Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 190 (2021) 564–570

Procedia Computer Science

www.elsevier.com/locate/procedia

2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society

Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features

Mechetin Artur[a]

[a]*Department of Computer Systems and Technologies, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow 101000, Russia*

Abstract

In this paper I propose a wrapped feature selection method using Recursive Feature Elimination and Cross-validated selection. In my work I use Bernoulli Naïve Bayes classifier on the NSL-KDD dataset.

Keywords: Network Security; NSL-KDD dataset; Bernoulli Naïve Bayes classifier; Naïve Bayes classifier; RFE; Recursive Feature Eliminaion

1. Introduction

Network security is one of the most actual problems nowadays. Organizations often deploy a firewall as a first line of defense in order to protect their private network from malicious attacks, but there are several ways to bypass the firewall which makes Intrusion detection system a second line of defense and a way to monitor the network traffic for any possible threat or illegal action [1].

Intrusion Detection Systems (IDS) provide an additional layer of protection for computer systems. IDS are used to detect certain types of malicious activity that can compromise the security of a computer system. Such activity includes network attacks against vulnerable services, privilege escalation attacks, unauthorized access to sensitive files, and malicious software (computer viruses, Trojans, and worms).

The accuracy of intrusion detection is one of the main components of IDS quality. To achieve maximum intrusion detection accuracy it is necessary to have a high quality data set. One way to obtain a high quality dataset is the feature selection. Feature selection is a crucial step in most classification problems which reduces the learning time and enhances the predictive accuracy [2].

10.1016/j.procs.2021.06.066

1.1. Feature selection

Feature selection algorithms are classified such as filter, wrapper and embedded methods. The filter methods are based on statistical methods and, as a rule, consider each feature independently. They allow us to estimate and rank the features according to their significance, which is taken as the degree of correlation of this feature with the target variable. The filter methods are much faster than wrapper and embedded methods. Moreover, they work well even when the number of features exceeds the number of examples in the training set. The essence of wrapper methods is that the classifier is run on different subsets of features of the original training set. Then a subset of features with the best parameters on the training sample is chosen. And then it is tested on the test set. All wrapper methods require much more computation than filtering methods. In case of large number of features and small training dataset size, the wrapper methods have a risk of overfitting. Embedded methods, do not allow to separate feature selection and classifier training, but select within the model computation process. In addition, the embedded methods require less computation than wrapper methods, but more than filtering methods.

1.2. Recursive Feature Elimination

In this research paper, I use the wrapper method Recursive Feature Elimination (RFE). RFE works by recursively removing features and building a model based on the remaining features. It uses model accuracy to determine which features (and combinations of features) contribute the most to predicting the target feature. RFE requires a specified number of features to keep, however it is often not known in advance how many features are optimal. To find the optimal number of features cross-validation is used with RFE to score different feature subsets and select the best scoring collection of features.

1.3. Cross-validation

Cross-validation (CV) is a procedure for empirically evaluating the generative ability of algorithms trained on precedents. The algorithm fixes some set of partitions of the original sample into two subsamples: a training subsample and a control subsample. For each partition the algorithm is tuned for the training subsample, and then its average error on the objects of the control subsample is estimated. The sliding control estimate is the average error on the control subsamples for all partitions. In this paper I will use Repeated Stratified K-Fold Cross-validation to evaluate the quality of the algorithm. In the Repeated Stratified K-Fold Cross-validation. The data sample will be shuffled prior to each repetition where each subset contains approximately the same percentage of samples of each target class as the complete set.

1.4. Naïve Bayes

The algorithm is based on Bayes' theorem and assumes that the features are independent of each other

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{1}$$

In this article I use the Naïve Bayes Bernoulli classification algorithm. This algorithm is well suited for binary classification [3].

$$p(x \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)} \tag{2}$$

where p_{ki} is the probability of class C_k generating the term x_i.

1.5. NSL-KDD dataset

NSL-KDD is a data set suggested to solve some of the inherent problems of the KDD'99 data set which are mentioned in [4]. The NSL-KDD has the following advantages over KDD'99:

- This does not include repetitive records, so classifiers will not be biased toward more frequent records
- The number of records in the training and test dataset is logical, making it more convenient to experiment with the entire dataset without having to select random small segments. In this way, the results of evaluating different jobs will be stable
- The number of records selected from each difficulty level group is inversely proportional to the percentage of records in the original KDD dataset. As a result, the classification levels of different machine learning methods vary over a wider range, which makes it more efficient to obtain accurate estimates of different learning methods.

Each record of the NSL-KDD has 42 features and separated by 3 data types: continuous, discrete and categorical data. The data is separated into 4 types into attack: Denial of Service (DoS), Probe, R2L, U2R.

Table 1. NSL-KDD dataset.

No.	Feature name	No.	Feature name	No.	Feature name
1	Durtaion	15	su_attempted	29	same_srv_rate
2	protocol_type	16	num_root	30	diff_srv_rate
3	service	17	num_file_creations	31	srv_diff_host_rate
4	flag	18	num_shells	32	dst_host_count
5	src_bytes	19	num_access_files	33	dst_host_srv_count
6	dst_bytes	20	num_outbound_cmds	34	dst_host_same_srv_rate
7	land	21	is_host_login	35	dst_host_diff_srv_rate
8	wrong_fragment	22	is_guest_login	36	dst_host_same_src_port_rate
9	urgent	23	count	37	dst_host_srv_diff_host_rate
10	hot	24	srv_count	38	dst_host_serror_rate
11	num_failed_logins	25	serror_rate	39	dst_host_srv_serror_rate
12	logged_in	26	srv_serror_rate	40	dst_host_rerror_rate
13	num_compromised	27	rerror_rate	41	dst_host_srv_rerror_rate
14	root_shell	28	srv_rerror_rate	42	class

2. Data preprocessing

Data preprocessing is an important task that must be done before the data set can be used to train the model. Unprocessed data is often garbled and unreliable, and values may be missing from it. Using such data in modeling can lead to incorrect results.

The NSL-KDD dataset, as mentioned above, does not have major problems with data quality. However, some processing will still have to be performed.

Discretization of categorical data is necessary for accurate classifier performance. The features ‘protocol_type’ , ‘service’ and ‘flag’ will be converted to discrete values according to their value. The feature 'class' will have a binary representation, where ‘1’ is a normal label and ‘0’ is an attack.

3. Experiments and results

In the experiment, I applied Repeated Stratified K-Fold Cross-validation with 10 splits and 5 repetitions to a full NSL-KDD training dataset. The results of the evaluations in the iterations were averaged. From the results of the experiment, it turned out that the optimal number of features is 32. The correlation between the cross-validation score and the number of features is shown on fig. 1. Their ranks are shown in the table below.

Table 2. Feature ranking.

Feature name	Rank	Feature name	Rank
duration	1	land	1
num_outbound_cmds	1	num_file_creations	1
dst_host_rerror_rate	1	urgent	1
is_guest_login	1	hot	1
serror_rate	1	wrong_fragment	1
srv_serror_rate	1	logged_in	1
rerror_rate	1	num_compromised	1
num_access_files	1	root_shell	1
srv_rerror_rate	1	su_attempted	1
srv_diff_host_rate	1	num_failed_logins	1
dst_host_diff_srv_rate	1	dst_host_srv_rerror_rate	1
dst_host_same_src_port_rate	1	dst_byes	2
dst_host_srv_diff_host_rate	1	src_bytes	3
dst_host_serror_rate	1	service	4
dst_host_srv_serror_rate	1	dst_host_same_srv_rate	5
diff_srv_rate	1	same_srv_rate	6
num_shells	1	count	7
is_host_login	1	srv_count	8
num_root	1	dst_host_count	9
protocol_type	1	dst_host_srv_count	10
flag	1		

3.1. Perfomance evaluation

Predictive accuracy is a poor measure and sometimes a misleading performance indicator especially in a skewed dataset [5].

There are several methods to assess the quality of the classifier, I will use the following:

- F-measure
- AUC ROC

F-measure is one of the effective evaluation metrics that is based on a combination of precision and recall. Alone, neither accuracy nor recall can accurately express the quality of an algorithm. We can have excellent accuracy with terrible recall or, alternatively, terrible accuracy with excellent recall. The F-measure allows us to express both problems with a single score. The larger the F-measure value, the higher the classification quality.

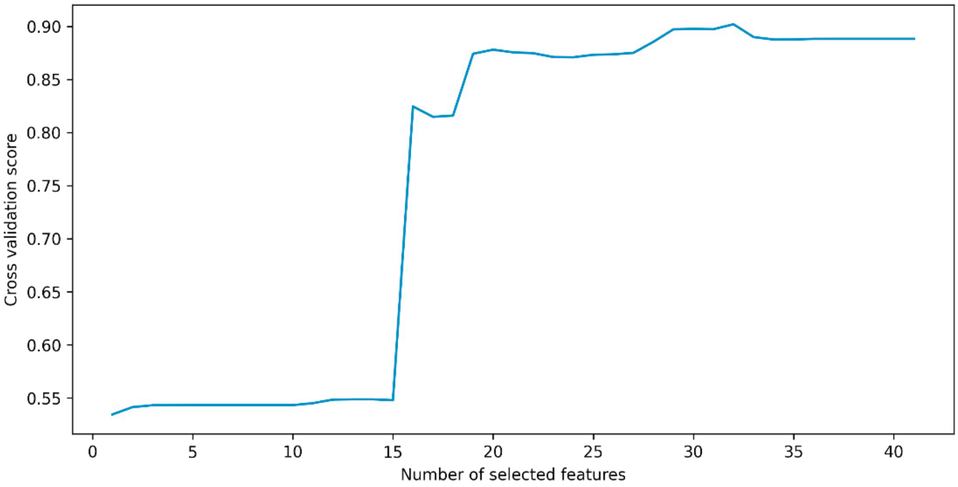

Fig. 1. result of the RFECV

$$recall = \frac{TP}{TP + FN} \tag{3}$$

$$precision = \frac{TP}{TP + FP} \tag{4}$$

$$F_{measure} = 2\frac{precision \times recall}{precision + recall} \tag{5}$$

TP – correctly classified positive examples
TN – correctly classified negative examples
FP – negative examples classified as positive
FN – positive examples classified as negative

ROC (Receiver Operator Characteristic) is the curve that is most often used to represent binary classification results in machine learning. The ROC curve shows the relationship between the number of correctly classified positive examples and the number of incorrectly classified negative examples. ROC score is calculated by the area under the curve. The numerical area under the curve is called the AUC (Area Under Curve) [6]. The higher the AUC, the better the prognostic power of the model. However, should be aware that:

- AUC is more for comparative analysis of several models
- AUC contains no information about the sensitivity or specificity of the model

Figure 2 shows the ROC curve with AUC. The result is in Table 3.

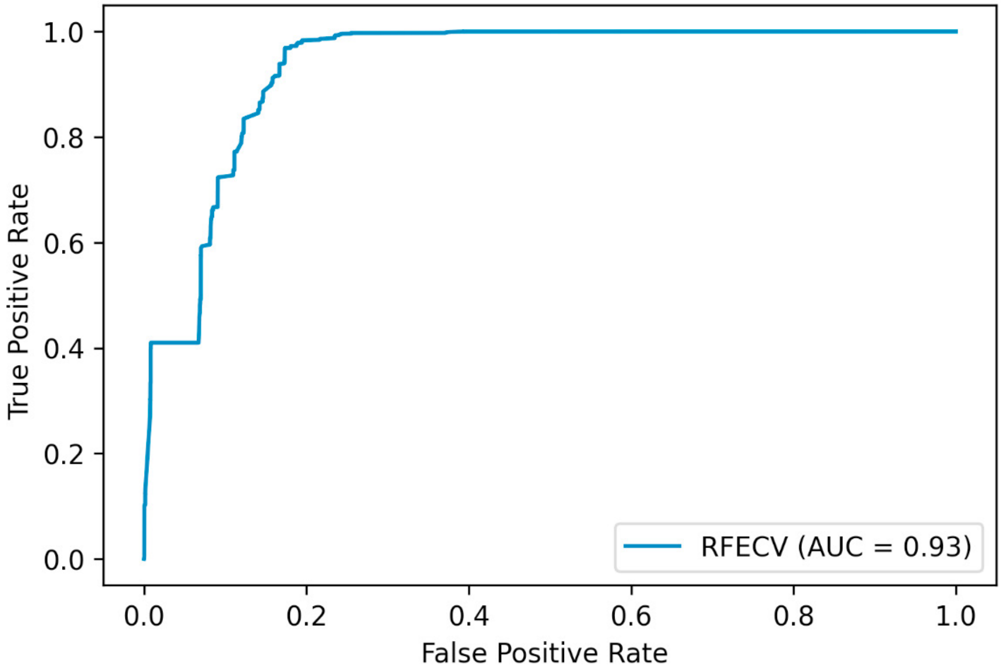

Fig. 2. ROC curve the result of experiment

Table 3. Results of the experiment

Evaluation metric	Value
recall	0.97
precision	0.87
F-measure	0.9
AUC ROC	0.93

4. Conclusion

In this paper the work of the Naïve Bayes classifier in combination with method of features selection FRECV was reviewed. The results of stratified cross-validation with 10 folds and 5 repetitions showed that for binary classification by Naïve Bayes method the optimal number of features is 32. In addition to the number of features, the name of these features was also obtained. The F-measure and AUC ROC scores indicate that the binary classification by the Bernoulli Naïve Bayes algorithm works well.

The results of this study can be used as a basis for new research or to summarize existing research in this area.

References

[1] M. Bahrololum, E. Salahi, and M. Khaleghi. (2009) "Machine Learning Techniques for Feature Reduction in Intrusion Detection Systems: A Comparison," *Fourth Int. Conf. Comput. Sci. Converg. Inf. Technol*

[2] Z. Karimi and A. Harounabadi. (2013) "Feature Ranking in Intrusion Detection Dataset using Combination of Filtering Methods," *Int. J. Comput. Appl.* **78** (**4**): 21–27.

[3] McCallum Andrew, Nigan Kamal. (1998) "A comparison of event models for Naive Bayes text classification." *AAAI-98 workshop on learning for text categorization*. 752.

[4] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani. (2009) "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on *Computational Intelligence for Security and Defense Applications* (CISDA).

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. (2002) "SMOTE: Synthetic minority oversampling technique" *Journal of Artificial Intelligence Research* **16**: 321–357.

[6] Zweig M.H., Campbell G. (1993) "ROC Plots: A Fundamental Evaluation Tool in Clinical Medicine" *Clinical Chemistry*, **39** (**4**).