SindhUniv. Res. Jour. (Sci. Ser.) Vol. 51 (03) 483-492 (2019)

http://doi.org/10.26692/sujo/2019.03.77

SINDH UNIVERSITYRESEARCH JOURNAL (SCIENCESERIES)

An Efficient Technique for Network Intrusion Detection Using Feature selection

R. A. SHAH, A. A. WAGAN*, M. ALI**, K. HUSSAIN***, R. BIBI****

Department of Computer Systems Engineering, The Islamia University, Bahawalpur, Punjab

Received 03rd February 2019 and Revised 20th August 2019

Abstract: Network Intrusion Detection System (NIDS) is one of the most significant parts of network security that can make secure transactions over a network. Despite many efforts in the field, we can observe increased sophistication and variety of attacks on networks. In such situation Machine learning (ML) based methods have emerged some of the most effective as well as popular methods to detect the attacks. One of the complexities involved in the ML-based method is that they are mostly of the black-box nature, so their inner working phenomena are very often quite complex to understand and interpret. Moreover, high-dimensional features and an inadequate number of training records have caused some problems in the classifications, such as over fitting of the results, noise sensitiveness, overload computation and lack of significant physical interoperability. In this paper, we propose a discriminative features selection and network intrusion classification by applying sparse modeling with Lasso and SVMs with two kernel functions. SVMs are standard ML techniques which can provide reasonable performance however it can have some shortcomings such as interpretability and huge computational cost. On the other hand, sparse modeling has been considered as an advanced technique for data analysis and processing via regularization. Sparse modeling can be used to simultaneously select discriminative features from the repository of the dataset. Moreover, it also determines the coefficient of the linear classifier where prior information about features structure can be mapped into various sparsity-inducing regularization such as Lasso. Furthermore, we apply sparse modeling for the multiclass-classification purpose; in this way, we can identify and select the features yielded by the network attacks that are the most significant ones. Our experimental in this correspondence suggest that the proposed techniques have better performance than most of the state-of-the-art methods.

Keywords: Network Intrusion Detection, Feature Selection, Network Security, Classification.

1. INTRODUCTION

Since the last two decades, the subject of network intrusions detection has got significant attention by researchers due to its sensitivity as well as the importance. An Intrusion Detection System (IDS) is a necessary part of a complete defense-in-depth architecture for network security. Network security can be described as the process of securing all three factors i.e., Confidentiality, Integrity, and Availability (CIA) (Hatada and Mori, 2017). (Al-mamory and Jassim, 2015) As a result, network security has been getting more and more attention and importance due to the tremendous escalation of network activities and network applications. An Internet Organized Crime Threat Assessment (IOCTA) report has been published (in 2016 and mid-2017), as the fourth annual presentation of the cybercrime threat landscape by Europol's European Cybercrime Center (EC3). It is explained that how cybercrime grows and develops, obtains new trends and ways starting with few of the intrusions and leading to the unprecedented scale (European Cybercrime Centre (EC3 (EC3). (2017). Moreover, in (2005), a joint report was published by computer security Institute and FBI; in the report, it was highlighted that the financial loss incurred by respondent organizations because of computer intrusions/attacks were around the US $130 million Institute and Investigation, (2005).

Due to these and many other reasons, IDS has become one of major research areas in computer network security although the concept dates back to 1980s when it was proposed by Anderson, 1980).

Intrusion Detection System (IDS) can be expressed as a significant component of a comprehensive security mechanism that automatically monitors and investigates various network activities and later classifies the system events as normal or abnormal/attack events. More precisely, the overall goal of an IDS is to identify intrusions through internet traffic with better accuracy, in order to provide secure and safe transactions over the network systems (Shah, *et al.*, 2016) Generally, there are two basic approaches exist for the IDS:

E-mail: ali.rehan@iub.edu.pk (corresponding author), asif.wagan@smiu.edu.pk, munwar.ali@sbbusba.edu.pk
kashif.hussain@iub.edu.pk, syedaridashah@gmail.com

*Department of Computer Science, SMIU Sindh, Pakistan

**Department of Information Technology Shaheed B Bhutto University. Sindh, Pakistan

***Department of Electronics, Mehran UET, Jamshoro, Pakistan

1- Misuse detection method: It is also known as the knowledge-base or signature-based detection method. It is based on resembling record data to identify the intrusion patterns. In other words, signature base detection has quite low false alarm percentage however it cannot find the novel attacks. Therefore, misuse detection method needs continuous updates in order to detect the new intrusions (Shah, *et al.,* 2016).

2- Anomaly detection method (ADM): It is also known as the behavioral-based detection system. It can be used to identify attacks when there is a change of user behavior in the network. It works on the assumption that intrusions can be detected by analyzing deviations from a normal activity of the monitored entities. However, several data samples need to be analyzed by the process on a similar network. Therefore various data samples, as well as features, need to be selected from analyzed data and then selected features would be considered as the most important characteristics of the data. As a result, an input feature is sufficiently enough to have knowledge about the normal and abnormal entity. Hence, the IDS could be used as a better mechanism to separate various types of users for improving the security of a network system. Moreover, the ADM method has the significant advantage of catching novel attacks over the signature-based technique (Shah, *et al.*, 2016).

Nowadays, one of the most significant problem for contemporary IDS research is feature selection (FS) from a tremendous volume of network data. More recently several researchers have changed their direction towards using feature selection techniques for the classification (Qian, and Zhou, 2013). The trend can be defined by the fact that feature selection used for the distinct purpose such as enhance an efficiency of the learning algorithm, reducing computational complexity, achieving high accuracy rates and getting a clear understanding for classification problem. Previously the goal of features selection process was limited to the selection of a subset of original features without significant modification. FS is one of the most significant data pre-processing stage in which different fields involoved such as pattern recognition, ML, and DM (Saeys, *et al.*, 2007). Feature selection techniques can be classified into three main types: wrapper technique, filter technique and embedded technique (Kohavi, and John, 1997) furthermore, filter technique has the advantage of being simple and easy to implement. Though, the method (filter) has several limitations such as the lack of interaction with various classifier models.

The wrapper method has been proposed in (Saeys, *et al.,* 2007) Wrapper schemes correspond to a black box method means, it selects a potential set of input features because it has predictive power. Moreover, wrapper approach is comprehensive and simple to understand. However, the stability of features selection can be described as "the insensitivity of feature selection approach to the variation of the training set". Wrapper method has some shortcomings such as it is computationally very expensive than filter as well as sparse method (Lasso). Wrapper method could not scale well into huge datasets and usually shows overfitting on high dimensional datasets (Cateni and Colla, 2016). An embedded methods can be employed FS scheme for learning classifier. In embedded schemes, the features need to be selected at the training phase in order to reduce the computational cost and increase the performance of the learning algorithm. It is necessary to note that one of the main distinction between embedded and wrapper methods is that the embedded technique need repeated updates, as well as evaluation of the process parameters which are based on the efficiency of the model under consideration. Though, in an embedded method such as in (Cateni, *et al.*, 2017). FS process is normally integrated at the training stage. These techniques' main purpose is to find the features that could be used for classification purpose. However, our technique (Lasso) main goal is to examine the essential discriminant features that are beneficial for intrusion classification while decreasing the noisy/irrelevant features that undermine the performance of classification.

It is essential to consider that features selection and attacks evaluation needs to be simpler for network administrator as they require to understand the selected features' role in various attack categories. Furthermore, our main goal is to examine security attacks by analyzing the contribution of (NSL-KDD) 41 input features and select most significant ones that are potentially significant to detect anomalies in a network system with respect to attack categories. In this paper, we examine the problem of discriminant/hidden feature selection for the IDS in order to detect different attacks on the network system. In this regard, we analyze the discriminative features in identifying well-known attacks by employing sparse modeling with (L_1- regularization) and SVMs with Radial Basis Function (rbf) and linear (lin) kernel function respectively. The sparse modeling is a currently developed method employed for features selection and classification. The Sparse modeling has been extensively applied for wide range of applications such as Image processing, pattern recognition and signal processing (Qian, and Zhou, 2013) (Hagos, *et al.*, 2017). In this research work, our major contributions as follows,

1. In this paper, we use sparse modeling (Lasso) and Support Vector Machine (SVM) with radial basis function (rbf) and linear kernel function to classify network attacks.

2. Regularization (ℓ_1) via Sparse Modeling: feature selection has been mapped into penalty term of sparsity in optimization function. The regularization has been performed for the goal of presenting more meaningful and interpretable feature selection for attack examination.

3. We employ Sparse Modeling with Lasso for Multi-class classification in order to present significant features of various classes of attack.

4. We also show a deep and strong intuition from a security engineering viewpoint that why the feature selection performed via Sparse Modeling method are so important in detecting several attacks in a network.

$$\boldsymbol{Y} = \boldsymbol{Xw} + \boldsymbol{\alpha} \tag{1}$$

5. Finally, we have conducted experimentation and through empirical results have been compared a sparse model with SVMs and other models. We have found that sparse models show better results at a lower computational cost as well as better detection rates.

2. SPARSE MODELING (LASSO)

The purpose of sparse modeling with (ℓ_1) regularization is to perform feature selection (discriminant/hidden feature selection), in order to obtain better performance of the system (IDS). We may consider a one stage approach that uses a sparse model with one versus all scheme. Whereas, the alternative methods such as SVMs are two stages and it needs more training and testing time also computationally inefficient. The linear sparse logistic regression as least Absolute Shrinkage and Selection Operator (Lasso) uses a shrinkage technique. It means a feature partly incorporated in the regression scheme in order to perform feature selection.

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\boldsymbol{arg\,max}} |\boldsymbol{Y} - \boldsymbol{Xw}|^2 + \lambda |\boldsymbol{w}|_1 \tag{2}$$

$$|\boldsymbol{w}|_{\mathbf{1}} = \sum_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{R}} |\boldsymbol{w}|_{\mathbf{1}} \tag{3}$$

Though, the coefficient is shrunken towards zero as λ increase. Hence, for the regression or classification problem, sparse model select only discriminant features for accurate prediction for the system (IDS). In the case of network intrusion detection, term sparsity suggests that simply part of features helpful for specifying the network intrusion.

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\boldsymbol{arg\,max}} \left\{ -\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \boldsymbol{ln(}1 + \boldsymbol{exp(}-\boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{x_i y_i}\boldsymbol{)} \right\} \tag{5}$$

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\boldsymbol{arg\,max}}\, \boldsymbol{l(w)} + \boldsymbol{\lambda g(w)} \tag{6}$$

Consider a prediction problem having N samples and $y_1, y_1, y_1, ..., y_n$ are results, whereas features x_{ij} and $i = 1,2,...,N,\ j = 1,2,...,R$ where R are the input features, let X denote the $N \times R$ input matrix and Y describe the $R \times 1$ output matrix. The generalized regression model is presented in Eq. (1).

Here, w is a vector of coefficients corresponding to the input features and α is the noise vector with zero mean and a permanent variance. To determine w, an early optimization technique which is known as least square can be employed. However, prediction performance may not be quite sufficient in various situations. Therefore, a constraint on w need to be applied in an extensive form. Nowadays an effective constraint is being employed is sparsity. One of the most prominent sparse regression models is Least Absolute Shrinkage And Selection Operator (Lasso) introduced by Tibshirani, (1996). It has a regularized least square scheme that is utilizing a (ℓ_1) penalty on the regression coefficients. It should be described as,

The Lasso believe that the input features are almost free/independent, that means not very correlated, it represents the precise construction of input features. However, it could be argued that a plausible solution may be obtained in practice.

Here the conditions, such as $Y = \{+1, -1\}$ represent class labels (+1) corresponding to class normal and (−1) corresponding to intrusions/attack. Furthermore, Logistic regression (LR) is a probability conditional model and could be represented as,

The maximum likelihood estimation of the parameter w is obtained by,

Moreover, joining the sparse constraint, then the sparse model (Lasso) could be described as (Meier, *et al.*, 2008), furthermore, $g(w) = \|w\|_1$ is the (ℓ_1) norm regularization and λ is a regularization parameter. If, we correspond to the logistic regression $l(w)$ means solve directly, it is ill-posed and may get overfitting in the classification results. In order to avoid above problem, a popular approach to reducing overfitting is the sparse regularization/constraint. The solution to the (ℓ_1) norm regularized logistic regression could be represented in a Bayesian structure as the "maximum a posteriori probability" estimation of $l(w)$. Moreover, **(Fig. 1),** shows the illustration of discriminant feature selection and classification by our proposed technique Lasso.

Fig. 1. Feature Selection by Lasso**,** the x_i are the features set and w is a sparse coefficient vector, a white elements in w the stand for zero elements (sparse data) and rest of all are selected discriminant features.

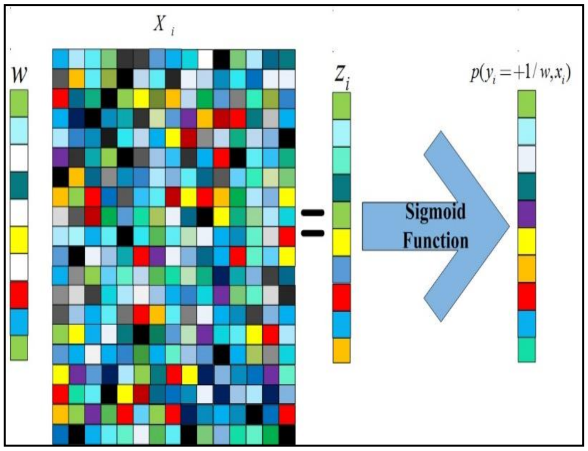


Fig. 1, shows the illustration of discriminant

2.1. Optimization of Algorithm

The Lasso has been represented as a convex function. It means that coefficient can be determined

$$\boldsymbol{w^{(j+1)} = prox_{\lambda}(u^{(j+1)})} \qquad (10)$$

$$= \boldsymbol{sgn(u^{(j+1)})\,max(\left|u^{(j+1)}\right| - \lambda, 0)}$$

by a constrained convex optimization problem. In current research on the subject (Li and Qian, 2009),

(Friedman, *et al.*, 2010) penalization convex optimization problem has been examined quite efficiently and provides various solutions such as given in (Li and Qian, 2009). The sparse model based on convex function and it has suggested that

$$\boldsymbol{p(y_i = +1 / w, x_i) = \frac{1}{1 + exp^{(-w^T x_i)}}} \qquad (4)$$

coefficient corresponds to an extension of the gradient-based method, whereas, a cost function is to be reduced to the level of the non-smooth element. In this work, we apply accelerated proximal gradient approach, it has an advantage of low computational cost and leads to linear convergence.

Assume an objective function (6) is a convex function with a loss function $l(w)$ and a regularization function $\lambda g(w)$, the accelerated proximal gradient technique solves this task repeatedly where each of the repetition indexed via $j+1$, and comprised of two key steps.

The initial step is a descent step for the function $l(w)$, now, to accelerate the convergence, it starts the initial step from search point by Eq. (7), and then the

$$\boldsymbol{S^j = w^j + \alpha^j (w^j - w^{(j+1)})} \qquad (7)$$

$$\boldsymbol{u^{(j+1)} = s^{(j)} + t^{(j)} \nabla f(s^{(j)})} \qquad (8)$$

adaptive backtracking line search strategy is employed to define the step size. This comprises starting with a comparatively large estimation of step size with respect to the search direction and frequently shrinkage in the step size (backtracking) till the reduction of the objective function is obtained. Which is an affine organization of $w^{(j+1)}$ and $w^{(j)}$.

In Eq. (7), α^j is a tuning parameter. The approximate solution $w^{(j+1)}$ can be estimated as a gradient step. Promptly, an adaptive backtracking line search (Liu, *et al.*, 2009) is employed to decide a particular step size $t^{(j)}$. In Eq. (9), the second stage is to project u^j into regularized space, where a proximal operator is used. The proximal operator defined as.

For Lasso regularization, a practical solution for every variable w could be obtained as given in Eq. (10).

Frequently employing the accelerated gradient descent method and proximal operator, the technique reaches an optimal solution. However, a method is useful through the use of accelerated gradient descent and proximal operator. A comprehensive steps of the Lasso algorithm is listed in the **(Fig. 2).**

Table 1. Feature list of NSL-KDD

S. No	Attributes	S. No	Attributes
1	Duration	22	is_guest_login
2	protocol_type	23	count
3	Service	24	srv_count
4	Flag	25	serror_rate
5	src_byte	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_error_rate
8	wrong_fragmnet	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	Hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_svr_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmd	41	dst_host_srv_rerror_rate
21	is_hot_login		

3. EXPERIMENTS

In section 2, methodology of SPLR thoroughly explained. In this section, efficiency for features selection and network intrusion analysis, as well as, intrusions classification is illustrated.

```
Input: Sparse function $f(\bullet)$ and sparse regularization function $g(\bullet)$ with regularization parameter $\lambda$ .
Initialize: Step size $t^{(0)}$ and affine combination parameter $w^{(0)}$
Output: Optimum Result $w^{(*)}$
• $w \leftarrow 0; j \leftarrow 0;$
• $j \leftarrow j+1;$
• Estimate the search point $S^{j}$ via Eq. 7.
• Estimate the gradient descent $u^{(j+1)}$ through Eq. 8 by adaptive step size.
• Employ the proximal operator to estimate $w^{(j+1)}$ via Eq. 9.
• Update $t^{(j+1)}$ and $w^{(j+1)}$ for next repetition.
• Repeat the above steps till the difference between $w^{(j+1)}$ and $w^{(j)}$ is smaller than a threshold.
• Return $w^{(*)} = w^{(j+1)}$
```

Fig.2 Algorithm 1for the Lasso

3.1. NSL-KDD'99 Dataset

The NSL-KDD T. N.K. D. S. [Online]. "http://iscx.ca/NSL-KDD/." benchmarking dataset experiments is utilized. It is an improved version of the KDD'99, and is recommended to understand some significant challenges, as presented in (Tavallaee, *et al.*, 2009). The KDD'99 dataset generated by processing the TCPdump portions of the 1998 DARPA evaluation dataset, which was collected from a military network at MIT´s Lincoln Labs, in order to analyze the network intrusion detection. The NSL-KDD public dataset contains the attacks in four main categories: denial-of-service (DoS), remote-to-local (R2L), user-to-root (U2R) and Probe. Furthermore, the NSL-KDD dataset includes a total of 41 features for the investigation and one target predictor that shows the attack category's name. The set of features identifying specific connection are shown in **(Table 1).**

4. RESULTS AND DISCUSSION

In this work, the primitive goal of experiments to examine the potential of a sparse model (Lasso) for feature selection, which can be effective for the classification and analysis of the intrusions detection

$$prox_{(\lambda)} g(u) = arg\, min_{(u)} \left(\frac{1}{2} \left| u - w \right|_2^2 + \lambda g(u) \right) \qquad (9)$$

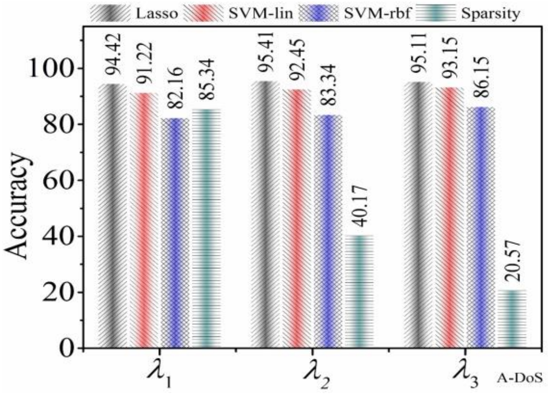
on a network.

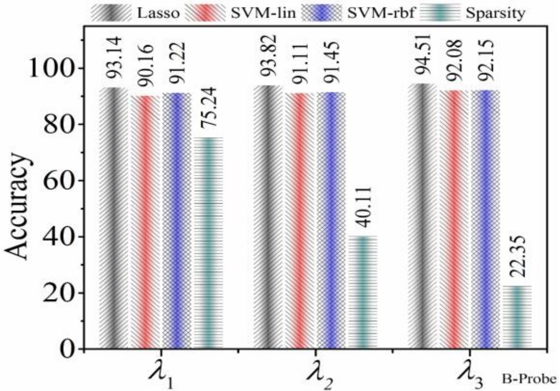
The Lasso selects the discriminative individual features, we have designed experiments to examine the performance of feature selection for network intrusion detection. It should be noted that a one Versus all procedure is employed to deal with the multiclass problem in our proposed technique. **(Table 2),** is the representation of datasets employed for the experiments.

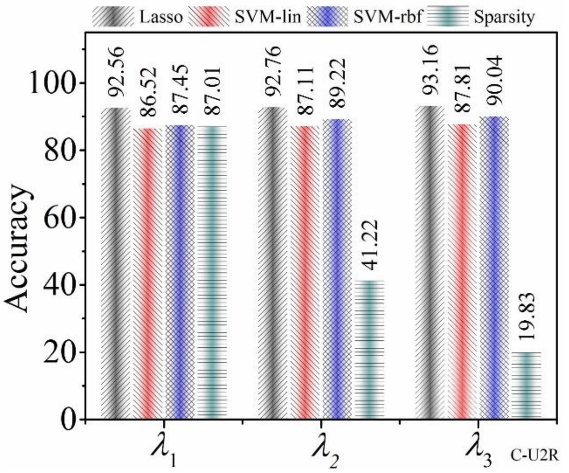
It is important that, λ present the degree of the sparsity. The higher the parameters are, the smaller the number of non-zero coefficient (important feature selection) will be, and the degree of sparsity obtained by Eq. (11). To evaluate the network

$$Sparsity = \frac{Number\ of\ discarded\ features}{Total\ number\ of\ input\ features} \qquad (10)$$

intrusion detection, the average classification accuracy (ACA) method is employed determine a particular class-specific accuracy.







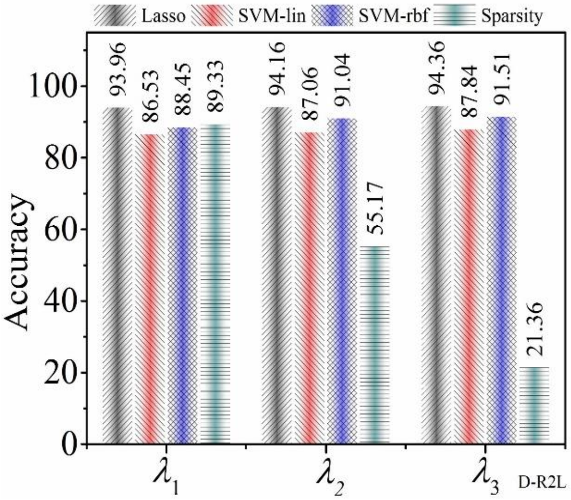


Fig. 3 (A-D). ACR of network intrusions and Degree of Sparsity on NSL-KDD datasets by Lasso and SVMs with selected features.

To illustrate the potential of the feature selection using the proposed methods, Fig. 3(A-D), displays the results of classification accuracies versus degree of sparsity on NSL-KDD. In the figures, the sparsity (discarded features) of model is displayed in the dark cyan color that is controlled by λ. To evaluate the validity and significance power of selected features, the selected features are employed as the input to SVM (rbf) and SVM (lin). The SVM (rbf) and SVM (lin) with the selected features achieve close or better classification results than employing all of the features. This investigation proves that features selected by sparse model are very important. Because, they reduce the feature dimensions, and help in preserving discriminant ability of the features.

Table 2. NSL-KDD - Training and Testing sets

Class	**Training Samples**	**Testing Sample**
Normal	9,727	6,059
Denial of Service (DoS)	39,145	22,985
Probe	411	417
Remote-to-Local (R2L)	113	24
User-to Root-(U2R)	06	1,619
Total Samples	49,902	31,104

Table 3. Feature Selection and ACA of Sparse Method and SVMs

Dataset	Feature Selection	LASSO	SVM-rbf	SVM-lin
NSL-KDD (Dos)	1,2,3,4,5,7,8,18,19,20,21,22,24,25,26,27,28, 29,30,32,33,34	95.51	83.43	92.23
NSL-KDD (Probe)	1,2,3,4,5,6,7,8,9,18,11,12,14,17-41	94.21	92.23	92.15
NSL-KDD (U2R)	1,2,6,7,8,9,11,12,13,14,17,18,20,21,22,23,25,26,26,27,28,33,34,36,39	93.76	89.34	87.15
NSL-KDD (R2L)	2,3,4,5,6,7,8,10,11,12,13,14,17-38,40,41	94.36	91.52	87.56

Table 4. Performance Comparison of Lasso and SVMs with other classifiers model.

Algorithm	Size Of Training Dataset	Size Of Testing Dataset	DR (%)	Train Time (Sec)	Average Training Time Per Sample (Sec)
VFDT [2]	1074985	67688	93.83	39.88	0.000003
RNN [21]	94409	31104	94.11	--	--
MLP [22]	49596	15437	92.03	350.15	0.007
C4.5 [23]	49596	15437	92.06	15.85	0.0003
SVM-rbf	49902	31104	92.78	16.74	0.0003
SVM-Lin	49902	31104	90.34	35.35	0.00007
Lasso	49902	31104	96.56	0.7051	0.000005

(Table 3), represents the classification accuracies of Sparse Logistics Regression (Lasso) and SVMs methods. The experiments in Table 3, are the average of the results from 10 times of random training sample selection. To find the DoS attacks, basic and time-based features are required. Features 7, 8, 29, (Land, Wrong – fragment, and same – srv- rate) are the most supporting features to find the DoS attacks. Which are effectively identified by the sparse (Lasso) method. Besides, these observations, the TCP fragmentation (Tear_Drop) belongs to the DoS attack. It prevents reassemble the protocols from putting together a fragmented UDP traffic packet send across the network to intended the destination by rebooting the target host. Furthermore, DoS attack also carries out several activities such as sending a massive data traffic to the corresponding service to block the connection channel. For instance, count, src_byte, flag, and Syn, are most contributing features for various attacks that belong to the DOS attack. Moreover, an intruder some time used the spoofed source IP address by sending many TCP connections with a flag to a port of the targeted host in the time window of T sec. However, the flag is a significant feature in recognizing the attack because it presents the summary information of connection behavior corresponding to the protocol specification on the network. The above-mentioned features have been effectively identified by Lasso.

Furthermore, for the Probe attack, time-based, and host-based traffic features are required. Which are automatically chosen through classifier training, listed in Table 3. Probe attack does not show an intrusions pattern with a time window of two seconds. In order to identify Probe attack, 'same-host' and 'same-service' types of features required. Moreover, it is based on the connection window of 100 connections rather than a time window of 2 seconds. Various kind of probing attacks have been used such as scan the Host (ports), using an increase time interval than two seconds and it occurred in every 1 minutes. Other associate features for probe attacks are remote job entry "service_rej" etc. R2L and U2R features have been shown in Table 3 respectively, and effectively detected by the Lasso. R2L and U2R attacks do not show any frequent pattern, because these attacks usually attached in the data portion of the packets and mostly involved in a single connection. Contributing features of R2L and U2R belongs the content features. In order to detect R2L and U2R attacks, need some data content features such as 'number – of – failed – logins 'and 'logged – in'. While in Table 3, these features effectively detected by Lasso. Moreover, some of the data content features are 'service (3)', "is- guest- login', 'Su_attempted' etc. A U2R attack usually detects, when an attacker/Intruder login as an administrator and creating multiple files and making a lot of changes to access control files. The num_root is one of the most efficient features for a U2R attack that provides the number of root access in the connection.

Ultimately, the computational cost on the training and testing sets of the proposed methods estimated. The Lasso method behaves as an optimization algorithm in an iterative procedure. The computational costs computed by $O(S \times M)$ for each iteration. Here, S is the total number of training records and M is the total features. The convergence includes several aspects such as step size, dataset, parameter settings etc. In most of the cases, the sparse method convergence rate is very fast, especially in the first numerous repetitions. In the experiments, the average training time of Lasso is smaller than the SVMs, presented in (Table 4). Once the Lasso model is achieved after the training phase. The test stage is very fast, only one simple linear decision function is to be executed irrespective of the size of the dataset. Therefore, the computational costs in the testing stage of SVMs are higher than the sparse model. In Table 4, a quantitative analysis of the performance of the sparse method and SVMs with different models are presented. Moreover, the table also provides an alternative comparison with different classifiers in term of detection rate (overall classification accuracy), data sizes, training time of the models and average training time per samples.

5. CONCLUSIONS

Recent literature on ML-based IDS shows that there have been quite a few attempts to address network intrusion in a comprehensive manner. Most of the work does not addresses all or one of the following aspects; method of determining proper input features, presenting feature selection, data standardization and the impact of ML with interpretable outcomes on security attack classification along with computational performance. In this paper, we have proposed the use of regularized sparse (lasso) method for selecting discriminant features and improve identification of network intrusion. We have mainly concentrated our focus on the contribution of the original input features that are well recognized within the networking domain so that it can be discovered that what types of attacks in a network are the most important.

In this regard, we have shown comprehensive simulation results where we have provided an association between the sparse model (lasso) and SVMs. We have noticed that sparse methods provide better results than SVMs; moreover, they are much more computationally efficient than SVMs. We concluded that using sparse method is easier, computationally faster and they can provide better performance with the most important features. Eventually, we can have deeper insights from the security engineering viewpoint on why the features acquired by regularized ML methods are so significant in explicitly recognizing (Dos, Probe, R2L, and U2R) numerous security attacks.

We believe that the method presented in this article may support future research on IDS. Hence, as part of our future work, we would prefer to deeply examine this problem (in terms of Group FS and Network Intrusion Detection) and verify our findings by using more realistic and recent network traffic data.

REFERENCES:

Al-mamory S. O. and F. S. Jassim, (2015) "On the designing of two grains levels network intrusion detection system," Karbala International Journal of Modern Science, vol. 1, 15-25.

Anderson, J. P. (1980) "Computer security threat monitoring and surveillance," Technical report, James P. Anderson Company, Fort Washington, Pennsylvania.

Cateni S. and V. Colla, (2016). "Improving the stability of wrapper variable selection applied to binary classification," Int. J. Comput. Inf. Syst. Ind. Manage.

Cateni, S., V. Colla, and M. Vannucci, (2017) "A fuzzy system for combining filter features selection methods," International Journal of Fuzzy Systems, vol. 19, 1168-1180.

European Cybercrime Centre (EC3 (EC3). (2017). Available:https://www.europol.europa.eu/activities-services/main-reports/internet-organised-crime-threat-assessment-iocta-2017

Friedman, J., T. Hastie, and R. Tibshirani, (2010) "Regularization paths for generalized linear models via coordinate descent," Journal of statistical software, vol. 33, 1,Pp.

Hatada M. and T. Mori, (2017) "Finding New Varieties of Malware with the Classification of Network Behavior," IEICE TRANSACTIONS on Information and Systems, vol. 100, 1691-1702.

Hagos, D. H., A. Yazidi, Ø. Kure, and P. E. Engelstad, (2017) "Enhancing Security Attacks Analysis Using Regularized Machine Learning Techniques," in Advanced Information Networking and Applications (AINA), IEEE 31st International Conference on, 909-918.

Goel, R., A. Sardana, and R. C. Joshi, (2012) "Parallel Misuse and Anomaly Detection Model," IJ Network Security, vol. 14, 211-222.

Institute C. S. and. F. investigation, (2005). "In: proceedings of the 10th annual computer crime and security "

Kohavi R. and G. H. John, (1997) "Wrappers for feature subset selection," Artificial intelligence, vol. 97, 273-324.

Li J. and Y. Qian, (2009) "Regularized multinomial regression method for hyperspectral data classification via pathwise coordinate optimization," in Digital Image Computing: Techniques and Applications,. DICTA'09. 540-545.

Liu, J., J. Chen, and J. Ye, (2009) "Large-scale sparse logistic regression," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 547-556.

Meier, L., S. Van De Geer, and P. Bühlmann, (2008) "The group lasso for logistic regression," Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 70, 53-71

Nguyen H. A. and D. Choi, (2008) "Application of data mining to network intrusion detection: classifier selection model," in Challenges for Next Generation Network Operations and Service Management, ed: Springer, 399-408.

Nguyen H. A. and D. Choi, (2008) "Application of data mining to network intrusion detection: classifier selection model," in Asia-Pacific Network Operations and Management Symposium, 399-408.

Qian, Y., M. Ye, and J. Zhou, (2013) "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," Geoscience and Remote Sensing, IEEE Transactions on, vol. 51, 2276-2291.

Shah, R. A., Y. Qian, and G. Mahdi, (2016) "Group Feature Selection via Structural Sparse Logistic Regression for IDS," in High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE 18th International Conference on, 2016, 594-600.

Shi, H., H. Li, D. Zhang, C. Cheng, and W. Wu, (2017) "Efficient and robust feature extraction and selection for traffic classification," Computer Networks, vol. 119, 1-16,

Saeys, Y., I. Inza, and P. Larrañaga, (2007) "A review of feature selection techniques in bioinformatics," bioinformatics, vol. 23, 2507-2517,

Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso," Journal of the Royal Statistical Society. Series B (Methodological), 267-288,

T. N.K. D. S. [Online]. "http://iscx.ca/NSL-KDD/."

Tavallaee, M., E. Bagheri, W. Lu, and A.-A. Ghorbani, (2009) "A detailed analysis of the KDD CUP 99 data set," in Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications .