An Investigation on Intrusion Detection System Using Machine Learning

Conference Paper · January 2019

DOI: 10.1109/SSCI.2018.8628676

Ripon Patgiri
National Institute of Technology, Silchar

Tanya Akutota
Stony Brook University

An Investigation on Intrusion Detection System Using Machine Learning

Ripon Patgiri, Udit Varshney, Tanya Akutota, and Rakesh Kunde
Department of Computer Science & Engineering
National Institute of Technology Silchar
Assam-788010, India
ripon@cse.nits.ac.in, {29udit, tanyaakutota75, rakeshkunde01}@gmail.com

Abstract—**With prevalent technologies like Internet of Things, Cloud Computing and Social Networking, large amounts of network traffic and data are generated. Hence, there is a need for Intrusion Detection Systems that monitors the network and analyzes the incoming traffic dynamically. In this paper, NSL-KDD is used to evaluate the machine learning algorithms for intrusion detection. However, not all features improve performance in a large datasets. Therefore, reducing and selecting a particular set of features improve the speed and accuracy. So, features are selected using Recursive Feature Elimination (RFE). We have conducted a rigorous experiment on Intrusion Detection System (IDS) that uses machine learning algorithms, namely, Random Forest and Support Vector Machine (SVM). We have demonstrated the comparison between the model's performance before and after feature selection of both Random Forest and SVM. We have also presented the confusion matrices.**

Index Terms—**Intrusion Detection System, Machine Learning, Random Forest, Support Vector Machine, Network Security.**

I. Introduction

With the recent advancement of technologies like Cloud Computing, Big Data, and Social Media, humongous amounts of data is being generated. It is very arduous to gain useful insights from this data, and it has become critical for marketers, data scientists and corporations. Transmitting this amount of data over a network has become a major concern. Today, there are numerous commercial intrusion detection systems (IDS) available. Also, there are numerous research works has been done on IDS which is described in Section II and also illustrated in Table I. A system, a program or a person who tries to breach network system or perform actions, not legally allowed is called as an intruder. IDS is a tool which monitors every event or packets occurring in a computer system or network and analyzes each of them to check whether the activity is malicious or not, if found malicious then take the necessary steps depends on the degree of damage can be caused by that activity.

Anomaly-based IDS makes the detection of the data packets in the network traffic, analyze the packets of data that unfit the normal profile that has been created. In this paper, we apply machine learning algorithms to detect intrusions effectively. Machine Learning is statistical methods for handling regression and classification tasks. These methods include Support Vector Machines (SVM) for regression and classification, Naive Bayes for classification, and k-Nearest Neighbors (KNN) for regression and classification.

1) Support Vector Machine (SVM): SVM method works on both regression and classification tasks by constructing optimal hyperplane which separates data in classes clearly. SVM transforms data to separate classes, and then, based on this transformation, it finds the optimal boundary. Simply it does complex data transformation, then figure out how to separate data based on labels.
2) Random forest: Random Forest is also used for both regression and classification. Random forest classifier creates a set of decision trees from randomly selected features. Then, it calculates votes from the different decision trees for each predicted target and the highest voted class is considered the final prediction. Random forest generates many classification trees. It is a better model in a goal prediction.

In this paper, we select and identify relevant features on the NSL-KDD dataset [1] which is an improvement of the KDD dataset. NSL-KDD does not have duplicate and null values like the KDD dataset. The key objectives are enumerated as follows- i

1) To investigate the application of Machine Learning for Network Intrusion Detection.
2) To develop an Intrusion Detection System that uses two algorithms: Random Forest and Support Vector Machine.
3) To reduce computational time and increase accuracy by feature selection using Recursive feature Elimination in both Random Forest and SVM.
4) To compare and present the results of these algorithms: accuracy, precision and recall.

II. Literature review

SVM maps low dimension space into high dimension space to find out a best hyper-plane to perform binary classification, such that error rate is always minimum. A comparative review is presented in Table I. SVM is trained using reduced NSL-KDD dataset [1]. According to this model, the SVM classifies a given unknown data an attack or a normal network data.

Attack types The types of attack are classified into four key categories, namely, Denial of Service, Probe, Remote to Local and User to Root. The types are defined as follows-

- Denial of Service (DoS): Blocks/restricts computer networks and systems.

978-1-5386-9276-9/18/$31.00 ©2018 IEEE

TABLE I: Literature review on IDS using Machine Learning

Author	Year	Key Outline	Remark
Zaman and Karray [2]	2009	Feature selection for intrusion detection systems based on Support Vector Machines	In this paper, features are ranked based on weights. Two algorithms have been proposed: Forward Selection ranking and Backward elimination ranking.
Jha and Ragha [3]	2013	Intrusion Detection System using Support Vector Machine	This paper explains the limitations of SVM. Next, NSL KDD is preprocessed and features are ranked based in Information Gain Ratio.Then the model is trained using SVM.
Revathi and Malathi [4]	2013	Description and analysis of NSL-KDD dataset and outline of data mining techniques	Machine learning algorithms applied on the NSL KDD dataset taking all the features. Repeated the same after selecting features using Correlation based Feature Selection Technique. Comparison of the various algorithms determined that Random Forest has the highest accuracy
Almseidin et. al. [5]	2017	Comparison of different machine learning algorithms applied for IDS	This paper calculated and compared measures such as accuracy, precision, RMS error, etc of machine learning algorithms. It also compares the true positive and false positive rates.
Thanthrige et. al. [6]	2016	Compared ML and feature selection techniques on dataset	Worked on Aegean Wi-Fi Intrusion Dataset (AWID) with different machine learning techniques. For evaluation of importance of features information gain and chi-squared statistics methods are used and features are selected bases on importance of features.
Anwer et al. [7]	2018	Feature selection for anomaly detection in networks	Presents a framework that uses filter and wrapper features selection techniques, to select the least number of features that achieve the best accuracy. UNSW-NB15 dataset is used and J48 decision tree classifier is applied.
Fadaeieslam et al. [8]	2007	Comparison of two feature selection method in IDS	A new method for feature selection based on Decision Dependent Correlation (DDC) is proposed and compared with Principle Component Analysis(PCA). The new method performs better than PCA and can effectively remove both irrelevant and redundant information.
Yun and Yang. [9]	2007	Experimental Comparison of Feature Subset Selection Methods.	Worked on different methods of feature selection which are newly proposed and make a comparison of these methods and tested with public data. Number of reduced features and the improvement of learning performance with chosen feature selection methods are measured.

- Probe: Attacker probes for vulnerabilities in a network. These can lead to attacks later
- Remote to Local (R2L): Intruder has remote unauthorized access to a system
- User to Root (U2R): Intruder who has user access later tries to access admin or root privilege

The intrusion detection system is an application that monitors a network and analyzes the packets to determine whether harmful or not. Once an abnormal activity is discovered, an alert is sent to the administrator. Intrusion detection systems are of two types: signature based and anomaly based. Signature based or rule based systems cannot detect new types of attacks. That is where anomaly-based detection comes in. It creates a baseline of what activities are ”Normal” and any activity that deviates from this, is considered an anomaly and is analyzed.

Zaman and Karray [2] works on intrusion detection system by selecting suitable features. A novel and simple method-Enhanced Support Vector Decision Function (ESVDF) is proposed for feature selection. In this method, there are two factors on which basis features are selected, namely, feature’s rank (weight), which is calculated using Support Vector Decision Function (SVDF), and correlation between features, that can be found by either Forward Selection Ranking (FSR) or Backward Elimination Ranking (BER) algorithm.

Yun and Yang [9] works on various feature selection methods and comparison among different methods. Feature selection methods deal with the selection of a few features from all features, that show the best performance in classification accuracy. By feature selection, we can reduce the cost of learning and provide better learning accuracy than all features combined.

Revathi and Malathi [4] works on the analysis of NSL-KDD data set using various machine learning algorithms. NSL-KDD dataset is an improvised version of the KDD dataset. In KDD dataset, a lot of redundancy is present in the training and testing dataset, which leads to biased results for a particular attack. In training data set 21 attacks are present and in testing data set, 37. NSL-KDD dataset doesn’t contain redundant data or duplicate records. The model is trained using various machine learning algorithms with 13 features, and it is compared with all features and accuracy of various algorithms. Jha and Ragha [3] works on intrusion detection system using SVM. In purposed system, NSL-KDD dataset is ranked using IGR and later feature subset selection is done using K-mean algorithm and svm is used for classification.

III. Proposed System

Intrusion detection system is a software application that monitors networks for malicious activities or unauthorized access. For the real-time monitoring of these malicious activities, machine learning approaches are utilized to train the model, so that a packet is dropped when it is found as a malicious packet. For training the model, various steps are shown in Figure 1 which includes **data collection**, **pre processing**, **feature selection**, **model for training**, **training**, **validation**.

A. Data Collection

For Intrusion Detection, NSL-KDD dataset [1] is used which consists of 41 features in each training and testing dataset. NSL-KDD dataset has redundant data and categorical features that needs to be preprocessed. In the data set, it

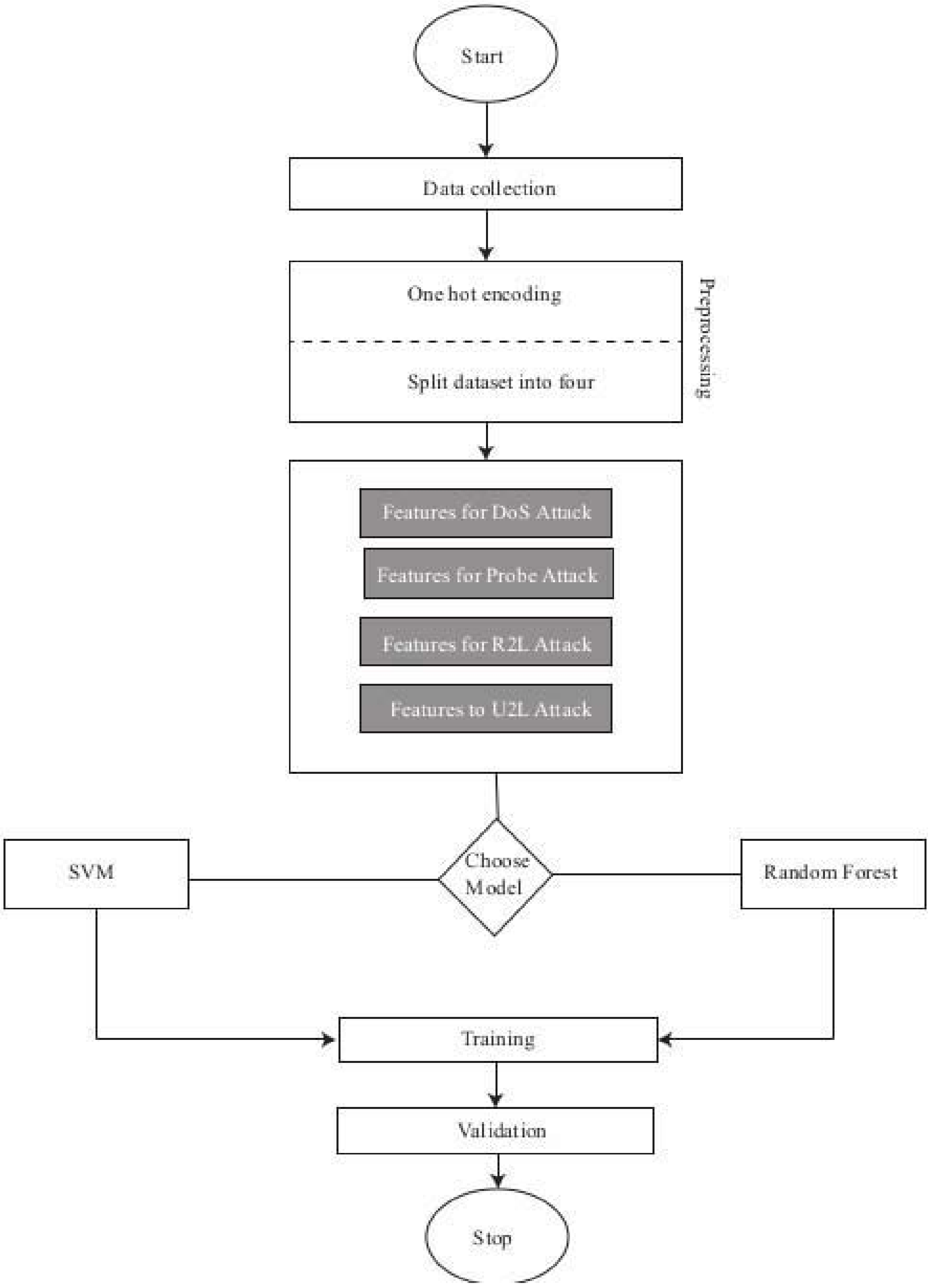


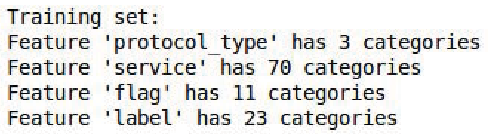
Fig. 1: Flow Diagram for the proposed system

contains 24 and 38 attack types has been found in training and testing dataset respectively.

B. Data Pre-processing

Data pre-processing transforms raw data into a more consistent format. Here, we use pre-processing to normalize and standardize the data.

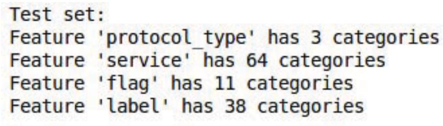
1) Identifying categorical Features: List of categorical features has been identified in training and testing datasets with number of categories in Figure 2 and Figure 3.



```
Training set:
Feature 'protocol_type' has 3 categories
Feature 'service' has 70 categories
Feature 'flag' has 11 categories
Feature 'label' has 23 categories
```

Fig. 2: Categorical Features in Training Dataset

2) One Hot Encoding: Used Machine learning algorithms can not work on categorical features. Therefore, all categorical features are being converted to binary vectors for training and testing purpose. First, categorical value is mapped to an integer value, and then, each integer is represented in binary vector



```
Test set:
Feature 'protocol_type' has 3 categories
Feature 'service' has 64 categories
Feature 'flag' has 11 categories
Feature 'label' has 38 categories
```

Fig. 3: Categorical Features in Testing Dataset

which has all 0 values except index of integer which is marked 1.

3) Adding missing category in testing data: Six categories have been found missing in service feature in testing data. These categories are stuffed with 0's.

4) Splitting Dataset: After performing one-hot encoding, different attack types found in training and testing dataset are mapped to DoS, Probe, R2L, U2R. Data is split into 4 data sets based on attack types (DoS, Probe, R2L, U2R) to train the model for all types of attacks and predict results for these attacks.

C. Feature Selection

Features are extremely important in machine learning because they are the only measurable properties of the phenomenon being observed. Choosing informative and independent features is a crucial step. Feature selection or attribute selection is a process of selecting a subset of informative features from the entire set. Feature selection ways are used to discover influenced factors and remove unnecessary, superfluous attributes from data which do not affect the accuracy of predictive models, if they are included or not, or may in fact decrease the accuracy of the model. Feature selection and Feature extraction are different. The key difference between feature selection and extraction is that feature selection tries to find out the best subset of features among original features while feature extraction creates a set of new features.

1) Filter Method: Filter method applies static measures to calculate score for each feature. A feature is either selected or discarded from dataset based on the score of each feature. For example, information gain, Chi squared test and correlation coefficient score.

2) Wrapper Method: Wrapper methods are similar to a search problem, where features are prepared in different combination, evaluated and compared to other combinations. A predictive model is used which assigns a score based on model accuracy for evaluation of features. Searching can be stochastic such as random hill-climbing algorithm, or heuristics, like forward and backward passed to add and remove features. For instance, a recursive feature elimination algorithm.

3) Embedded Method: Embedded method checks each features that increases accuracy of model while model is being created.

D. Recursive Feature Elimination(Method Used):

Algorithm 1 Recursive Feature Elimination With Random Forest

```
Train the random forest model with full feature set
Evaluate the model performance with RMSE and rank feature importance.
for $i = 1\ to\ n$ do
    $Eliminate\ last\ d\ features\ with\ smallest\ importance$
    $Train\ the\ rf\ model\ with\ tunes\ subset$
    $Evaluate\ the\ model\ performance\ with\ RMSE\ and$
    $rank\ feature\ importance$
end for
Select the optimal Feature Length and its feature rank
```

Algorithm 2 Recursive Feature Elimination With SVM

```
    Input Initial data subset G=1,2..n
    Output Rank list according to smallest weight criteria,R.
1: Set R={}
2: Repeat steps 3 to 8 until G is not empty.
3: Train the SVM using G
4: compute the weight vector
5: Compute the ranking criteria.
6: Rank the features as in sorted manner.
   $New_rank = Sort(Rank)$
7: Update the feature rank list
   $Update\ R = R + G(New_rank)$
8: Eliminate the feature with the smallest rank
   $Update\ G = G - G(New_rank)$
```

RFECV is used to find the ranks together with the optimal number of features via cross validation

E. Features selected for Random Forest Classifier using RFE:

- **Features for DoS:** [logged_in, count, 'serror_rate', 'srv_serror_rate', 'same_srv_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_serror_rate', dst_host_srv_serror_rate', 'service_http', 'flag_S0', 'flag_SF'].
- **Features for Probe:** ['logged_in', 'rerror_rate', 'srv_rerror_rate', 'dst_host_srv_count', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'dst_host_rerror_rate', 'dst_host_srv_rerror_rate', 'Protocol_type_icmp', 'service_eco_i', 'service_private', 'flag_SF']
- **Features for R2L:** ['src_bytes', 'dst_bytes', 'hot', 'num_failed_logins', 'is_guest_login', 'dst_host_srv_count', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'service_ftp', 'service_ftp_data', 'service_http', 'service_imap4', 'flag_RSTO']
- **Features for U2L:** ['urgent', 'hot', 'root_shell', 'num_file_creations', 'num_shells', 'srv_diff_host_rate', 'dst_host_count',

’dst_host_srv_count’, ’dst_host_same_src_port_rate’, ’dst_host_srv_diff_host_rate’, ’service_ftp_data’, ’service_http’, ’service_telnet’]

F. Features selected for SVM Classifier using RFE:

- **Features for DoS:** [’duration’, ’wrong_fragment’, ’hot’, ’num_compromised’, ’num_root’, ’num_file_creations’, ’is_guest_login’, ’srv_count’, ’dst_host_rerror_rate’, ’Protocol_type_icmp’, ’Protocol_type_tcp’, ’Protocol_type_udp’, ’flag_S0’]
- **Features for Probe:** [’hot’, ’logged_in’, ’count’, ’rerror_rate’, ’dst_host_count’, ’dst_host_same_srv_rate’, ’dst_host_same_src_port_rate’, ’dst_host_rerror_rate’, ’Protocol_type_udp’, ’service_eco_i’, ’service_http’, ’service_private’, ’flag_RSTOS0’]
- **Features for R2L:** [’hot’, ’num_root’, ’count’, ’srv_count’, ’srv_rerror_rate’, ’same_srv_rate’, ’dst_host_count’, ’dst_host_srv_count’, ’dst_host_same_srv_rate’, ’Protocol_type_tcp’, ’service_ftp_data’, ’service_other’, ’flag_REJ’]
- **Features for U2R:** [’duration’, ’hot’, ’is_guest_login’, ’dst_host_count’, ’dst_host_srv_count’, ’dst_host_same_srv_rate’, ’dst_host_serror_rate’, ’Protocol_type_tcp’, ’Protocol_type_udp’, ’service_ftp’, ’service_ftp_data’, ’service_http’, ’service_telnet’]

IV. Classification

A classification model attempts to draw some conclusion from observed values.We have trained data set on Random forest and Support Vector Machine.

A. Random Forest

Random Forest is a supervised classification algorithm. Random forest can be used for regression and classification tasks. Random forest classifier forms a bunch of number of decision trees from randomly selected features. Then, it calculates votes from the different decision tree for each predicted target and the highest voted class is considered the final prediction. Let training set is provided as: **[A1, A2, A3, A4]** with their corresponding label as **[B1, B2, B3, B4]** random forest can generate three decision tree taking a subset of input, for example

1.**[A1, A2, A3]**
2.**[A1, A2, A4]**
3.**[A2, A3 A4]**

Finally, it predicts based on the majority of votes from each decision made the decision trees. The random-forest algorithm brings extra randomness into the model while growing the trees. Instead of searching best feature while splitting node, it searches for the best features among random random subset of features.

B. Support Vector Machine

Support vector machine is a supervised classification algorithm. SVM is a discriminative classifier that separates defined by separating hyper-plane. More clearly SVM takes training data and separates data into categories divided by a clear gap called the hyper-plane. SVM tries to find out best or optimal hyper-plane, which has the largest distance from the nearest point, in high dimensions, which clearly separates training set into categories. Support vectors are those vectors which are nearest to the hyper-plane. The goal is to select a hyper-plane having margin as much as possible between hyperplane and any vector within training set, giving a greater chance of new data being classified correctly.

V. Prediction and Evaluation

Prediction is performed after training the model using SVM or Random Forest on the given dataset for various attacks that is represented by confusion matrices. A confusion matrix is a technique for summarizing the performance of the classification algorithm. Accuracy, precision, and recall are calculated using the information given in a confusion matrix.

TABLE II: Confusion Matrix

		Predicted	
		NO	YES
Actual	NO	$\sum TN$	$\sum FP$
	YES	$\sum FN$	$\sum TP$

Accuracy can be calculated by confusion matrix as follows

$$\frac{TP+TN}{TP+TN+FP+FN}$$

Precision can be calculated by confusion matrix as follows

$$\frac{TP}{TP+FP}$$

Recall can be calculated by confusion matrix as follows

$$\frac{TP}{TP+FN}$$

VI. Experimental Results and Discussions

We have conducted an extensive experiment to plot the number of attacks vs the type. Figure 4 shows different types of attacks found in the training data set. Log values of the attack counts are plotted for the various attack types. In the data set, normal packets are found to be the largest in number. All the attacks plotted are categorized into four basic attacks, particularly, Denial-of-Service (DoS), Probe, Remote to Location (R2L), and User to Root (U2R).

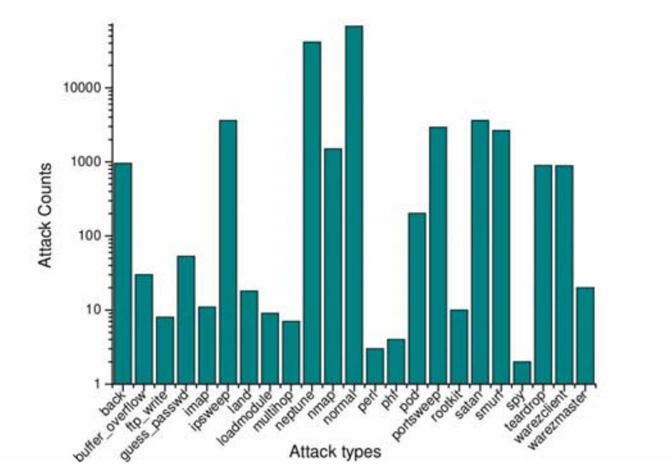


Fig. 4: Attacks types in training data set. Here x-axis shows types of attacks found in dataset and y-axis shows log value of number of attacks of each type.

Our proposed model is trained for each attack type using all features present in training data using random forest. It is tested using testing data. The prediction results are shown in confusion matrix for each attack type. Figure 6 extrapolates the accuracy of each model. Accuracy is calculated by confusion matrix as depicted in Figure 5.

		Predicted	
		NO	YES
Actual	NO	9663	48
	YES	2651	4809

(a) DoS Attack

		Predicted	
		NO	YES
Actual	NO	9195	516
	YES	1086	1335

(b) Probe Attack

		Predicted	
		NO	YES
Actual	NO	9711	0
	YES	2885	0

(c) R2L Attack

		Predicted	
		NO	YES
Actual	NO	9711	0
	YES	67	0

(d) U2R Attack

Fig. 5: Confusion matrix for different type of attaks (DoS, Probe, R2L, U2R) using Random Forest algorithm using all features.

Again, our proposed model is trained for each attack type using the best 13 features present in the training data using Random Forest in Figure 7. Features are selected using RFE method. It is tested using testing data. The prediction results are depicted in Figure 8 confusion matrix for each attack type and also, Figure 8 depicts accuracy of each model. Accuracy is calculated by confusion matrix.

For SVM, our model has been trained for each attack type using all features present in training data. Features are selected using RFE method. Model has been tested using testing data and prediction as depicted in Figure 9. Figure 10 extrapolates accuracy of each model. Accuracy has calculated by confusion matrix.

Also, our model has been trained for each attack type using best 13 features present in training data using SVM. Features are selected using RFE method. Model has been tested using testing data and prediction as shown in Figure 11. Figure 12 extrapolates accuracy of each model. Accuracy is calculated by resultant confusion matrix.

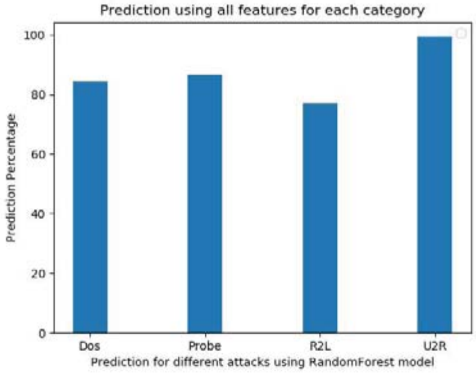


Fig. 6: Accuracy result for DoS, Probe, R2L, U2R that is predicted using Random Forest algorithm using all features. Here X-axis shows type of attack and Y-axis shows accuracy percentage for each type of attack.

		Predicted	
		NO	YES
Actual	NO	9230	481
	YES	2248	5212

(a) DoS Attack

		Predicted	
		NO	YES
Actual	NO	9337	374
	YES	1033	1338

(b) Probe Attack

		Predicted	
		NO	YES
Actual	NO	9711	0
	YES	2832	53

(c) R2L Attack

		Predicted	
		NO	YES
Actual	NO	9710	1
	YES	57	10

(d) U2R Attack

Fig. 7: Confusion matrix for different type of attaks (DoS, Probe, R2L, U2R) using Random Forest algorithm using 13 features.

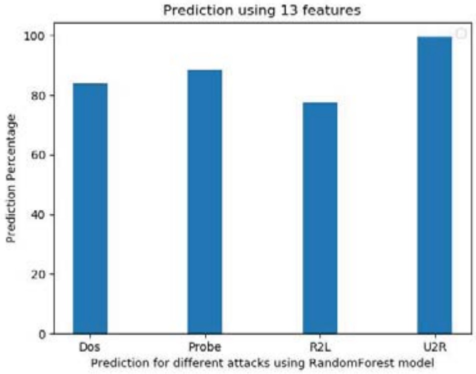


Fig. 8: Accuracy result for DoS, Probe, R2L, U2R that is predicted using Random Forest algorithm using 13 features. Here X-axis shows type of attack and Y-axis shows accuracy percnetage for each type of attack.

		Predicted	
		NO	YES
Actual	NO	9455	256
	YES	1359	6101

(a) DoS Attack

		Predicted	
		NO	YES
Actual	NO	9576	135
	YES	1285	1136

(b) Probe Attack

		Predicted	
		NO	YES
Actual	NO	9639	72
	YES	2737	148

(c) R2L Attack

		Predicted	
		NO	YES
Actual	NO	9710	1
	YES	67	0

(d) U2R Attack

Fig. 9: Confusion matrix for different type of attaks (DoS, Probe, R2L, U2R) using SVM algorithm using all features.

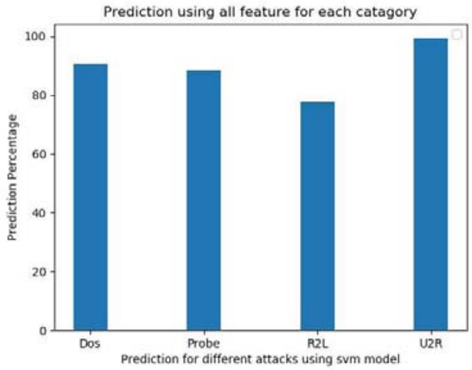


Fig. 10: Accuracy result for DoS, Probe, R2L, U2R that is predicted using SVM algorithm using all features. Here X-axis shows type of attack and Y-axis shows accuracy percentage for each type of attack.

		Predicted	
		NO	YES
Actual	NO	9596	115
	YES	3101	4359

(a) DoS Attack

		Predicted	
		NO	YES
Actual	NO	9556	155
	YES	1825	596

(b) Probe Attack

		Predicted	
		NO	YES
Actual	NO	9704	7
	YES	2882	3

(c) R2L Attack

		Predicted	
		NO	YES
Actual	NO	9708	3
	YES	59	8

(d) U2R Attack

Fig. 11: Confusion matrix for different type of attacks (DoS, Probe, R2L, U2R) using SVM algorithm using 13 features.

A. RFECV with Random Forest

A technique which is used to assess predictive models by partitioning the original sample into a training set to train model, and a test set to valuate is known as Cross-validation. Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. Moreover, recursive feature elimination method is used to select number of features that give best result based on random forest. Cross Validation score is extrapolated in Figure 13 and Figure 14 for the selected number of features.

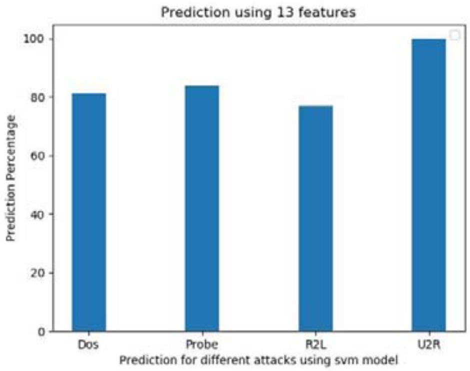
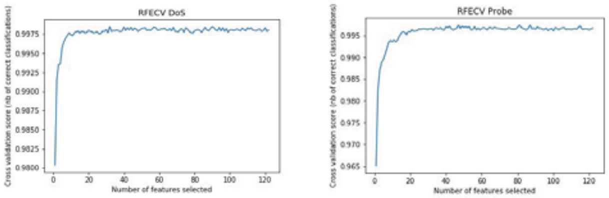
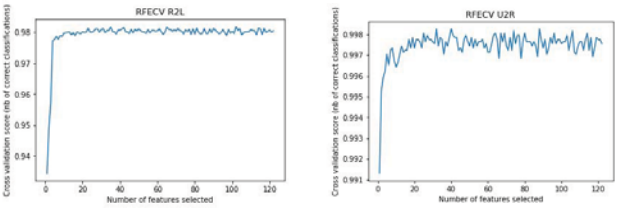


Fig. 12: Accuracy result for DoS, Probe, R2L, U2R that is predicted using SVM algorithm using 13 features. Here X-axis shows type of attack and Y-axis shows accuracy percentage for each type of attack.



(a) Cross validation score for DoS (b) Cross validation score for Probe

Fig. 13: Cross validation using RFECV with Random Forest for DoS and Probe attack. Here x-axis shows number of features selected and y-axis shows cross validation score for selected features.

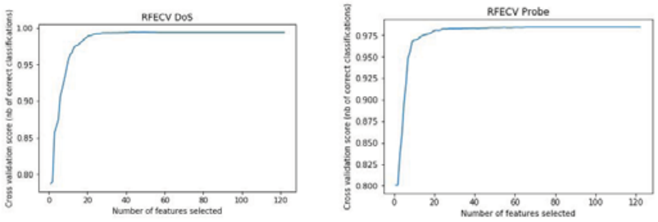


(a) Cross validation score for R2L (b) Cross validation score for U2R

Fig. 14: Cross validation using RFECV with Random Forest for R2L and U2R attack. Here x-axis shows number of features selected and y-axis shows cross validation score for selected features.

B. RFECV with SVM

Recursive feature elimination method is used to select number of features that gives best result based on SVM. It ranks features and eliminates features one by one whose rank is low. Cross-Validation score is disclosed in Figure 15 and Figure 16 for the selected number of features.



(a) Cross validation score for DoS (b) Cross validation score for Probe

Fig. 15: Cross validation using RFECV with SVM for DoS and Probe attack. Here x-axis shows number of features selected and y-axis shows cross validation score for selected features.

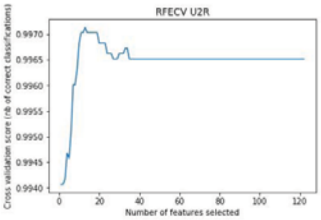


Fig. 16: Cross validation score for U2R using RFECV with SVM. Here x-axis shows number of features selected and y-axis shows cross validation score for selected features.

C. Comparison

Comparison between Random Forest and SVM (all feature) is exposed in Figure 17. The Figure 17 shows comparison of accuracy for each model (Random Forest and SVM) for each type of attack (DoS, Probe, R2L, U2R). SVM performs better than Random Forest.

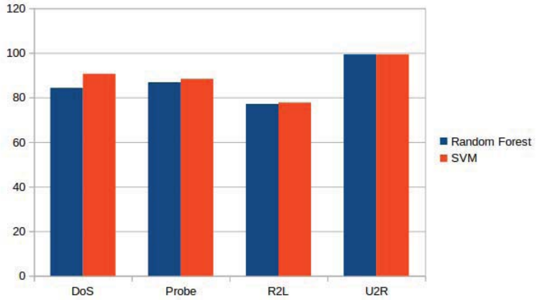


Fig. 17: Accuracy comparison between Random Forest and SVM algorithm using all features. X-axis shows type of attacks DoS, Probe, R2L, U2R and Y-axis shows accuracy percentage predicted by Random Forest and SVM.

In addition, comparison between Random Forest and SVM (13 feature) is extrapolated in Figure 18. Figure 18 depicts comparison of accuracy for each model (Random Forest and SVM) for each type of attack (DoS, Probe, R2L, U2R). Random Forest performs better than SVM.

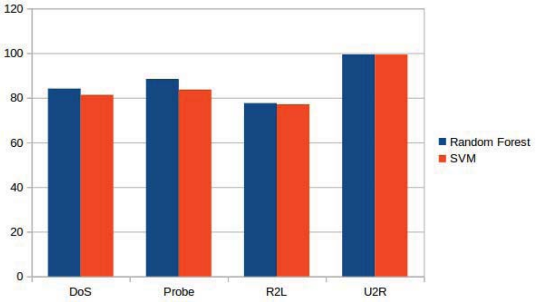


Fig. 18: Accuracy comparison between Random Forest and SVM using 13 features. X-axis shows type of attacks DoS, Probe, R2L, U2R and Y-axis shows accuracy percentage predicted by Random Forest and SVM

VII. Conclusion

In this paper, we presented our proposed model on intrusion detection that worked on two machine learning algorithms, namely, Random Forest and SVM. We have conducted an extensive experimentation on these two machine learning algorithms using all the features, and it is observed to be time consuming and performance degrading. As per our experience, a few of the features in the dataset are redundant and irrelevant. Therefore, feature selection also plays a vital role in our proposed work. Recursive Feature Elimination is deployed to reduce the dimensionality of the dataset. In the comparison, Random forest performed better than SVM before feature selection. On the contrary, SVM performed better than Random Forest after feature selection for most of the attacks.

References

[1] C. H. Low, “NSL-KDD dataset,” Retrieved from https://github.com/defcom17/NSL_KDD.

[2] S. Zaman and F. Karray, “Features selection for intrusion detection systems based on support vector machines,” in *2009 6th IEEE Consumer Communications and Networking Conference*, Jan 2009, pp. 1–8.

[3] J. Jha and L. Ragha, “Intrusion detection system using support vector machine,” *International Journal of Applied Information Systems (IJAIS)*, 2013.

[4] Revathi and Malathi, “A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection,” *International Journal of Engineering Research and Technology*, 2013.

[5] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, “Evaluation of machine learning algorithms for intrusion detection system,” in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, Sept 2017, pp. 000 277–000 282.

[6] U. S. K. P. M. Thanthrige, J. Samarabandu, and X. Wang, “Machine learning techniques for intrusion detection on public dataset,” in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2016, pp. 1–4.

[7] H. M. Anwer, M. Farouk, and A. Abdel-Hamid, “A framework for efficient network anomaly intrusion detection with features selection,” in *2018 9th International Conference on Information and Communication Systems (ICICS)*, April 2018, pp. 157–162.

[8] M. J. Fadaeieslam, B. Minaei-Bidgoli, M. Fathy, and M. Soryani, “Comparison of two feature selection methods in intrusion detection systems,” in *Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on*. IEEE, 2007, pp. 83–86.

[9] C. Yun and J. Yang, “Experimental comparison of feature subset selection methods,” in *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 367–372.