A Novel Algorithm for Feature Selection Used in Intrusion Detection

Yongle Hao[1,2(✉)], Ying Hou[3], and Longjie Li[1,3]

[1] China Information Technology Security Evaluation Center, Beijing, China
haoyl@itsec.gov.cn, li.longjie@hotmail.com
[2] Beijing University of Posts and Telecommunications, Beijing, China
[3] Lanzhou University, Lanzhou, China
houyl5@lzu.edu.cn

Abstract. Intrusion detection systems play an important role in securing computer networks. The existing methods for intrusion detection deal with huge amount of data which contains irrelevant or redundant features. Accordingly, feature selection is critical for improving classification accuracy in an intrusion detection system. In this paper, we proposed a novel algorithm combining a variety of feature selection methods based on majority voting rule, and used the SVM as the basic classification algorithm. Experiments on NSL-KDD dataset indicate that the proposed algorithm selects superior feature subset than the state-of-the-art feature selection approaches used in the field of intrusion detection.

Keywords: Intrusion detection systems · Feature selection · SVM · Majority voting algorithm

1 Introduction

Nowadays, computer networks play an important role in people's work and life. The security of network has become very crucial in protecting the confidentiality, integrity and availability of computer systems [1]. Intrusion detection systems (IDSs), which detect attack activities by analyzing network traffics, are excellent security countermeasure in securing network.

In recent years, a growing number of machine learning methods have been used in the area of network intrusion detection [2]. One reason is huge amount of network traffic data brings about a big challenge for IDSs. So, data reduction method is a need when building an intrusion detection model.

Feature selection is a pre-processing step to reduce the number of features in a machine learning task, which focuses on removing redundant and irrelevant features from all features [3]. Generally speaking, one feature can be defined as relevant when it is highly correlated with class labels, redundant when it doesn't provide more information than any relevant feature, or irrelevant when it is uncorrelated with class labels. Most relevant features are selected by using a feature selection method before the process of training a model. In general, feature selection methods are divided into three kinds: wrapper-based, filter-based and embedded methods [4]. Wrapper-based methods

L. Barolli and T. Enokido (eds.), *Innovative Mobile and Internet Services in Ubiquitous Computing*, Advances in Intelligent Systems and Computing 612,
DOI 10.1007/978-3-319-61542-4_98

use classifiers to score given feature subsets according to their predictive power. Filter-based methods do not use classification model, but rely on the general characteristics of the training data to choose feature subsets with high ranking scores. Embedded methods select features in the process of training, which are usually specific to given classifiers. In those cases in which the number of features is lager, filter-based methods are computationally more efficient [4].

In practice, different feature selection methods may generate different feature subsets. Hence, how to choose an appropriate feature selection method is still a problem in building an intrusion detection model. Inspired by the idea of ensemble learning, we propose a new feature selection algorithm in this paper, in which several different feature selection methods are employed and the significance features are elected based on the majority voting algorithm.

In this paper, five feature selection methods, i.e., CFS [5], LS [6], ReliefF [7, 8], χ^2 [9] and UDFS [10], are integrated into our algorithm. The description of these methods will be given in Sect. 2. In this paper, we present a new intrusion detection model by combining the SVM (support vector machine) algorithm [11] and our feature selection approach. This model is marked as *SVM+Our*. In order to verify the performance of our approach, we also build other five detection models by combing SVM and the above five methods, respectively. The five models are *SVM+CFS*, *SVM+LS*, *SVM+ReliefF*, *SVM+*χ^2 and *SVM+UDFS*, respectively. We conduct experiments on the NSL-KDD dataset [12]. The results show that the SVM+Our model achieves better performance than others.

The remainder of this paper is organized as follows. In Sect. 2, the feature selection methods used in this work are introduced. Section 3 describes the proposed feature selection algorithm and Sect. 4 experimentally evaluates the performance of the proposed algorithm. The conclusion of this work is given in Sect. 5.

2 Feature Selection Methods

In many practical applications, the number of features is high. High dimensional feature space will decrease the efficiency and accuracy of a predictive model. Feature selection can eliminate irrelevant or redundant features, and then improve the classification accuracy of a model and reduce the running time. Therefore, feature selection is an important step before building a model for classification problem.

Feature selection can be understood as finding a subset of features that leads to the largest possible generalization. Among the common feature selection strategies, *ReliefF* [7, 8] belongs to a kind of feature weighting algorithms, which estimates the quality of the features according to the correlation of features and classes.

Selecting relevant features in unsupervised learning cases is hard due to the absence of class labels. *Laplacian Score* (LS) is an unsupervised method, in which the importance of a feature is evaluated by its locality preserving power [6]. LS value is

computed based on the fact that two samples are probably related to the same class, if they are close to each other.

Nguyen [5] focused on the *Correlation based Feature Selection* (CFS) method, which aims to calculate the connection degree of feature subset according to the correlation formula. CFS method was developed by Hall in 2000, which goal is to choose optimal feature subset highly associated with the class based on heuristic estimating [13].

Chi-square (χ^2) method was proposed based on the statistical theory [9], which used to measure the statistical correlation between the features and classes. The method describes the independence or the deviation degree between the actual values and expectations. The chi-square values between attributes and class labels are calculated, sorted based on the principle of statistics. The optimal feature subset is selected from the sorted chi-square values.

Unsupervised Discriminative Feature Selection (UDFS) algorithm aims to select the most discriminative features. This algorithm considers to use the manifold structure, which makes it different from the existing unsupervised feature selection algorithms [10].

3 The Proposed Algorithm

For large-scale data, feature filtering method is usually adopted to reduce the dimension, speed up the execution efficiency and improve the detection accuracy. However, the significance of each feature is inconsistent by using different feature selection methods, then the important features selected by different methods are also different. So far, there is not a unified standard to judge the merit of each feature selection method. The selection of important features has brought certain difficulties for us. In this paper, we propose a novel feature selection approach to solve the above problem.

The proposed approach combines several feature selection methods based on the majority voting algorithm, which is the most simple and effective way in terms of data fusion. In this paper, majority voting algorithm is employed as the basic selection strategy to form a new feature selection algorithm.

Given the dataset D, $X_1, X_2, \cdots, X_n$ are samples in D. For each sample, it has m features: $F_1, F_2, \cdots, F_m$. Suppose g feature selection methods are used in our approach, and k important features are expected. The execution process of the proposed approach is described as follow:

Step 1. Evaluating the importance of each feature by means of g feature selection methods respectively, and sorting them in descending order of importance. The sequence of features generated by the *ith* feature selection method is denoted as $a_{i1}, a_{i2}, \ldots, a_{ik}, \ldots, a_{im}$. Then, a matrix about feature sequences from g feature selection methods is defined as:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2k} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ a_{g1} & a_{g2} & \cdots & a_{gk} & \cdots & a_{gm} \end{bmatrix}.$$

Step 2. Let B_h be a matrix which is composed of the first h columns of matrix A, i.e.,

$$B_h = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1h} \\ a_{21} & a_{22} & \cdots & a_{2h} \\ \vdots & \vdots & \ddots & \vdots \\ a_{g1} & a_{g2} & \cdots & a_{gh} \end{bmatrix}.$$

Suppose $\varphi(B_h)$ is the number of features selected from matrix B_h by majority voting algorithm.

Step 3. Find an integer l, such as

$$l = \arg\min_h(\varphi(B_h) = k).$$

Then, select elements from matrix B_l according to majority voting algorithm. The selected elements are the important features, which number is k.

In this paper, we integrate five feature selection methods into the proposed approach. The five algorithms are CFS, LS, ReliefF, χ^2 and UDFS, which are introduced in Sect. 2.

4 Experiments

4.1 Dataset

In this paper, we use the NSL-KDD dataset[1] as benchmark to verify the performance of the proposed approach. The NSL-KDD dataset is a modified version of the famous KDDCup99 dataset, which solved some inherent problems existing in the KDDCup99 dataset [12].

In our experiments, we used one training set (KDDTrain+20%) and two testing sets (KDDTest+ and KDDTest-21). KDDTrain+20% is a 20% subset of the full NSL-KDD training set. KDDTest+ is the full NSL-KDD testing set and KDDTest-21 is a subset of KDDTest+ which removes the records correctly classified with 21 learners. The features and attack types are described in Tables 1 and 2, respectively.

[1] http://www.unb.ca/cic/research/datasets/nsl.html.

Table 1. Description of features

Feature no.	Feature name	Feature no.	Feature name	Feature no.	Feature name
F1	Duration	F15	Su attempted	F29	Same srv rate
F2	Protocol type	F16	Num root	F30	Diff srv rate
F3	Service	F17	Num file creations	F31	Srv diff host rate
F4	Flag	F18	Num shells	F32	Dst host count
F5	Source bytes	F19	Num access files	F33	Dst host srv count
F6	Destination bytes	F20	Num outbound cmds	F34	Dst host same srv rate
F7	Land	F21	Is host login	F35	Dst host diff srv rate
F8	Wrong fragment	F22	Is guest login	F36	Dst host same srv port rate
F9	Urgent	F23	count	F37	Dst host srv diff host rate
F10	Hot	F24	Srv count	F38	Dst host serror rate
F11	Number failed logins	F25	Serror rate	F39	Dst host srvserror rate
F12	Loginned in	F26	Srvserror rate	F40	Dst host rerror rate
F13	Num compromised	F27	Rerror rate	F41	Dst host srvrerror rate
F14	Root shell	F28	Srvrerror rate	F42	Class label

Table 2. Description of attack types

Attack group	Attacks	Label
Normal		1
Dos	Back, Land, Neptune, Pod, Smurf, Teardrop, Mailbomb, Processtable, Udpstorm, Apache2, Worm	2
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint	3
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps	4
R2L	Guess_passwd, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Htttptunnel, Sendmail, Named	5

4.2 Evaluation Criteria

Confusion Matrix is a useful tool for comparing the outcome with ground truth (see Table 3). In Table 3, *TP* is the number of attacks that are correctly predicted; *TN* is the number of normal events that are correctly identified; *FP* is the number of normal connections that are wrongly predicted; *FN* is the number of attacks that are wrongly detected.

Table 3. Confusion matrix

		Predicted	
		Attacks	Normal
Actual	Attacks	*TP*	*FN*
	Normal	*FP*	*TN*

Four evaluation criteria used in our experiments are *Accuracy* (Acc), *Precision*, *False alarm rate* (FAR) and *Recall* [14, 15]. The calculations of these four criteria are defined at follows.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$FAR = \frac{FP}{TN + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

4.3 Experimental Results

In this section, we conduct two experiments to verify the performance of the proposed approach. The effectiveness of our feature selection approach is measured by combining the SVM. The corresponding detection models are listed in Sect. 1.

In the first experiment, we estimate the influence of the number of important features k to the accuracy of each detection model. In this experiment, we test six different k, i.e., 10, 15, 20, 25, 30 and 35. The results are shown in Table 4 (the best result for each column is highlighted in boldface). From Table 4, we can observe that the model of SVM+Our achieves five best accuracy out of six columns on KDDTest+. Only when k = 10, SVM+Our obtains the second-best result on KDDTest+. Similar results can be found on KDDTest-21. From the results in Table 4, we can conclude that our feature selection algorithm can choose more useful features than single feature selection method. In addition, we can find that SVM+Our gets the best accuracy on both testing set when the number of important features is 25. Thus, for our feature selection algorithm, we suggest to choose 25 important features.

Table 4. The accuracy of each model with different number of features (k)

Testing set	Model	Number of features					
		10	15	20	25	30	35
KDDTest+	SVM+CFS	43.08	67.46	69.87	72.09	73.88	76.02
	SVM+χ2	68.11	67.29	68.51	71.07	71.22	70.35
	SVM+LS	68.93	68.48	69.97	71.98	76.18	76.03
	SVM+ReliefF	**72.97**	71.64	70.76	71.79	70.72	68.48
	SVM+UDFS	43.08	64.18	70.24	72.05	73.39	74.25
	SVM+Our	70.62	**71.74**	**71.59**	**76.46**	**76.21**	**76.11**
KDDTest-21	SVM+CFS	18.16	40.16	43.42	47.18	50.42	54.20
	SVM+χ2	39.72	38.13	40.89	45.15	45.56	43.94
	SVM+LS	42.06	40.96	43.88	47.87	**54.73**	54.31
	SVM+ReliefF	**48.75**	46.33	44.51	46.46	44.38	40.45
	SVM+UDFS	18.16	33.97	43.78	47.24	49.76	51.11
	SVM+Our	45.06	**46.64**	**46.33**	**55.21**	54.65	**54.51**

Table 5. Comparison of different models in terms of four criteria

Testing set	Model	Acc	Precision	Recall	FAR
KDDTest+	SVM	67.31	44.62	97.19	42.23
	SVM+CFS	72.09	54.78	97.39	36.78
	SVM+χ2	71.07	51.74	97.97	38.55
	SVM+LS	71.98	59.95	91.08	34.89
	SVM+ReliefF	71.79	52.51	**98.73**	38.11
	SVM+UDFS	72.05	58.73	97.34	**33.11**
	SVM+Our	**76.46**	**61.41**	97.58	33.74
KDDTest-21	SVM	38.97	27.44	94.19	77.73
	SVM+CFS	47.18	39.39	95.76	73.52
	SVM+χ2	45.15	35.74	**96.16**	74.78
	SVM+LS	47.87	46.45	86.73	75.96
	SVM+ReliefF	46.46	36.73	94.76	74.11
	SVM+UDFS	47.24	43.21	96.09	71.71
	SVM+Our	**55.21**	**48.61**	95.94	**71.21**

The second experiment in this section is to compare the performance of each model in terms of four evaluation criteria depicted in Sect. 4.2. The results are illustrated in Table 5. In the experiment, number of important features is 25. From Table 5, we can see that the overall performance of SVM on both testing sets is the worst. That indicates that feature selection is necessary in intrusion detection. For Accuracy and Precision, SVM+Our achieves the best, and for Recall, the results of SVM+Our approximate the best. Additionally, SVM+Our obtains the best and second-best false alarm rates on KDDTest-21 and KDDTest+, respectively. The results in Table 5 indicate that our algorithm is more effective than individual feature selection method.

5 Conclusion

In this paper, a novel feature selection algorithm is proposed, which combining several feature selection methods and choosing important features by using majority voting rule. In this work, our algorithm integrates five individual feature selection methods. To verify the effectiveness of the proposed approach, we build different intrusion detection models by combing the SVM classification method with the proposed approach and the corresponding individual feature selection method. Experiments on NSL-KDD dataset show that our feature selection algorithm is more effective than single method.

References

1. Scarfone, K., Mell, P.: Guide to Intrusion Detection and Prevention Systems (IDPS). NIST special publication, vol. 800, p. 94 (2007)
2. Sangkatsanee, P., Wattanapongsakorn, N., Charnsripinyo, C.: Practical real-time intrusion detection using machine learning approaches. Comput. Commun. **34**, 2227–2235 (2011)
3. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning, vol. 103. Springer, Heidelberg (2013)
4. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
5. Nguyen, H.T., Petrović, S., Franke, K.: A comparison of feature-selection methods for intrusion detection. In: International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, pp. 242–255 (2010)
6. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: NIPS, vol. 189 (2005)
7. Kira, K., Rendell, L.A.: The feature selection problem: traditional methods and a new algorithm. In: AAAI, pp. 129–134 (1992)
8. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Mach. Learn. **53**, 23–69 (2003)
9. Liu, H., Setiono, R.: Chi2: feature selection and discretization of numeric attributes. In: The Seventh International Conference on Tools with Artificial Intelligence, pp. 388–391 (1995)
10. Yang, Y., Shen, H.T., Ma, Z., Huang, Z., Zhou, X.: L2, 1-norm regularized discriminative feature selection for unsupervised learning. In: IJCAI Proceedings of International Joint Conference on Artificial Intelligence, p. 1589 (2011)
11. Hosseini Bamakan, S.M., Wang, H., Yingjie, T., Shi, Y.: An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomputing **199**, 90–102 (2016)
12. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.: A detailed analysis of the KDD CUP 99 data set. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, pp. 1–6 (2009)
13. Hall, M.A.: Correlation-based feature selection of discrete and numeric class machine learning. In: ICML, pp. 359–366 (2000)
14. Lin, W.-C., Ke, S.-W., Tsai, C.-F.: CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl. Based Syst. **78**, 13–21 (2015)
15. Wang, G., Hao, J., Ma, J., Huang, L.: A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Syst. Appl. **37**, 6225–6232 (2010)