L_1-Regulated Feature Selection and Deep Learning based Classification for Intrusion Detection System in MANET

Suma R
Research Scholar, Department of Computer Science and Engineering,
Dr. Ambedkar Institute of Technology, Visvesvaraya Technological University, Bengaluru-56

Dr. C. Siddaraju
Professor and Head, Department of Computer Science and Engineering,
Dr. Ambedkar Institute of Technology, Visvesvaraya Technological University, Bengaluru-56.

Abstract: In mobile adhoc network (MANET), the network efficiency and the lifetime of network partly depends upon the competence of the robust intrusion detection system to handle the attacks. The intrusion detection system (IDS) is one of the major component to be integrated with any of the routing protocols so that the data packets reach the intended destination. Due to the infrastructure less architecture of the MANET, it is quite complex to develop an accurate and robust intrusion detection system. In this context, we have made an attempt to come out with an accurate IDS based on L_1-regulated feature selection, and deep learning is applied for classification. The L_1-regulated feature selection is based on Linear Support Vector Machine (LSVM) which is characterized by adding a penalty term to the prediction error in order to reduce the weight of the irrelevant features and to make the relevant features having nonzero weights. For classification purpose, deep learning neural network is initialized with sigmoid activation function in the input and hidden layers. In order to accommodate multiclass classification, the softmax activation function is used in the output layer. In order to demonstrate the suitability of the proposed approach, experiments are conducted on the standard datasets and to argue the predictive capability of the proposed approach. Comparative study is also provided with state-of-the-art approaches and the results are presented considering classification accuracy, precision, recall, f1-scoreand detection rate metrics to exhibit the performance of the proposed approach.

Keywords:L_1-regularization, Feature Selection, Deep Learning, Intrusion Detection System, Mobile Adhoc Network.

I. INTRODUCTION

Mobile adhoc network (MANET) based applications are extensively growing in these days due to their wide-spread applications in several areas including geo-spatial data analytics, weather analytics, disaster management, IoT based systems, military engineering services etc. We have seen equal number of adverse effects in MANET during the data transfer in many of the machine critical applications. This is due to the fact that the MANET is an infrastructure-less networked environment which allows the node movements within the network randomly and hence the topology is changing frequently. Because of the dynamic nature of topology and the absence of a centralized monitoring point, these MANETs become highly susceptible to several kinds of attacks. In addition, MANETs suffer from the security threats such as passive eavesdropping, authorization, denial of service, etc. At the same time, they also suffer from threats due to its wireless properties, such as sleep deprivation, selfishness, black-hole, wormhole, etc. Trust equality is another major problem in case of ad hoc networks i.e., when all the nodes are equally trusted they can easily alter, drop control or drop packets. However, the nodes are self-reliant and self-defense systems which are to be given a protected environment. Efforts are being made by the research community to provide a self-defense mechanism for each node which are part of the actual network. This process is referred as Intrusion Detection System (IDS). Intrusion can be defined as any kind of unauthorized activities that cause damage to a networked-information system. Different kinds of defense mechanisms have been developed to avoid attacks and are broadly classified as *signature based detection* and *abnormality based* detection[14].

Signature-based intrusion detection systems (SIDS): The signature-based intrusion detection systems (SIDS) are based on pattern matching techniques to find a known attack; these are also known as Knowledge-based Detection or Misuse Detection [14]. In SIDS, matching methods are used to find a previous intrusion. In other words, when an intrusion occurs, the signature matches with the signature of a previous intrusion that already exists in the signature database, an alarm signal is triggered. For SIDS, host's logs are inspected to find sequences of commands or actions

which have previously been identified as malware. The increasing rate of zero-day attacks [23] has rendered SIDS techniques progressively less effective because no prior signature exists for any such attacks. Polymorphic variants of the malware and the rising amount of targeted attacks can further undermine the adequacy of this traditional paradigm.

Anomaly-based intrusion detection system (AIDS): AIDS has drawn interest from a lot of scholars due to its capacity to overcome the limitation of SIDS. In AIDS, a normal model of the behavior of a computing system is created using machine learning, statistical-based or knowledge-based methods. Any significant deviation between the observed behavior and the model is regarded as an anomaly, which can be interpreted as an intrusion. The assumption for this group of techniques is that malicious behavior differs from typical user behavior. The behaviors of abnormal users which are dissimilar to standard behaviors are classified as intrusions. Development of AIDS comprises of two phases: the training phase and the testing phase. In the training phase, the normal traffic profile is used to learn a model of normal behavior, and then in the testing phase, a new data set is used to establish the system's capacity to generalize to previously unseen intrusions. AIDS can be classified into a number of categories based on the method used for training namely statistical based, knowledge-based and machine learning based [4]

AIDS methods can be categorized into three main groups: Statistics-based [5], knowledge-based [9, 5], and machine learning-based [2, 15]. The statistics-based approach involves collecting and examining every data record in a set of items and building a statistical model of normal user behavior. On the other hand, knowledge-based approaches try to identify the requested actions from existing system data such as protocol specifications and network traffic instances, while machine-learning methods acquire complex pattern-matching capabilities from training data.

In the recent days, we have seen an excellent works in the field of machine learning for data analytics and hence we were motivated to tackle the intrusion detection problem exploring the recently developed algorithms in machine learning. Numerous feature selection and classification methods for intrusion data are investigated and still there is a challenge with respect to the small sample size, the high curse of dimensionality, and imbalanced class problems. Several feature selection and classification approaches are proposedby different researchers. Garro et al. [11] proposed artificial bee colony based featureselection. Chen et al. [6] introduced particle swarm and decision tree based featureselection and variants of ridge regression methods for classification. Aziz et al.[1] proposed a combination of fuzzy backward feature elimination and independentcomponent analysis for feature selection. Nguyent et al. [17] introduced an aggregatefeature selection method based on statistical ranking methods. Feature selectionbased on game theory is introduced by Sasikala et al. [20]. Moayedikia et al. [16]introduced symmetric uncertainty and harmony search based feature selection. Sharbafet al. [21] proposed filter-based feature selection. According to Ravi et al. [19],the deep learning is playing major role in optimizing many of the algorithms inBiomedical and Health Informatics. In our work, to handle the curse of dimensionalitywith respect to the small sample size, we introduce the L_1-regulated featureselection followed by classification of multiclass networked intrusion data using deep learning techniques. To evaluate the performance of the model, classification accuracy,precision, recall, f1-score and detection rate metrics are used.The rest of the paper is outlined as follows. In section II, we present the proposed methodology. In section III, we present the description of the datasets and experimental results are presented in Section IV. Finally, the conclusions are drawn in Section V.

II. PROPOSED METHODOLOGY

We propose an SVM-based L1-regulated feature selection in IDS forclassification purpose. The method uses linear SVC with parameters **α**regularizerwhich controls the scarcity of the data. The smaller the value of **α** in linear SVC,the fewer features selected. A fully connected neural network architecture is createdalong with its parameters such as the number of epochs, batch size, and the activationfunction which are initialized to sigmoid in the input and hidden layers of the network.Since all the datasets considered in this work are multiclass, the softmax activationfunction is initialized in the output layer to yield the ratio of the probability of a givenclass to the summation of all classes in a given dataset. Moreover, the random state isinitialized to a constant seed value 7 so as to keep results of the model reproducible.The workflow of theproposed approach is shown in Fig. 1, which shows the steps carried out mainly forfeature selection, splitting of the data into training and test data, model creation,evaluation, and visualization are described in the following subsection.

L1-Regularized Feature Selection: In the field of machine learning and artificial neural networks, regularization is an important technique to control the model complexity and hence to select to the suitable discriminative features. In this work, L_1-regularized feature selection approach is explored for feature selection. It applies shrinking strategy by adding penalizing term to the least square errors in a linear regression and hence to assign zero coefficient to the

irrelevant features to discard them from the model. Only nonzero coefficient variables are considered so as to minimize prediction error by tackling over-fitting and simplified the model complexity and computationally stability. The L_1-regularization is a method which helps in reducing the complexity of the model by adding a penalty term as given in Equation 1 and to the least square errors as given in Equation 2. In addition, it helps in lessening some of the weights of data points so as to avoid over-fitting. It allows removing certain features from the model if they are not helpful in training the model.

$$p = \alpha \sum_{i=1}^{d} |w| \quad \text{... (1)}$$

Here, $\boldsymbol{p}$ is the magnitude of the penalty, $\boldsymbol{d}$ is the dimension of the features, $\boldsymbol{\alpha}$ is the control parameter, and $\boldsymbol{w}$ is the weight of each feature.

$$E(w) = \sum_{j=1}^{n} \left(y_j - \sum_{i=0}^{d} w.x_i\right)^2 \text{... (2)}$$

L_1-regularized feature selection is the newly introduced feature selection method to the field of MANET. It works with a classifier model such as SVMs and logistic regression to select an optimal number of features. L_1-based feature selection uses LSVM to fit the data and returns the best fit hyper-plane that divides the data into categories. It uses local optimal solutions to remove features with zero coefficients. It uses the control parameter α to control the scarcity of the data. Scarcity allows a few features in a matrix to have large nonzero coefficient values. As the control parameter $\boldsymbol{\alpha}$ gets maximum value such as α = 0.1, α = 0.01, many features are selected and as the value of $\boldsymbol{\alpha}$ become small such as α = 0.001, α = 0.0001 up to a certain limit which gives the optimal features

$$E(w) = \sum_{j=1}^{n} \left(y_j - \sum_{i=0}^{d} w.x_i\right)^2 + \alpha \sum_{i=1}^{d} |w| \text{ ... (3)}$$

Here E is the error, w is a weight coefficient, y is the label, x is the input features, α is the controller parameter, d is the dimension of features, and n is number of samples.

*Fully Connected Neural Network Method for Multiple type Intrusion Detection Classification:*In our work, a fully connected neural network model is proposed. The model contains the conventional structure of the neural network, namely, input, hidden, and output layers, except it is deep in the sense number of hidden layers goes up to five layers to make it deep learning. The parameters such as the number of nodes in the input layer which is set equal to the number of features in the input data are used. The Kernel initializer parameters are set to the normal distribution to show if data are normally distributed. The kernel at each layer is initialized as a normal distribution and activation functions are sigmoid in the input and hidden layers and softmax in the output layer. Since multiclass classifier is working in terms of numerous binary classifiers in the hidden layers it is sufficient to assign sigmoid activation function in the hidden layer and due to the reason that all intermediate binary classifiers are integrated at the output layer, softmax is initialized to accommodate the multiclass behavior of the data. It shall be noted here that the number of neurons at the output layer is equal to the number of classes in the dataset. The *argmax* function is applied to get the predicted classes. The *argmax* checks the input values of each class and picks the maximum probability as an index of the particular class. To train and test the model, the cross-validation (CV) based method is employed which divides the full dataset into $\boldsymbol{k}$ number of folds. In a CV method, the data is nearly evenly distributed among the folds and each fold has a chance of being test data which helps the model to overcome over-fitting. The fold-1 is a test case in the first phase, fold-2 in the second phase, and fold-$\boldsymbol{k}$ is a test case in the $\boldsymbol{k}^{th}$ phase. The ultimate goal of this method is to create out-of-sample prediction in $\boldsymbol{k}$ folds CV and $\boldsymbol{k}$ neural network models.

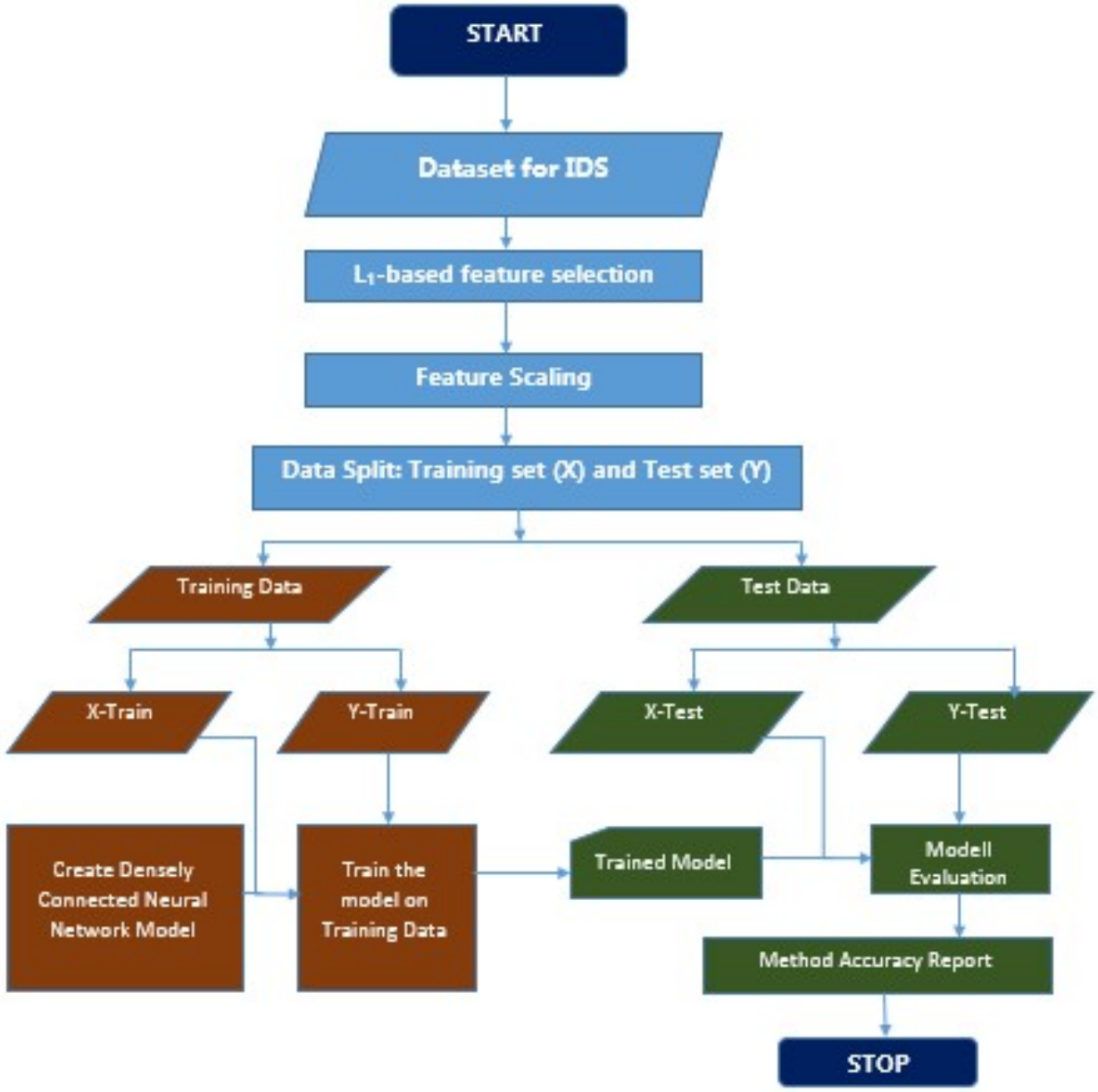

Fig. 1. Architecture of the proposed intrusion detection approach.

The proposed model takes an input vector $x = [x_1, x_2, x_3 \ldots x_n]$ and each input vector is multiplied by its corresponding weight. Hence, the weight vector for the input data is represented as $w = [w_1, w_2, w_3 \ldots w_n]$ and the bias $\boldsymbol{b}$ is added to the weighted input vectors. In each layer of the model, the weighted input vectors are multiplied by the sigmoid activation function to yield the intermediate probabilistic results in the hidden layers based on Equation 4.

$$y_i = f(\sum_{i=1}^{n} w_i . x_i + b) \quad \ldots (4)$$

Here, y_i is the dependent variable to be predicted, w_i are the weight matrix and x_i are the feature vectors, and f is the sigmoid activation function based on Equation 5.

$$f(x) = \frac{1}{1+e^{-(w_i x_i)}} \quad \ldots (5)$$

The softmax activation function is applied in the output layer to predict the class of each sample to yield the final prediction result as shown in Equation 6.

$$P(y = j|x = \frac{e^{x^T w_j}}{\sum_{c=1}^{C} e^{x^T w_j}} \quad \ldots (6)$$

Here, c ∈ {1. . . C} ranges over all the classes and $x^T w_j$ stands for the inner product of each feature and corresponding weight.

As shown in Fig. 2, selected features in each node of MANET are fed to the input layer. Once the data passes to the hidden layer, the model understands the relationship between each feature and their respective class labels. The model uses the one-versus-rest approach to classify the data in the hidden layer using sigmoid activation function. The result of hidden layer is integrated into the output layer using the softmax activation function to give the prediction class levels.

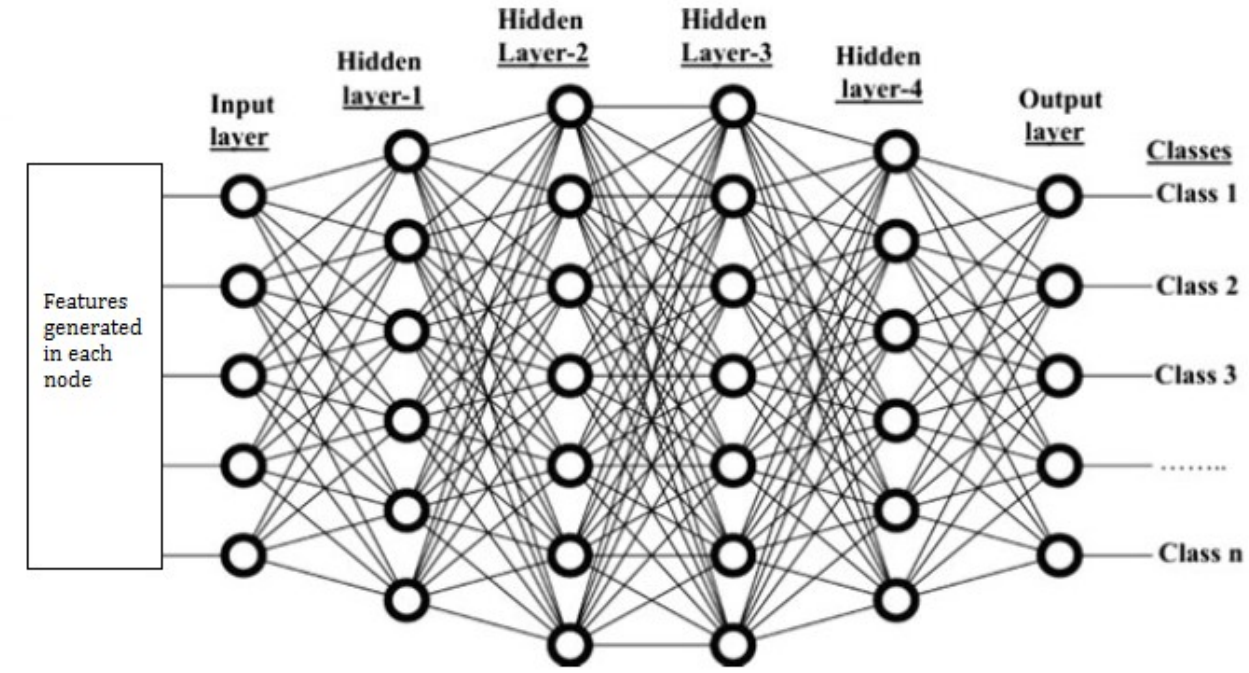

Fig. 2. Deep Network Architecture.

III. DATASETS

The evaluation datasets play a vital role in the validation of any IDS approach, by allowing us to assess the proposed method's capability in detecting intrusive behavior. The commonly used datasets for network packet analysis in commercial products are not easily available due to privacy issues. However, there are a few publicly available datasets such as DARPA, KDD, NSL-KDD and ADFA-LD and they are widely used as benchmarks. Existing datasets that are used for building and comparative evaluation of IDS are discussed in this section along with their features and limitations.

*DARPA / KDD Cup99:*The earliest effort to create an IDS dataset was made by DARPA (Defense Advanced Research Project Agency) in 1998 and they created the KDD98 (Knowledge Discovery and Data Mining (KDD)) dataset. In 1998, DARPA introduced a programme at the MIT Lincoln Labs to provide a comprehensive and realistic IDS benchmarking environment (MIT Lincoln Laboratory, 1999). Although this dataset was an important contribution to the research on IDS, its accuracy and capability to consider real-life conditions have been widely criticized (Creech & Hu, 2014b). These datasets were collected using multiple computers connected to the Internet to model a small US Air Force base of restricted personnel. Network packets and host log files were collected.

Lincoln Labs built an experimental test bed to obtain 2 months of TCP packets dump for a Local Area Network (LAN), modelling a usual US Air Force LAN. They modelled the LAN as if it were a true Air Force environment, but interlaced it with several simulated intrusions. The collected network packets were around four gigabytes containing about 4,900,000 records. The test data of 2 weeks had around 2 million connection records, each of which had 41 features and was categorized as normal or abnormal. The extracted data is a series of TCP sessions starting and ending at well-defined times, between which data flows to and from a source IP address to a target IP address, which contains a large variety of attacks simulated in a military network environment. The 1998 DARPA Dataset was used as the basis to derive the KDD Cup99 dataset which has been used in Third International Knowledge Discovery and Data Mining Tools Competition (KDD, 1999).

*DDoS Evaluation Dataset (CICDDoS2019):*Distributed Denial of Service (DDoS) attack is a menace to network security that aims at exhausting the target networks with malicious traffic. Although many statistical methods have been designed for DDoS attack detection, designing a real-time detector with low computational overhead is still one of the main concerns. On the other hand, the evaluation of new detection algorithms and techniques heavily relies on the existence of well-designed datasets. The CICDDoS2019 contains benign and the most up-to-date common DDoS attacks, which resembles the true real-world data (PCAPs). It also includes the results of the network traffic analysis using CICFlowMeter-V3 with labeled flows based on the time stamp, source, and destination IPs, source and destination ports, protocols and attacks.
In this dataset, the authors have provided different modern reflective DDoS attacks such as PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP. The capturing period for the training day on January 12th started at 10:30 and ended at 17:15, and for the testing day on March 11th started at 09:40 and ended at 17:35. Attacks were subsequently executed during this period. The authors executed 12 DDoS attacks that include NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN and TFTP on the training day and 7 attacks including PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag and SYN in the testing day.
The dataset has been organized per day. For each day, they recorded the raw data including the network traffic (Pcaps) and event logs (windows and Ubuntu event Logs) per machine. The features extraction process include the usage of CICFlowMeter-V3 and extracted more than 80 traffic features and saved them as a CSV file per machine. The details of this dataset can be found in (Iman et al. 2019).
NSL-KDD: NSL-KDD is a public dataset, which has been developed from the earlier KDD cup99 dataset [24]. A statistical analysis performed on the cup99 dataset raised important issues which heavily influence the intrusion detection accuracy, and results in a misleading evaluation of AIDS [24]. The main problem in the KDD data set is the huge amount of duplicate packets. Tavallaee et al. [24] analyzed KDD training and test sets and revealed that approximately 78% and 75% of the network packets are duplicated in both the training and testing dataset. This huge quantity of duplicate instances in the training set would influence machine-learning methods to be biased towards normal instances and thus prevent them from learning irregular instances which are typically more damaging to the computing system. Tavallaee et al. [24] built the NSL-KDD dataset in 2009 from the KDD Cup'99 dataset to resolve the matters stated above by eliminating duplicated records.

The NSL-KDD train dataset consists of 125,973 records and the test dataset contains 22,544 records. The size of the NSL-KDD dataset is sufficient to make it practical to use the whole NSL-KDD dataset without the necessity to sample randomly. This has produced consistent and comparable results from various research works. The NSL_KDD dataset comprises 22 training intrusion attacks and 41 attributes (i.e., features). In this dataset, 21 attributes refer to the connection itself and 19 attributes describe the nature of connections within the same host. The 41 features of the KDD Cup99 dataset are presented in Table 1.

Performance metrics for IDS: There are many classification metrics for IDS, some of which are known by multiple names. IDS are typically evaluated based on the following standard performance measures:
True Positive Rate (TPR): It is calculated as the ratio between the number of correctly predicted attacks and the total number of attacks. If all intrusions are detected then the TPR is 1 which is extremely rare for an IDS. TPR is also called a Detection Rate (DR) or the Sensitivity. The TPR can be expressed mathematically as

$$TPR = \frac{TP}{TP+FN} \quad \ldots (7)$$

False Positive Rate (FPR): It is calculated as the ratio between the number of normal instances incorrectly classified as an attack and the total number of normal instances.

$$FPR = \frac{FP}{FP+TN} \quad \ldots (8)$$

False Negative Rate (FNR): False negative means when a detector fails to identify an anomaly and classifies it as normal. The FNR can be expressed mathematically as:

$$FNR = \frac{FN}{FN+TP} \quad \ldots (9)$$

Classification rate (CR) or Accuracy: The CR measures how accurate the IDS is in detecting normal or anomalous traffic behavior. It is described as the percentage of all those correctly predicted instances to all instances:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \ldots (10)$$

Receiver Operating Characteristic (ROC) curve: ROC has FPR on the x-axis and TPR on the y-axis. In ROC curve the TPR is plotted as a function of the FPR for different cut-off points. Each point on the ROC curve represents a FPR and TPR pair corresponding to a certain decision threshold. As the threshold for classification is varied, a different point on the ROC is selected with different False Alarm Rate (FAR) and different TPR. A test with perfect discrimination (no overlap in the two distributions) has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity). The ROC Curve is shown in Fig. 3.

Table 1. The 41 features of NSL-KDD Dataset [14].

Label	Network data feature	Label	Network data feature	Label	Network data feature	Label	Network data feature
A	duration	L	Logged in	W	count	AH	dst_host_same_srv_rate
B	protocol-type	M	num_ comprised	X	srv_count	AI	dst_host_diff_srv_rate
C	service	N	root_shell	Y	serror_rate	AJ	dst_host_same_src_port_rate
D	flag	O	Stu attempted	Z	srv_serror_rate	AK	dst_host_srv_diff_host_rate
E	src_bytes	P	num_root	AA	rerror_rate	AL	dst_host_serror_rate
F	dst_bytes	Q	Num of file	AB	srv_rerror_rate	AM	dst_host_srv_serror_rate
G	land	R	Number of shell	AC	same_srv_rate	AN	dst_host_rerror_rate
H	wrong_fragment	S	num_access_files	AD	diff_srv_rate	AO	dst_host_srv_rerror_rate
I	urgent	T	num_outbound_cmds	AE	srv_diff_host_rate		
J	hot	U	Is host login	AF	dst_host_count		
K	num_falied_logins	V	Is guest login	AG	dst_host_srv_count		

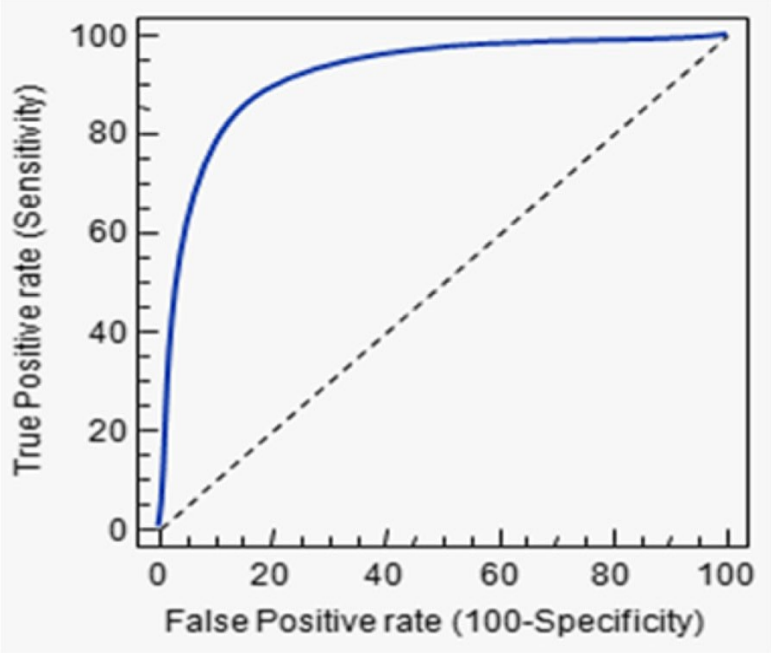

Fig. 3. A typical ROC curve.

4. Experimental Results and Discussion

In this section, detailed discussion of experimental results is covered. Extensiveexperiments are conducted on the standard multiclass dataset. The tenfold cross-validationmethod is employedto evaluate the performance of the proposed model. The average of the tenfold crossvalidationgives the classification accuracy along the standard deviation which measuresthe deviation of the predicted class from the mean of the classification accuracyon the tenfold evaluation results.

In order to evaluate our approach we simulated a mobile ad hoc network (MANET) and we conducted a series of experiments. For our experiments we have made the assumption that the network has no preexisting infrastructure and that the employed ad hoc routing protocol is the Ad hoc On Demand Distance Vector (AODV). We have implemented the simulator within the GloMoSim [4] library. Our simulation models a network of 50 hosts placed randomly within an 850 x 850 m2 area. Each node has a radio propagation range of 250 meters and the channel capacity was 2 Mbps. The nodes in the simulation move according to the 'random way point' model. At the start of the simulation, each node waits for a pause time, then randomly selects and moves towards a destination with a speed uniformly lying between zero and the maximum speed. On reaching this destination it pauses again and repeats the above procedure till the end of the simulation. The minimum and maximum speed is set to 0 and 20 m/s, respectively, and pause times at 0, 200, 400, 700 sec. The simulation time of the experiments was 700 sec, thus a pause time of 0 sec corresponds to the continuous motion of the node and a pause time of 700 sec corresponds to the time that the node is stationary.

Each node is simulated to generate Constant Bit Rate (CBR) network traffic. The size of the packets sent by each node varies from 128 to 1024 bytes. We have studied the performance of the classification algorithms for various sampling intervals (5, 10, 15, 20, 25, 30) in order to study how quickly these algorithms can perform intrusion detection. The sampling interval dictates both the interval for which the statistical features are calculated, and the period between each classification decision. We expect that longer intervals may provide more information, but with the cost of slower detection. We have also evaluated the performance of the classification algorithmsfor 5, 15 and 25 malicious nodes. In each case the number of all nodes in the network is set to 50.

In our experiments, we have simulated four different types of attacks:

Flooding attack: We have simulated a flooding attack [18] for multiple paths in the network layer, where each malicious node sends forged RREQ (Route REQuest) packets randomly to all nodes of the network every 100 msec.

Forging attack: We have simulated a forging attack [25] for RERR (RouteERRor) packets, where each malicious node modifies and broadcasts (to a selected victim) a RERR packet every 100 msec leading to repeated link failures.

Packet Dropping attack: We have simulated a selective packet dropping [7]attack, where each malicious node drops all RERR packets leading legitimate nodes to forward packets in broken links.

Black Hole attack: In a black hole attack [22], a malicious node advertises spurious routing information, thus receiving packets without forwarding them but dropping them. In the black hole attack we have simulated the scenario where each time a malicious-black hole node receives a RREQ packet it sends a RREP (RouteREPly) packet to the destination without checking if the node has a path towards the selected destination. Thus, the black hole node is always the first node that responds to a RREQ packet and it drops the received RREQ packets. Furthermore, the malicious-black hole node drops all RREP and data packets it receives if the packets are destined for other nodes.

A very important decision to be made is the selection of feature vectors that will be used in the classification. The selected features should be able to represent the network activity and increase the contrast between "normal" and "abnormal" network activity. Here, we have considered L_1-regularized feature selection approach which select the best features from the given set of features.

For each sampling interval time (5, 10, 15, 20, 25, 30) we have created one training dataset, where each training instance contains summary statistics of network activity for the specified interval using all the above features and in addition, the type of attack performed during this interval. This enables us to use supervised learning techniques for classification. Each training dataset was created by running different simulations with duration 700 sec for different network mobility (pause time equal to 0, 200, 400, 700 sec) and varying numbers of malicious nodes. The derived datasets from each of these simulations were merged and one training dataset was produced for each sampling interval. A similar procedure was followed in order to produce the testing datasets.

Figure 4 depicts the average Detection Rate (DR) for all classification algorithms on multiclass classification. The best Detection Rate (DR) is achieved for the proposed classifier for multiclass classification and is equal to 99%. The second best classifier with a high Detection Rate (DR) equal to 98% is achieved with the SVM classifier for multiclass classification. The classifier that presents the poorest performance is the Naïve Bayes classifier with Detection Rate (DR) equal to 90%. The above analysis is with respect to flooding attacks. The same kind of detection rate can be observed with respect to different types of attacks.

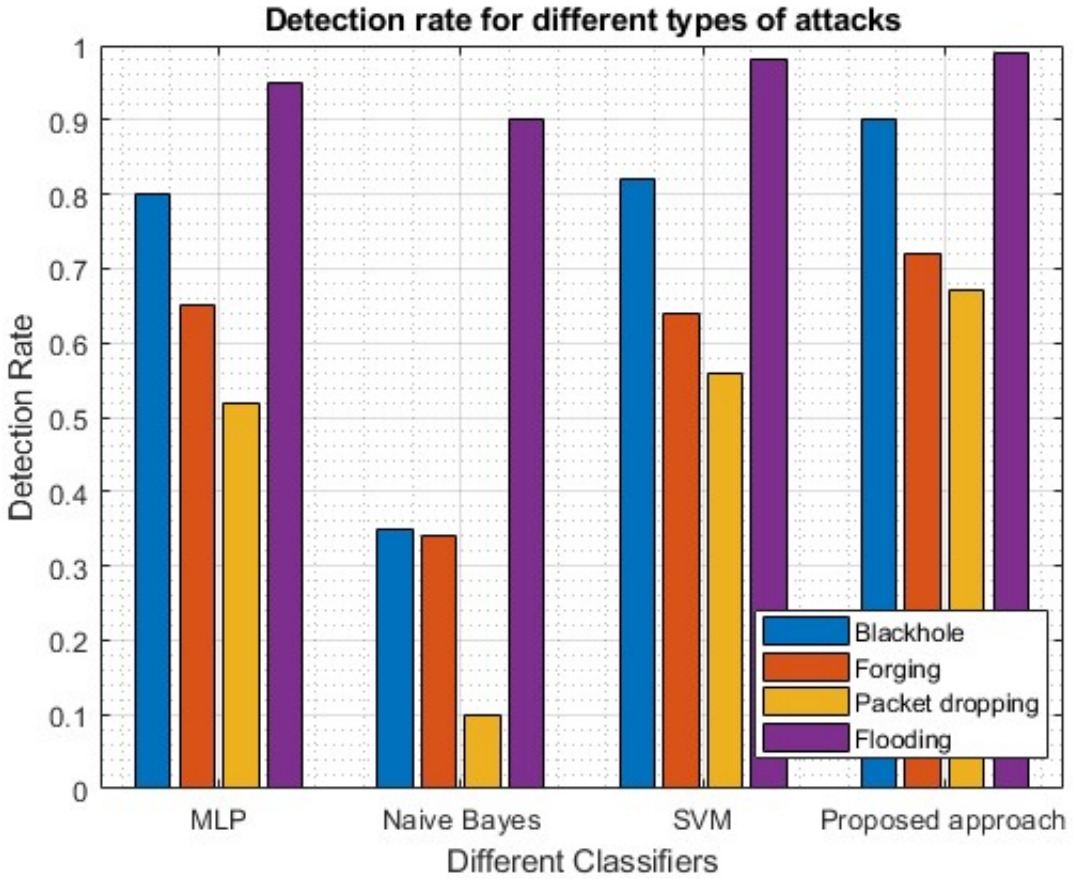

Fig. 4. Detection Rate due to different classifiers on different types of attacks.

In Figure 5, we have presented the classification error considering the varying number of malicious nodes. It shall be observed from the figure that the proposed approach classification error rate is quite less when compared to other classifiers. In addition, one can also observe that the classification error decreases with an increase in the number of malicious nodes. Higher the number of attacks, better is the accuracy and probably, it depends upon the training set also.

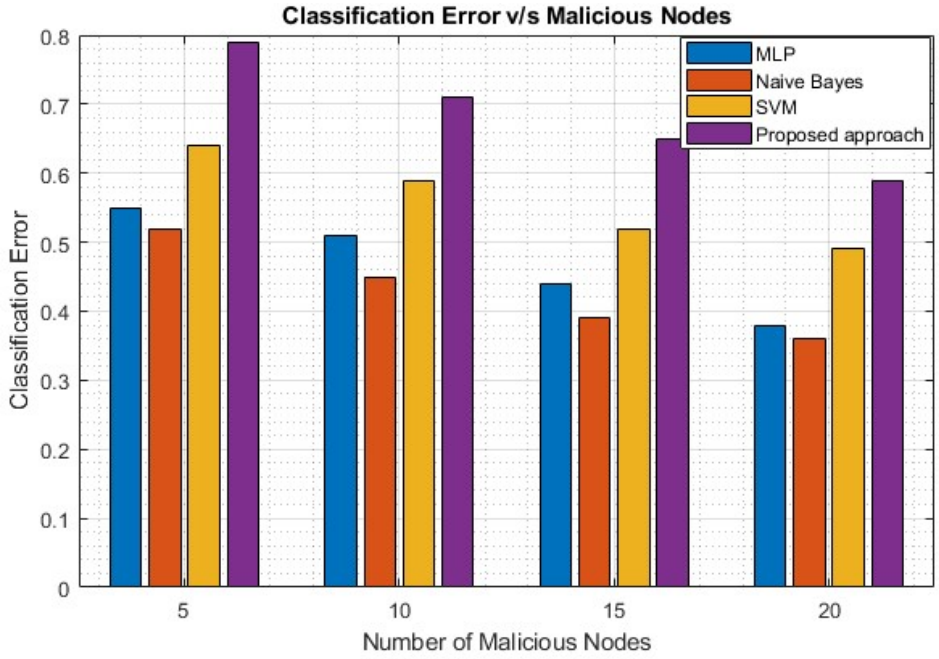

Fig. 5. Classification error due to different classifiers with varying number of malicious nodes.

We have also conducted experimentation with different classifiers by considering NSL-KDD dataset which is one of the extensively used intrusion detection data set in the literature. Table 1 shows Precision, Recall, F1-score due to different supervised machine learning classifiers in identifying the intrusion. Based on the results shown in the Table 1, we can conclude that the SVM based classifier and the proposed approach produces comparable results where as Naïve Bayes classifier performance is relatively poor when compared to other approaches.

Table 2. Precision, Recall and F1-score due to different classifiers with respect to NSL-KDD dataset.

Methods	Precision	Recall	F1-score
MLP	0.76	0.79	0.773
Naïve-Bayes	0.73	0.67	0.698
SVM	0.83	0.85	0.84
Proposed approach	0.87	0.84	0.854

V. CONCLUSION

The network efficiency and the lifetime of network depends upon the competence of the robust intrusion detection system to handle the attacks. In this work, we have developed an accurate IDS based on L_1-regulated feature selection, and deep learning approach is explored for classification. The L_1-regulated feature selection is based on Linear Support Vector Machine that is characterized by adding a penalty term to the prediction error in order to reduce the weight of the irrelevant features and to make the relevant features having nonzero weights. For classification purpose, deep learning neural network is initialized with sigmoid activation function in the input and hidden layers. The suitability of the proposed approach is demonstrated experimentally by considering the standard datasets and comparative analysis is presented with state-of-the-art approaches and the results are shown exhibiting classification accuracy, precision, recall, f1-score and detection rate metrics. Comparative study of our method, using the standard metrics, with someselected works shows that our method achieves better results.

REFERENCES

[1] Aziz, R., Verma, C.K., Srivastava, N. (2016). A fuzzy based feature selection from independent component subspace formachine learning classification ofmicroarray data. GenomicsData 8, 4–15.

[2] Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials 18(2):1153–1176.

[3] Bühlmann, P., Van De Geer, S. (2011). Statistics for High-dimensional Data: Methods, Theory and Applications. Springer Science & Business Media.

[4] Butun I, Morgera SD, Sankar R (2014) A survey of intrusion detection systems in wireless sensor networks. IEEE Communications Surveys & Tutorials 16(1):266–282.

[5] Can, O and O. K. Sahingoz, (2015). A survey of intrusion detection systems in wireless sensor networks, in 2015 6th international conference on modeling, simulation, and applied optimization (ICMSAO), pp. 1–6.

[6] Chen, K.-H., Wang, K.-J., Wang, K.-M., Angelia, M.-A. (2014): Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data. Appl. Soft Comput. 24, 773–780.

[7] Djenouri D., Mahmoudi O., Bouamama M., Llewellyn-Jones D., Merabti M. (2007): On Securing MANET Routing Protocol against Control Packet Dropping. In: Proceedings of IEEE International Conference on Pervasive Services (ICPS' 07), pp. 100-108, Istanbul, Turkey.

[8] Ebrahimpour, M.K., Eftekhari, M. (2017): Ensemble of feature selection methods: a hesitant fuzzy sets approach. Appl. Soft Comput. 50, 300–312.

[9] Elhag, S., A. Fernández, A. Bawakid, S. Alshomrani, and F. Herrera, (2015). On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems, Expert System Applications.vol. 42, no. 1, pp. 193–202.

[10] Fonti, V., Belitser, E. (2017): Feature selection using LASSO, VU Amsterdam Research Paper in Business Analytics.

[11] Garro, B.A., Rodríguez, K., Vázquez, R.A. (2016): Classification of DNA microarrays using artificial neural networks and ABC algorithm. Appl. Soft Computing 38, 548–560.

[12] GloMoSim: Global Mobile Information Systems Simulation Library. http://pcl.cs.ucla.edu/projects/glomosim/.

[13] ImanSharafaldin, ArashHabibiLashkari, SaqibHakak, and Ali A. Ghorbani, 2019. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy, IEEE 53rd International Carnahan Conference on Security Technology, Chennai, India.

[14] Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2, 20 (2019). https://doi.org/10.1186/s42400-019-0038-7.

[15] Meshram A, and Haas C (2017) Anomaly detection in industrial networks using machine learning: a roadmap. In: Beyerer J, Niggemann O, Kühnert C (eds) Machine learning for cyber physical systems: selected papers from the international conference ML4CPS 2016. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 65–72.

[16] Moayedikia, A., Ong, K.-L., Boo, Y.L., Yeoh, W.G.S., Jensen, R. (2017): Feature selection for high dimensional imbalanced class data using harmony search. Eng. Appl. Artificial Intelligence. 57, 38–49 (2017)

[17] Nguyen, T., Khosravi, A., Creighton, D., Nahavandi, S. (2017): A novel aggregate gene selection method for microarray data classification. Pattern Recognition. Letters. 60, 16–23 (2015).

[18] Ning, P. and Sun, K.: How to Misuse AODV, 2003. A Case Study of Insider Attacks against Mobile Ad-hoc Routing Protocols. In: Proceedings of the 2003 IEEE Workshop on Information Assurance, pp.60-67, NY.

[19] Ravi D.,Wong, C., Deligianni, F., Berthelot, M., Andreu- Perez, J., Lo, B., Yang, G.-Z. (2017): Deep learning for health informatics. IEEE J. Biomed. Health Inform. 21(1), 4–21.

[20] Sasikala, S., Appavu alias Balamurugan, S., Geetha, S. (2017): A novel adaptive feature selector for supervised classification. Inf. Process. Lett. 117, 25–34.

[21] Sharbaf, F.V., Mosafer, S., Moattar, M.H. (2016): A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization. Genomics 107(6), 231–238.

[22] Shurman M.AI, Yoo S.M., Park S. (2004): Black Hole Attack in Wireless Ad Hoc Networks. In: Proceedings of ACM 42nd Southeast Conference (ACMSE 04),pp. 96-97, Alabama.

[23] Symantec, "Internet security threat report 2017," April, 2017, vol. 22 Available: https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf

[24] Tavallaee, M., E. Bagheri, W. Lu, and A. A. Ghorbani, (2009). A detailed analysis of the KDD CUP 99 data set, in 2009 IEEE symposium on computational intelligence for security and defense applications, pp. 1–6.

[25] Yi P., Hou Y.F., Zhong Y., Zhang S., Dai Z. (2006): Flooding Attack and Defence in Ad hoc Networks. In: Systems Engineering and Electronics, Vol. 17, No. 2, pp. 410-416.

[26] You, W., Yang, Z., Ji, G. (2014): Feature selection for high-dimensional multi-category data using PLS-based local recursive feature elimination. Expert Syst. Appl. 41(4), 1463–1475.

[27] Zhu, Z., Ong, Y.-S., Dash, M. (2007): Markov blanket-embedded genetic algorithm for gene selection. Pattern Recognition. 40(11), 3236–3248.