})=\boldsymbol{0}, \tag{26}$$

here $\boldsymbol{p_j}-\widehat{\boldsymbol{p}_j}$ is the residual.

Sum of squares dues to errors (SSE) is defined as

$$\sum_{j=1}^{m}(\boldsymbol{p_j}-\widehat{\boldsymbol{p}_j})^{2}, \tag{27}$$

Sum of squares due to regression (SSR) is given as

$$\sum_{j=1}^{m}(\widehat{\boldsymbol{p}_j}-\overline{\boldsymbol{p}})^{2}, \tag{28}$$

Total variability in p is,

$$\mathrm{P(SST} = \frac{1}{m}\sum_{j=1}^{m}(\boldsymbol{p_j}-\overline{\boldsymbol{p}})^{2}), \tag{29}$$

$$R^2 = \mathrm{SSR/SST} = 1\text{-}(\mathrm{SSE/SST}), \tag{30}$$

$0 \leq \boldsymbol{R^2} \leq \boldsymbol{1}$, when R^2 is near to 1 then it is to be found that themost of the variability falls in y and the model is best fit and when R^2 is close to 0 then it defines that there is no much variability in Y and the model is not good fit.

The variability of Y values around the predicted regression line is measured by estimating standard error as $S_{pq} = \sqrt{\frac{\boldsymbol{SSE}}{\boldsymbol{m-2}}}$, when predicted outcomes is close to the observed values then the standard error is lesser, for the hypothesis as in equation (31,32,33,34),(Sumaiya Thaseen et al., 2021)

$$\mathrm{H}\boldsymbol{\alpha}(\boldsymbol{A}) = \propto_{\boldsymbol{0}} + \propto_{\boldsymbol{1}} \boldsymbol{A}, \tag{31}$$

We need to fit for training data. The cost function is used as Mean squared error:

$$\text{Mean Squared Error} = \frac{1}{k}\sum_{j=1}^{k}(q_j - q_{vector\ j})^2, \tag{32}$$

This error is minimized by using gradient descent optimization algorithm as:

$$\propto_k = \propto_k - \theta \frac{\partial}{\partial \propto_k} B(\propto_0, \propto_1)\ for\ k = 0\ and\ k = 1, \tag{33}$$

Testing of hypothesis using correlation coefficient along slope is given by test statistic,

$$T_c = \frac{\hat{b}}{\sqrt{\frac{SSE}{(n-2)SSX}}}, \tag{34}$$

Testing the hypothesis H$_o$ where b=0 and H$_a$ $where\ b \neq 0$.

To predict the Y value using fitted line the confidence interval and prediction interval is used which defines the point estimation of population.

Confidence interval as in equation (35), (Sumaiya Thaseen et al., 2021)

$$\hat{p}_j \pm s_{\propto/2}(d.f. = m - 2)S_{pq}\sqrt{\frac{1}{m} + \frac{(q_j - \bar{q})^2}{SSX}}\ , \tag{35}$$

Prediction Interval as in equation (36), (Sumaiya Thaseen et al., 2021)

$$\hat{p}_j \pm s_{\propto/2}(d.f. = m - 2)S_{pq}\sqrt{1 + \frac{1}{m} + \frac{(q_i - \bar{q})^2}{SSX}}\ , \tag{36}$$

To predict the individual value of variable for the given value is obtained by using prediction interval.

The gradient descent algorithm uses gradient of function parameters to identify search space. HNADAM-SDG is based on NADAM Nestrov Momentum version of gradient descent. The hypertuning is used for attaining the betterment of performance. This algorithm follows negative values of objective parameters to relocate the minimum of function. It measures the displacement in weights with respect to displacement in error. The gradient is also understood as the function of a slope. The gradient is inversely proportional to the steepness of slope.

The learning of model depends on steepness of slope. If the slope tends to zero then the model stops learning is given by equation (37,38,39,40,41,42,43,44,45,46,47,48,49),(Al-Safi et al., 2021).

$$\boldsymbol{\overline{p} = nq + b}, \tag{37}$$

$$\text{Error} = \boldsymbol{\overline{p} - p}, \tag{38}$$

$$\text{Error}^2 \text{ (Cost Function)} = \text{Estimated Value – Actual Value} = ((\boldsymbol{\overline{p} - p})^{\boldsymbol{2}}, \tag{39}$$

$$\text{L.F} = ((\text{nq+b})\text{-b})^2\text{, loss function=L.F}, \tag{40}$$

$$\text{L.F} = ((\text{nq+b})\text{-p})^2, \tag{41}$$

$$\text{L.F=f(n,b)}, \tag{42}$$

$$\text{Total Loss= T.L}_{\text{Total}} = \boldsymbol{\frac{1}{M}} \boldsymbol{\sum_{j=1}^{M}}((\boldsymbol{n_j + b}) - \boldsymbol{p_j})^{\boldsymbol{2}}, \tag{43}$$

$$\boldsymbol{\frac{\partial s}{\partial n}} = \boldsymbol{\frac{\partial}{\partial b}}(\boldsymbol{\sum_{j=1}^{M}}(\boldsymbol{nq_j + b}) - \boldsymbol{p_j})^{\boldsymbol{2}}), \tag{44}$$

$$\boldsymbol{\frac{\partial s}{\partial b}} = \boldsymbol{\frac{\partial}{\partial b}} (\boldsymbol{\sum_{j=1}^{M}}((\boldsymbol{nq_j + b}) - \boldsymbol{p_j})^{\boldsymbol{3}}), \tag{45}$$

Gradient descent with respect to “b”

$$\boldsymbol{\frac{\partial}{\partial b}} = \boldsymbol{\frac{\partial}{\partial b}} (\boldsymbol{\sum_{j=1}^{M}}((\boldsymbol{nq_j + b}) - \boldsymbol{p_j})^{\boldsymbol{2}}), \tag{46}$$

$$\boldsymbol{\frac{\partial}{\partial b}} = \boldsymbol{2} (\boldsymbol{\sum_{j=1}^{M}}((\boldsymbol{nq_j + b}) - \boldsymbol{p_j})), \tag{47}$$

Gradient descent with respect to “n” is given by equation (48,49), (Sumaiya Thaseen et al., 2021)

$$\boldsymbol{\frac{\partial s}{\partial n}} = \boldsymbol{\frac{\partial}{\partial n}} (\boldsymbol{\sum_{j=1}^{M}}((\boldsymbol{nq_j + b}) - \boldsymbol{p_j})^{\boldsymbol{2}}), \tag{48}$$

$$\boldsymbol{\frac{\partial s}{\partial n}} = \boldsymbol{2} (\boldsymbol{\sum_{j=1}^{M}}((\boldsymbol{nq_j + b}) - \boldsymbol{p_j})), \tag{49}$$

Algorithm: HNADAM-SDG

Initialize “m” and “c” in the start with random number.
Calculate Gradient
Update calculated gradient with respect to “m” and “c” using equation (50,51,52), (Khan ., 2021)

$$\frac{\partial s}{\partial n} = \frac{\partial}{\partial n}\left(\sum_{j=1}^{M}\left(\left(nq_j + b\right) - p_j\right)^2\right), \tag{50}$$

$$\frac{\partial s}{\partial n} = 2\left(\sum_{j=1}^{M}\left(\left(nq_j + b\right) - p_j\right)\right), \tag{51}$$

The learning rate is multiplied by “n” and “b”

Update the value of “n” and “b” for every step

The class-conditional densities are not being modelled in logistic discrimination

Model Parameter $\boldsymbol{\beta}$

Parameter Initialization

Normal distribution $\boldsymbol{\beta} \sim \boldsymbol{M}(\boldsymbol{Q}, \boldsymbol{\theta}^2)$

Initial vector n=0

Initial vector u=0

Initialize steps $\boldsymbol{T}$=0

Initialize convergence parameter as Boolean= F

While Boolean= = F

Do shuffle the training set $\boldsymbol{T}$ for each mini-batch b⊂ $\boldsymbol{T}$ do update step $\boldsymbol{T} = \boldsymbol{T} + \mathbf{1}$

Compute gradient vector G=$\nabla_{\beta}\boldsymbol{\mathcal{L}}(\boldsymbol{\beta}; \boldsymbol{b})$ on the mini-batch b

Update Vector n

m=$\boldsymbol{\alpha_1} . \boldsymbol{n} + (\mathbf{1} - \boldsymbol{\alpha_1}). \boldsymbol{G}$

Update Vector n=$\boldsymbol{\alpha_2} . \boldsymbol{u} + (\mathbf{1} - \boldsymbol{\alpha_2}). \boldsymbol{G} \odot \boldsymbol{G}$

Rescal Vector ∃= $\frac{\mathbf{n}}{1-\alpha_1^T}$

Rescal Vector U=U/(1-$\boldsymbol{\alpha_2^T}$)

Update Variable $\boldsymbol{\beta} = \boldsymbol{\beta} - \frac{m}{\sqrt{u+x}} \odot (\boldsymbol{\alpha_1}.\mathrm{m}+\frac{(1-\alpha_1)}{1-\alpha_2^T}.\boldsymbol{G})$, (52)

End for

If convergence condition holds then

Boolean=T

End if

End while

Return model variable $\boldsymbol{\beta}$

The iterative loop, For i=0,------,c

$\boldsymbol{P_i} \leftarrow \boldsymbol{rand}(-\mathbf{0.10}, \mathbf{0.10})$ to compute random variable ranging between -0.10 to 0.10

The iteration is made:

Repeat

For i=0,-----,

$$\Delta P_i \leftarrow 0$$

```
For i=0,-----,c
    0← $0 + P_i A_i^T$
    B← $sigmoid(0)$
    For i=0,-----,c
    $\Delta P_j \leftarrow \Delta P_j + (R^T - B)A_i^T$
    For j=0,-----,d
```

```
$P_i \leftarrow P_i + n\Delta P_i$
Until convergence repeat the iteration
The iterative loop For j=1,----,v
For i=0,----,c
    $P_{ij} \leftarrow rand(-0.10,0.10)$

The iteration is made using
Repeat
For j=1,----,v
    For i=0,----,c
        $\Delta P_{ij} \leftarrow 0$
For T=1,----,m
    For j=1,-----,v
        $Q_i \leftarrow 0$
    For i=0,-----,c
        $Q_i \leftarrow Q_i + P_{ij}A_i^T$
For i=1,-----,v
        $B_j \leftarrow exponential(Q_i)/\sum_v exponential(Q_v)$
For j=1,-----,v
    For i=0,-----,c
        $\Delta P_{ij} \leftarrow \Delta P_{ij} + (R_i^T - B)x_j^T$
For j=1,-----,v
    For i=0,-----,d
        $P_{ij} \leftarrow P_{ij} + n\Delta P_{ij}$
    Repeat the iteration until convergence.
```

Training T; Learning Rate N; Normal Distribution Std θ; Decay Parameters α_1, α_2

5.Results and Discussion

The performance of HNADAM-SDG algorithm is compared with logistic regression, ridge classifier and ensemble techniques. The UNSW-NB15 training and testing data is used to train and test the IDS model on the basis of occurances of attacks. The performance measures are accuracy and error rate as true positive, true negative, false positive and false negative is driven from confusion matrix.

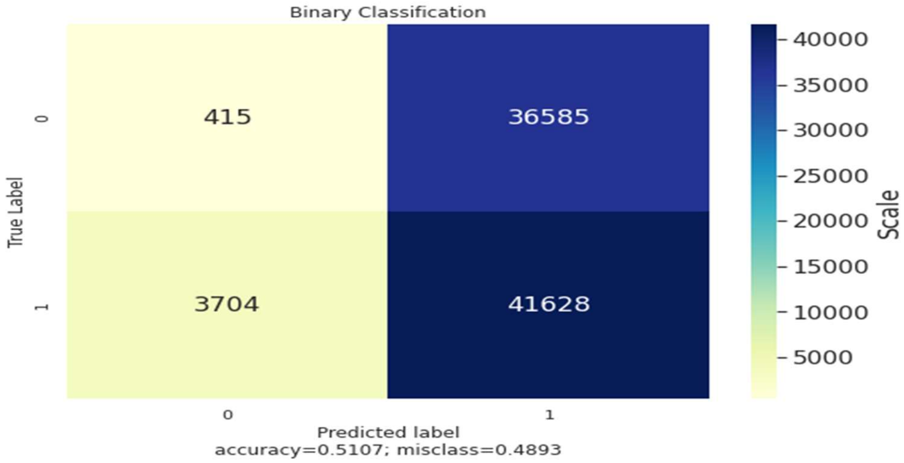
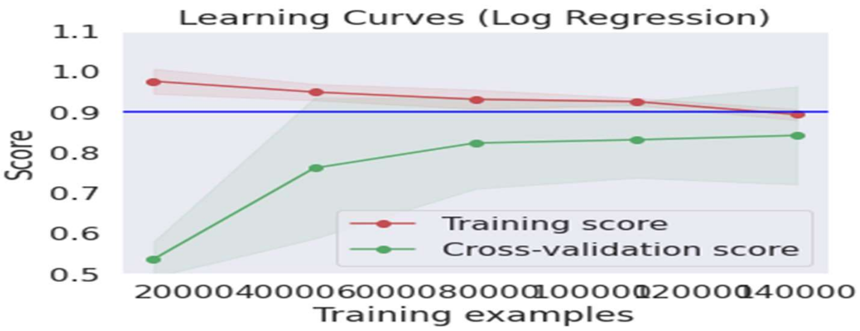
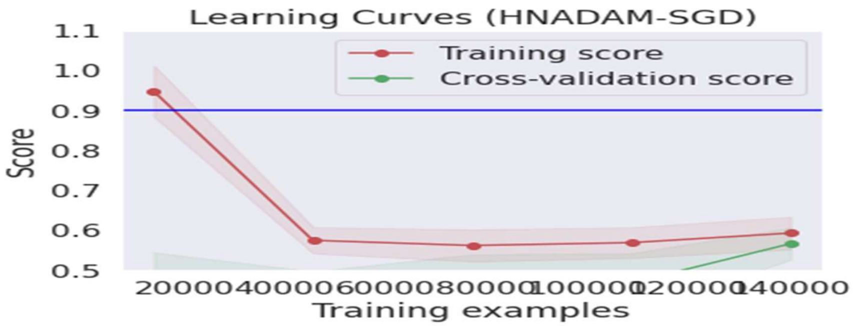


Figure 6: Binary Classification Confusion Matrix

Figure 6 shows the binary classification of data using confusion matrix which is two by two matrix consist of outcomes produced by binary classifier as overall accuracy, error-rate, sensitivity, precision and specificity. The binary classifier produces result with labels as 0/1 and Yes/No. The instances of all the test data is predicted using classifier as true positive, true negative, false positive and false negative. The matrix derives error rate and accuracy as primary measure. Here the confusion matrix computes accuracy as 0.51 and error rate as 0.489, the matrix is built between true label and predicted label with labels as 0/1 and having data scale.





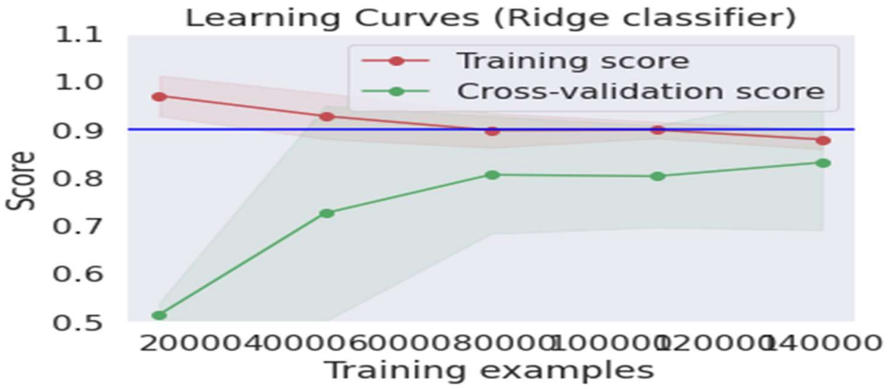


Figure 7: Learning Curves

The figure 7 shows the learning curves for logistic regression, ridge classifier and HNADAM-SDG techniques. The learning curve is measure by taking instance from training dataset to measure a model performance by computing error rate over validation dataset. The best fit algorithm is having zero error rate to fit data points. The error rate of the model varies as the size of the training instance fluctuates. The curve shows the change in error, training score and cross validation score as the training instance changes.

Table 1: Performance Matrix

Logistic Regression				
	Precision	Recall	F1-Score	Support
0	0.48	0.82	0.61	3971
1	0.99	0.96	0.98	78361
Accuracy			0.96	82332
Macro Average	0.74	0.89	0.80	82332
Weighted Average	0.98	0.96	0.96	82332
Ridge Classifier				
	Precision	Recall	F1-Score	Support
0	0.70	0.72	0.71	3971
1	0.99	0.98	0.98	78361
Accuracy			0.98	82332
Macro Average	0.85	0.86	0.85	82332
Weighted Average	0.98	0.98	0.98	82332
HNADAM-SDG				
	Precision	Recall	F1-Score	Support
0	0.71	0.70	0.71	3971
1	0.99	0.98	0.99	78361
Accuracy			0.99	82332
Macro Average	0.87	0.88	0.86	82332

Weighted Average	0.98	0.98	0.98	82332
Ensemble				
	Precision	Recall	F1-Score	Support
0	0.49	0.80	0.60	3971
1	0.98	0.98	0.98	78361
Accuracy			0.97	82332
Macro Average	0.76	0.88	0.86	82332
Weighted Average	0.98	0.96	0.97	82332

The table 1 represents the performance measures for different algorithms as logistic regression, ridge classifier, HNADAM-SDG and Ensemble technique applied over the dataset to predict the emerging attack patterns. The performance is measured by computing precision, recall, F1-score and support value for 0/1, macro average, accuracy and weighted average.

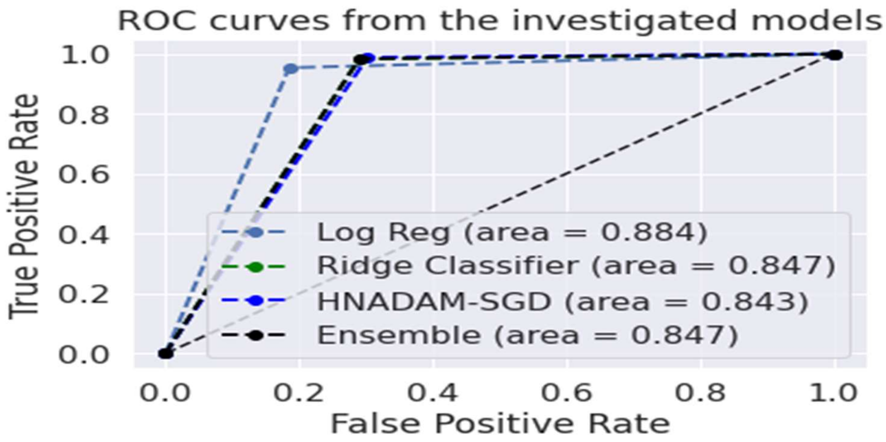


Figure 8: Roc Curve Areas

The figure 8 shows the ROC curve areas for logistic regression, ridge classifier, HNADAM-SDG and ensemble algorithm to find the area under curve of receiver characteristic operator (ROC) to find plots denoting TPR and FPR considering threshold value by obtaining probability curve to separate signal from noise. The higher value of area under curve represents the better performance of algorithm.

AUC=1, denotes that all positive and negative classes are pointed correctly using classifier.

0.5<AUC<1, shows that the chance of distinguishing between positive and negative class is higher.

AUC=0.5, shows that the classifier is not able to distinguish between positive and negative classes.

The higher the value of X-axis shows the higher number of false positives while the Y-axis shows the higher number of true positives.

doi:10.20944/preprints202112.0323.v1

Table 2: Roc-Auc Score

Roc-Auc Score		
1.	Logistic Regression	0.884319
2.	Ridge Classifier	0.846685
3.	HNADAM-SDG	0.843447
4.	Ensemble	0.847184

The table 2 shows the area under curve (AUC) of receiver characteristic operator (ROC) for different algorithms applied over dataset. The area under curve graph is used to represent the best fir model for testing and training.

6.Research Limitations

- Hyper-parameters optimization algorithm performance depends on various factors as, number of hidden layers, number of units per-layer, activation function, dropout amount, regulizer learning rate and weight decay.

- The non-optimal setting of hyperparameters will drastically affect the algorithm performance varies from extremely low learning rate to very large learning rate.

- The hyper-tuning approach varies depending type of dataset, nature of dataset and its size as there is no well-defined formula to find hyperparameters.

- The criticality is to choose what number of parameters are going to test, due to which performance get affected as extremely low learning rate of (1-e^{-5}) or very large learning rate of 10   by opting wrong hyperparameters.

- There exist no well-defined formula to find hyperparameters as, it depends on algorithm type, dataset and dataset size.

- The performance of algorithm varies with the change in dataset and parameters.

7.Conclusion and Future Scope

In this paper, IDS model is determined using hybridization of nestrov-accelerated adaptive moment estimation–stochastic gradient descent (HNADAM-SDG) algorithm. The performance of the algorithm is compared with other classification algorithms as logistic regression, ridge classifier and ensemble algorithm by adapting feature selection and optimization techniques. The algorithm is used for testing and training UNSW-NB15 dataset. The HNADAM-SDG techniques is used to measure the relationship between population of variables based on collected paired samples. The correlation of variables is determined to further compute the mathematical relationship so as to predict the value of one variable based on other variable and to observe the change in the value of variable on the basis of other variable. The best fit algorithm is having zero error rate to fit data points. The error rate of the model varies as the size of the training instance fluctuates. The performance is visualized using learning curves and AUC-ROC curve areas. The performance of the HNADAM_SDG algorithm is compared with other classification algorithms as logistic regression, ridge classifier and ensemble algorithm by adapting feature selection and optimization techniques. In future the IDS helps in mitigating the impact of malicious activities over emerging information sharing platforms. The IDS will evolve in diversified research areas as,

- The IDS for Internet of Things (IoT) which is the expanding surface for attackers. This includes breaching security of automotive, wearables and connected devices.
- IDS for cyber insurance which is the upcoming ideology to receive attention to mitigate the damages from upcoming data loss, sabotage and theft events.

References

Ahmad, I. 2015,'Feature selection using particle swarm optimization in intrusion detection',*International Journal of Distributed Sensor Networks*, Vol.11, No.10, pp.806-954.

Al-Safi, A. H. S., Hani, Z. I. R., & Zahra, M. M. A. 2021, 'Using A Hybrid Algorithm and Feature Selection for Network Anomaly Intrusion Detection', *Journal of Mechanical Engineering Research and Developments*, Vol.44, No.4, pp.253-262.

Aleesa, A., Younis, Mohammed, A. A., & Sahar, N.2021, 'Deep-intrusion detection system with enhanced unsw-Nb15 dataset based on deep learning techniques', *Journal of Engineering Science and Technology*, Vol.16, No.1, pp.711-727.

Alkafagi, S. S., & Almuttairi, R. M.2021, 'Enhance density peak clustering algorithm for anomaly intrusion detection system, '*Periodicals of Engineering and Natural Sciences (PEN),* Vol.9, No.2, pp.965-975.

Alrajeh, N. A., & Lloret, J.2013, 'Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks', *International Journal of Distributed Sensor Networks*, Vol.9, No.10, pp.351-047.

Azarkasb, S. O., Kashi, S. S., & Khasteh, S. H.2021, 'A Network Intrusion Detection Approach at the Edge of Fog', *International Computer Conference, Computer Society of Iran (CSICC)*, pp. 1-6.

Bamhdi, A. M., Abrar, I., & Masoodi, F. 2021, 'An ensemble based approach for effective intrusion detection using majority voting',*Telkomnika*, Vol.19, No.2, pp.664-671.

Benisha, R. B., & Ratna, S. R. 2020, 'Detection of data integrity attacks by constructing an effective intrusion detection system', *Journal of Ambient Intelligence and Humanized Computing*, Vol.11, No.11, pp.5233-5244.

Bhati, N. S., & Khari, M.2021, 'A new ensemble based approach for intrusion detection system using voting', *Journal of Intelligent & Fuzzy Systems*, (Preprint), pp.1-11.

Bhattacharjee, PS, Fujail, AKM, Begum, SA. 2017, 'Intrusion detection system for NSL-KDD data set using vectorised fitness function in genetic algorithm' Adv Comput Sci Tech,Vol.10, No.2, pp.235–246.

Derhab, A., Bouras, A., Senouci, M. R., & Imran, M. 2014,'Fortifying intrusion detection systems in dynamic Ad Hoc and wireless sensor networks', *International Journal of Distributed Sensor Networks*, Vol.10, No.12, pp.608-162.

Drewek-Ossowicka, A., Pietrołaj, M., & Rumiński, J. 2021, 'A survey of neural networks usage for intrusion detection systems', *Journal of Ambient Intelligence and Humanized Computing*, Vol.12, No.1, pp.497-514

Hsu, C. M., Azhari, M. Z., Hsieh, H. Y., Prakosa, S. W., & Leu, J. S.2021, 'Robust network intrusion detection scheme using long-short term memory based convolutional neural networks', *Mobile Networks and Applications*, Vol.*26,* No.3, pp.1137-1144.

Iman, A. N., & Ahmad, T. 2020, 'Data Reduction for Optimizing Feature Selection in Modeling Intrusion Detection System', *International Journal of Intelligent Engineering and Systems*, Vol.13, No.6, pp.199-207.

Khan, M. A. 2021, 'HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System', *Processes*, Vol.9, No.5, pp.800-834.

Li, Y., Ghoreishi, Sm. & Issakhov, A. 2021, 'Improving the Accuracy of Network Intrusion Detection System in Medical IoT Systems through Butterfly Optimization Algorithm', *Wireless Person Communication ,* https://doi.org/10.1007/s11277-021-08756-x.

Liggett, K. K., & Thomas, G. F.2015, 'Determining the Effectiveness of Visualization Techniques for Representing Intrusion Detection System Log Files', In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 59, No. 1, pp. 1392-1396.

Loo, C. E., Ng, M. Y., Leckie, C., & Palaniswami, M. 2006,'Intrusion detection for routing attacks in sensor networks', *International Journal of Distributed Sensor Networks*, Vol.2, No.4, pp.313-332.

Mohd, N., Singh, A. & Bhadauria, H.S. 2021,'Intrusion Detection System Based on Hybrid Hierarchical Classifiers', *Wireless Pers Commun*, https://doi.org/10.1007/s11277-021-08655-1.

Özer, E., İskefiyeli, M., & Azimjonov, J. 2021,'Toward lightweight intrusion detection systems using the optimal and efficient feature pairs of the Bot-IoT 2018 dataset', *International Journal of Distributed Sensor Networks*, Vol.17, No.10, pp.15501477211052202.

Pokuri, S. R. 2021, 'A Hybrid Approach for Feature Selection Analysis on The Intrusion Detection System Using Navi Bayes And Improved BAT Algorithm', *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, Vol.12, No.11, pp.5078-5087.

Pragya, M., Arya, K.V. & Pal, S.H. 2018, 'Intrusion Detection System Against Colluding Misbehavior in MANETs', *Wireless Person Communication*, Vol.100, pp.491–503.

Ren, Z., Tang, Y., & Zhang, W.2021,'Quality-related fault diagnosis based on k-nearest neighbor rule for non-linear industrial processes', *International Journal of Distributed Sensor Networks*, Vol.17, No.11, pp.15501477211055931.

Salih, A. A., & Abdulazeez, A. M.2021, 'Evaluation of classification algorithms for intrusion detection system: A review', *Journal of Soft Computing and Data Mining*, Vol.2, No.1, pp.31-40.

Sasikumar, S.2021, 'Network Intrusion Detection and Deduce System', *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, Vol.12, No.9, pp.404-410.

Singh, D. K., & Shrivastava, M. 2021, 'Evolutionary Algorithm-based Feature Selection for an Intrusion Detection System', *Engineering, Technology & Applied Science Research*, Vol.11, No.3, pp.7130-7134.

Singh, P., Krishnamoorthy, S., Nayyar, A., Luhach, A. K., & Kaur, A. 2019 'Soft-computing-based false alarm reduction for hierarchical data of intrusion detection system', *International Journal of Distributed Sensor Networks*, Vol.15, No.10,1550147719883132.

Tahir, S., Bakhsh, S. T., & Alsemmeari, R. A. 2019, 'An intrusion detection system for the prevention of an active sinkhole routing attack in Internet of things', *International Journal of Distributed Sensor Networks*, Vol.15, No.11, 1550147719889901.

Talita, A. S., Nataza, O. S., & Rustam, Z.2021, 'Naïve Bayes Classifier and Particle Swarm Optimization Feature Selection Method for Classifying Intrusion Detection System Dataset', *Journal of Physics: Conference Series*, Vol. 1752, No. 1, pp. 012-021.

Wu, J., Liu, S., Zhou, Z., & Zhan, M. 2012, 'Toward intelligent intrusion prediction for wireless sensor networks using three-layer brain-like learning', *International Journal of Distributed Sensor Networks*, Vol.8, No.10, pp.243-841.