International Journal of Security and Its Applications
Vol. 9, No. 4 (2015), pp. 307-316
http://dx.doi.org/10.14257/ijsia.2015.9.4.28

A Wavelet Transform Based Support Vector Machine Ensemble Algorithm and Its Application in Network Intrusion Detection

Xuesen Cai [*1] and Fanhua Yu

College of Computer Science and Technology, Changchun Normal University, Changchun, China,130032
caixuesen_jl@126.com

Abstract

Traditional network intrusion detection algorithms are time consuming due to the existence of redundant attributes. In order to improve the efficiency of network intrusion detection, in this paper, we propose a wavelet transform based support vector machine ensemble algorithm. Firstly, we use wavelet transform to remove the redundant attributes from the original dataset. Then we train a support vector machine ensemble on the simplified dataset. As the wavelet transform in this algorithm can effectively remove the redundant attributes, the proposed algorithm is with high efficiency. Simulation experiments on KDD CUP 99 data set show that the proposed algorithm has good intrusion detection performance.

Keywords*: Intrusion detection, redundant attributes, wavelet transform, support vector machine ensemble*

1. Introduction

With the development of network science, people's lives become more convenient. However, at the same time, it also seriously affects people's privacy and property safety. Recent years, network security has aroused widespread concern, and has become a hot topic in computer research fields.

Network intrusion detection [1] is the discovery of the intrusion behavior. An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a Management Station. Network intrusion detection technology is the most important part in an intrusion detection system. As it can be regarded as a pattern recognition problem, most pattern recognition algorithms can be applied in this problem.

Li et al [2] proposed a fuzzy multi-class support vector machine algorithm, denoted as FMSVM. In the algorithm, a fuzzy membership function was introduced into the multi-class support vector machine. The membership function acquired different values for each input data according to their different affects on the classification result. Yu et al[3]presented an ensemble approach to intrusion detection based on improved multi-objective genetic algorithm. The algorithm generated the optimal feature subsets, which achieve the best trade-off between detection rate and false positive rate through an improved MOGA. And the most accurate and diverse base classifiers were selected to constitute the ensemble intrusion detection model by selective ensemble approach. Guo et al[4] employed random forests algorithm (RFA) in intrusion detection. They devised an improved variation of random forests algorithm (IRFA) and presented an IRFA based model for intrusion detection in information exchanged through network connections.

The above algorithms improve the performance of IDS to some extent, but they are time-consuming because of the redundant attributes in the network links data. Unlike the

[1] Corresponding author: Xuesen Cai

ISSN: 1738-9976 IJSIA

above algorithms, in this paper, we proposed a wavelet transform based support vector machine ensemble algorithm and applied the algorithm on intrusion detection. The algorithm firstly uses wavelet transform [5] to make dimension reduction and then trains support vector machine[6] ensemble classifiers on the simplified dataset. As the proposed algorithm can effectively reduce the dimension of the original dataset, it is with a high efficiency.

The rest of this paper is organized as follows. Section 2 introduces the theories of wavelet transform and support vector machine ensemble. Section 3 presents the wavelet transform based support vector machine ensemble algorithm. In Section 4, we apply the proposed algorithm in intrusion detection problems by using KDD CUP 99 dataset. Section 5 summaries the main contribution of this paper.

2. Related Theories

2.1. Wavelet Transform

The discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector $\boldsymbol{X}$, transforms it to a numerically different vector, $\boldsymbol{X}'$, of wavelet coefficients. The two vectors are of the same length. When applying this technique to data reduction, we consider each tuple as an n-dimensional data vector, that is, $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$, depicting n measurements made on the tuple from n database attributes[7].

The usefulness of wavelet transform lies in the fact that the wavelet transformed data can be truncated. A compressed approximation of the data can be retained by storing only a small fraction of the strongest of the wavelet coefficients. For example, all wavelet coefficients larger than some user-specified threshold can be retained. All other coefficients are set to 0.

The DWT is closely related to the discrete Fourier transform (DFT), a signal processing technique involving sines and cosines. In general, however, the DWT achieves better lossy compression. That is, if the same number of coefficients is retained for a DWT and a DFT of a given data vector, the DWT version will provide a more accurate approximation of the original data. Hence, for an equivalent approximation, the DWT requires less space than the DFT. Unlike the DFT, wavelets are quite localized in space, contributing to the conservation of local detail.

Popular wavelet transforms include the Haar-2, Daubechies-4, and Daubechies-6 transforms. The general procedure for applying a discrete wavelet transform uses a hierarchical pyramid algorithm that halves the data at each iteration, resulting in fast computational speed. The method is as follows:

1. The length, L, of the input data vector must be an integer power of 2. This condition can be met by padding the data vector with zeros as necessary ($L \geq n$).

2. Each transform involves applying two functions. The first applies some data smoothing, such as a sum or weighted average. The second performs a weighted difference, which acts to bring out the detailed features of the data.

3. The two functions are applied to pairs of data points in X, that is, to all pairs of measurements (x_{2i}, x_{2i+1}). This results in two sets of data of length L/2. In general, these represent a smoothed or low-frequency version of the input data and the high frequency content of it, respectively.

4. The two functions are recursively applied to the sets of data obtained in the previous loop, until the resulting data sets obtained are of length 2.

5. Selected values from the data sets obtained in the above iterations are designated the wavelet coefficients of the transformed data.

2.2. Support Vector Machines

(1) Linear SVMs

Let training samples be $(x_1, y_1),(x_2, y_2),\cdots,(x_l, y_l), x \in R^n, y \in \{+1,-1\}$. Suppose that the training set is linear separable, i.e., there exist a *N* dimensional *w* and a scaler *b*, which subject to the following inequality constraints

$$y_i(w \cdot x_i + b) \geq 1 \tag{1}$$

For linear separable problems, SVMs solve the following quadratic programming problem

$$\begin{aligned} &\min \quad \frac{1}{2}\|w\|^2 \\ &\text{s.t.} \quad y_i((w \cdot x_i) + b) \geq 1, \qquad i = 1,...,l \end{aligned} \tag{2}$$

aiming to find the maximum margin hyperplane *y*= *g*(*x*).

For approximate linear separable problem, slack variables, $\xi_i \geq 0, i = 1,...,l$, are introduced to allow some of the training points to be misclassified. Thus the exact classification constraints (1) are then replaced with

$$y_i((w \cdot x_i) + b) \geq 1 - \xi_i, \qquad i = 1,...,l \tag{3}$$

Our goal is now to maximize the margin while softly penalizing points that lie on the wrong side of the margin bound. We therefore minimize

$$\frac{1}{2}\|w\|^2 + C \cdot \sum_{i=1}^{l} \xi_i \tag{4}$$

where the parameter *C*>0 controls the balance between maximum margin hyperplane and minimum experience risk.

Using a Lagrangian, this optimization problem can be converted into a dual form

$$\begin{aligned} &\min_a \quad \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j (x_i \cdot x_j) a_i a_j - \sum_{i=1}^{l} a_i \\ &\text{s.t.} \quad \sum_{i=1}^{l} a_i y_i = 0 \\ &0 \leq a_i \leq C, \quad i = 1,...,l \end{aligned} \tag{5}$$

Once the Lagrange multipliers are determined, the normal vector w^* and the threshold b^* can be derived from the Lagrange multipliers:

$$\begin{aligned} w^* &= \sum_{i=1}^{l} a_i^* y_i x_i \\ b^* &= y_i - \sum_{i=1}^{l} a_i^* y_i (x_i \cdot x_j) \end{aligned} \tag{6}$$

Thus we have the decision function given by the following

$$f(x) = \text{sgn}(g(x)) \tag{7}$$

where $g(x) = (w^* \cdot x) + b^*$, and $\text{sgn}(\cdot)$ is a sign function.

(2) Non-linear SVMs

For non-linear separable problem, SVMs first map the low dimensional original space to high or even infinite dimensional feature space. Then, the solution of the quadratic programming problem will be performed in feature space with linear SVMs. The severe difficulty that can arise in spaces of many dimensions is sometimes called the curse of dimensionality[8]. In fact, kernel method is introduced to avoid it, and the dual representation can be represented as

$$\min_a \qquad \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j K(x_i \cdot x_j) a_i a_j - \sum_{i=1}^{l} a_i$$

$$\text{s.t.} \qquad \sum_{i=1}^{l} a_i y_i = 0$$

(8)

$$0 \le a_i \le C, \quad i = 1,...,l$$

Similar to linear SVMs, the decision function of non-linear SVMs is by the following

$$f(x) = \text{sgn}(g(x))$$

(9)

Where

$$g(x) = \sum_{i=1}^{l} a_i^* y_i K(x_i \cdot x) + y_j - \sum_{i=1}^{l} y_i a_i^* K(x_i, x_j)$$

(10)

(3) Kernel functions

For models which are based on a fixed nonlinear feature space mapping φ(x), the kernel function is given by the relation

$$K(x, x') = \phi(x)^T \phi(x')$$

(11)

From this definition, we see that the kernel is a symmetric function of its arguments so that $K(x, x') = K(x', x)$. The kernel concept was introduced into machine learning in the context of large margin classifiers by Boser et al. [9] giving rise to the technique of support vector machines. Since then, there has been considerable interest in this topic, both in terms of theory and applications[10].

A necessary and sufficient condition for a function $K(x, x')$ to be a valid kernel is that the Gram matrix K, whose elements are given by $k(x_n, x_m)$, should be positive semidefinite for all possible choices of the set $\{x_n\}$. Several common kernel function are given by the following

Polynomial kernel $K(x, x_i) = [(x, x_i) + 1]^d$;

Gaussian kernel $K(x, y) = \exp(\frac{-\|x - y\|^2}{2\sigma^2})$;

Sigmoid kernel $K(x, x_i) = S(v(x, x_i) + c)$.

2.3. Ensemble

Classification Ensemble combines a set of trained weak learner models and data on which these learners were trained. It can predict ensemble response for new data by aggregating predictions from its weak learners. It also stores data used for training and can compute resubstitution predictions. It can resume training if desired.

The most straightforward way of manipulating the training set is called Bagging. On each run Bagging presents the learning algorithm with a training set that consists of a sample of *m* training examples drawn randomly with replacement from the original training set of m items. Such a training set is called a bootstrap replicate of the original training set, and the technique is called boot strap aggregation[11]. Each bootstrap replicate contains, on the average, 63.2% of the original training set with several training examples appearing multiple times.

Another training set sampling method is to construct the training sets by leaving out disjoint subsets of the training data. For example the training set can be randomly divided into 10 disjoint subsets. Then 10 overlapping training sets can be constructed by dropping out a different one of these 10 subsets. This same procedure is employed to construct training sets for 10 fold cross validation so ensembles constructed in this way are sometimes called cross validated committees[12].

The third method for manipulating the training set is illustrated by the AdaBoost algorithm developed by Freund and Schapire[13]. Like Bagging AdaBoost manipulates the training examples to generate multiple hypotheses. AdaBoost maintains a set of weights over the training examples. In each iteration *l*, the learning algorithm is invoked to minimize the weighted error on the training set, and it returns a hypothesis. The weighted error is computed and applied to update the weights on the training examples. The effect of the change in weights is to place more weight on training examples that were misclassified and less weight on examples that were correctly classified.

In SVM ensemble, individual SVMs are aggregated to make a collective decision in several ways such as the majority voting, least-squares estimation-based weighting, and the double layer hierarchical combing. The training SVM ensemble can be conducted in the way of bagging or boosting. In bagging, each individual SVM is trained independently using the randomly chosen training samples via a boostrap technique. In boosting, each individual SV M is trained using the training samples chosen according to the sample's probability distribution that is updated in proportion to the error in the sample. SVM ensemble is essentially a type of cross-validation optimization of single SVM, having a more stable classification performance than other models [14].

3. The Wavelet Transform based Support Vector Machine Ensemble Algorithm

As there are many redundant attributes for the network link data, the traditional intrusion detection algorithms are time consuming. In order to improve the efficiency of intrusion detection algorithms, we proposed a wavelet transform based support vector machine ensemble algorithm, denoted as WTSVME. In WTSVME, firstly the dimensions of original attributes of network link data are reduced by wavelet transform. Then the support vector machine ensemble classifier is trained on the dataset simplified. The pseudo-code of the WTSVME algorithm is shown in Table 1.

Table 1. The Pseudo-code of the WTSVME Algorithm

```
Input: the original training set D;
Output: decision rule F
Process:
T=wavelet_transform(D);          /*using wavelet transform to obtain a
                                   simplified data set T */
SVME = Boosting(SVM, T)          /*using Boosting sampling to train a support vector
                                   machine ensemble on training set T*/
 F=SVME(T)                       /*obtaining the decision rule F by training support
                                   vector machine ensemble classifier on
                                   training set T*/
```

As the dataset simplified by wavelet transform removed the redundant attributes from the original training set effectively, so the efficiency of the WTSVME algorithm can be higher than the previous algorithms working on the original dataset. In the next section, we will perform an experiment on KDD CUP 99 dataset to test the effective of the proposed algorithm.

4. Experiments

4.1. Experimental Data

The intrusion detection dataset in this paper is kddcup_10_per, which comes from the data set of KDD CUP 99. The data set contains 494021 records. Each record has 41 attributes, among which, 34 attributes are continuous, and 7 attributes are discrete. The data set also contains a class attribute. There are 23 classes in all, among which Normal is normal network behavior, and the other 22 classes (Back, Neptune, Smurf and etc) are intrusion behaviors. In this experiment, we map the 23 classes to 5 types, namely, Normal, Dos, R2L, U2R and Probing. The distribution of different types is shown in Table 2.

Table 2. The Distribution of Different Types in kddcup_10_per

Type	*Data number*	*Percentage (%)*
Normal	97278	19.69
Dos	391458	79.24
Probing	4107	0.83
R2L	1126	0.23
U2R	52	0.01

As the kddcup_10_per dataset is very large, for convenience, we select a subset from kddcup_10_per dataset to perform the intrusion detection experiment. The sample numbers of training set and testing set are shown in Table 3.

Table 3. The Selected Samples for Training and Testing

Type	*Training set size*	*Testing set size*
Dos	6000	3000
Probing	2000	1000
R2L	500	250
U2R	30	20

4.2. Data Preprocessing

(1) Conversion from character attribute value to numerical value

There are four character attributes in the data set. Since SVM algorithm only accepts numerical vectors, the character attribute values need to be converted into numerical values[15].

(2) Data normalization

Since the value range of each attribute in original data set is different, the data need to be normalized. We would like to map the continuous attribute value to the range [0.0, 1.0] by computing

$$V = (v - \min(f_i)) \Big/ (\max(f_i) - \min(f_i)) \tag{12}$$

where V is the attribute value after normalization, v is the attribute value of original data, and $min(f_i)$, $max(f_i)$ are the minimum and maximum values of attribute f_i respectively.

4.3. Evaluation Indexes

In this paper, we use DR, FR and Time as the evaluation indexes of classification performance, where Time records the running time of the experimental algorithms. The meanings of DR and FR are by the following.

Detection Rate (DR) =the number of samples detected / the total number of the abnormal samples.

False Positive (FR) =the number of the misclassification normal samples / the number of normal samples.

4.4. Experimental Method and Result

In order to test the performance of WTSVME algorithm, we compare FMSVM algorithm proposed in[2] and WTSVME algorithm on the selected dataset. We select C-SVC[6] as the classification algorithm in both algorithms, where C is a penalty factor. Since radial basis function (RBF) has a good adaptability on non-linear, and high dimensional data set, this experiment selects Gaussian kernel as kernel function:

$$K(x,y)=\exp(\frac{-\|x-y\|^2}{2\sigma^2}) \tag{13}$$

where σ is a width parameter, "x" and "y" are n-dimensional vectors in the original feature space.

As this experiment is a multi-classification problem, so we select one-against-all (1-v-r) approach[16], which is to transform a c-class problem into c two-class problems, where one class is separated from the remaining ones. In this experiment, the best σ and C are obtained by 10-fold cross-validation[17].

The experimental platform is as follows: Intel Core2 Duo CPU T6500, 2.10GHz, 2.00GB RAM, Windows 7 OS. Five runs of 10-fold cross-validation are performed on each data set, and the average classification result is reported in Figure 1 and Figure 2, and the running time of both algorithms is shown in Table 4.

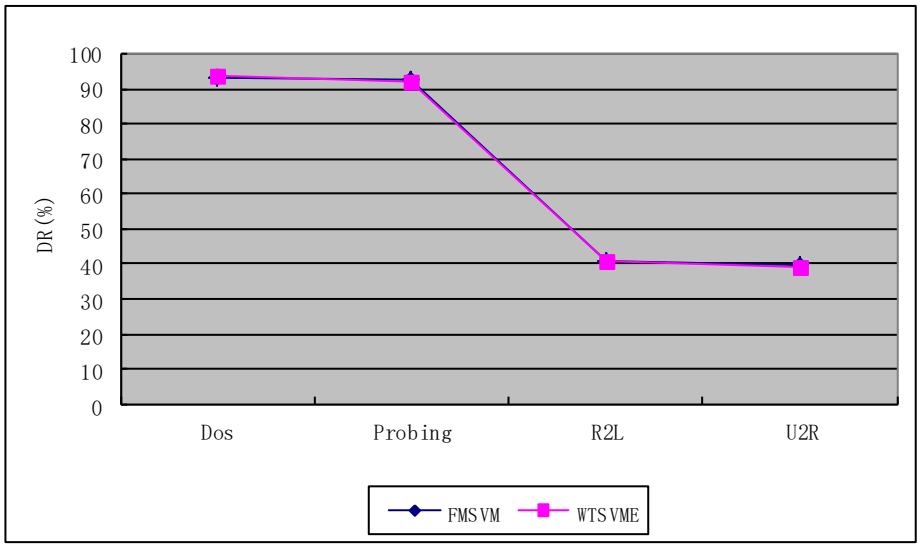

Figure 1. Detection Rates of FMSVM and WTSVME

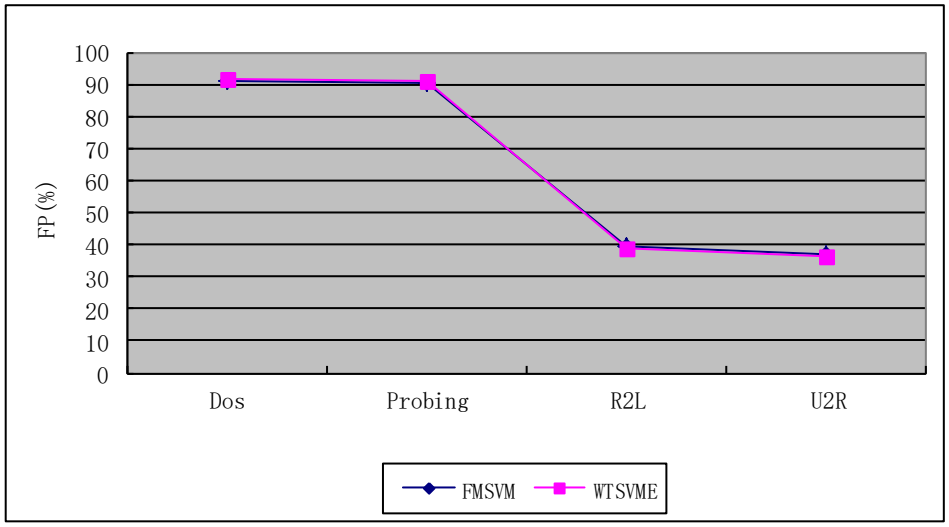

Figure 2. False Positive of FMSVM and WTSVME

Table 4. Runing time of FMSVM and WTSVME

Algorithm	*Time (s)*
FMSVM	722.1
WTSVME	596.3

4.5. Experimental Result Analysis

As shown in Figure 1 and Figure 2, the classification performance of WTSVME can match the performance of FMSVM algorithm, but from Table 4, we can see that the running time of WTSVME is much less than that of FMSVM algorithm. Therefore, WTSVME algorithm proposed in this paper exhibits a good way to network intrusion detection.

5. Conclusion

As network link data contain many redundant attributes, traditional intrusion detection algorithms are time consuming. In this paper, we proposed a wavelet transform based support vector machine ensemble algorithm. By removing redundant attributes effectively by wavelet transform, the efficiency of the proposed is higher than traditional algorithms. Experiment on KDD CUP 99 dataset shows that the proposed algorithm has a good intrusion detection performance.

Acknowledgements

This project is supported by the "Twelfth Five-Year Plan " science and technology project of the Education Department of Jilin province (Grant [2011] No. 360).

References

[1] X. Q. Wang, “Study on Genetic Algorithm Optimization for Support Vector Machine in Network Intrusion Detection”, AISS: Advances in Information Sciences and Service Sciences, vol. 4, no. 2, **(2012)**, pp. 282-288.

[2] K. L. Li, H. K. Huang and S. F. Tian, “Fuzzy multi-class support vector machine and application in intrusion detection”, Chinese Journal of Computers, vol. 28, no. 2, **(2005)**, pp. 274-280.

[3] Y. Yu and H. Hang, "An ensemble approach to intrusion detection based on improved multi-objective genetic algorithm", Journal of software, vol. 18, no. 6, **(2007)** , pp. 1369-1378.
[4] S. Q. Guo, C. Gao, J. Yao and L. Xie, "An Intrusion Detection Model Based on Improved Random Forests Algorithm", Journal of software, vol. 16, no. 8, **(2005)**, pp. 1490-1498.
[5] J. L. Wei and S. P. Zhai, "Image Fusion Method Based on Mean Square and Multi-wavelets", AISS: Advances in Information Sciences and Service Sciences, vol. 4, no. 2, **(2012)**, pp. 250- 257.
[6] H. M. Huang, H. S. Liu and G. P. Liu, "Face Recognition Using Pyramid Histogram of Oriented Gradients and SVM", AISS: Advances in Information Sciences and Service Sciences, vol. 4, no. 18, **(2012)**, pp. 1- 8.
[7] J. W. Han, M. Kamber and J. Pei, "Data Mining: Concepts and Techniques", 3rd edition, Morgan Kaufmann, **(2011).**
[8] R. Bellman, "Adaptive Control Processes: A Guided Tour", Princeton University Press, **(1961).**
[9] B. E. Boser, I. M. Guyon and V. N. Vapnik, "A training algorithm for optimal margin classifiers", D. Haussler (Ed.), Proceedings Fifth Annual Workshop on Computational Learning Theory.
[10] C. M. Bishop, "Pattern recognition and machine learning", Springer, **(2006)**.
[11] L. Breiman, "Bagging predictors", Machine learning, vol. 24, no. 2, **(1996)**, pp. 123-140.
[12] B. Parmanto, P. W. Munro and H. R. Doyle, "Improving committee diagnosis with resampling techniques", Advances in Neural Information Processing Systems, vol. 8, **(1996)**, pp. 882-888.
[13] Y. Freund and R. E. Schapire , "Experiments with a new boosting algorithm", ICML, vol. 96, **(1996)**, pp. 148-156.
[14] S. N. Pang, "SVM Aggregation: SVM, SVM Ensemble, SVM Classification Tree", IEEE SMC eNewsletter, **(2005)**.
[15] C. Y. Dong, "Study of support vector machines and its application in intrusion detection systems", PhD thesis, School of Electronic Engineering, Xidian University, **(2004).**
[16] C. W. Hsu and C. J. Lin, "A comparison on methods for multi-class support vector machines", IEEE Transactions on Neural Networks, vol. 13, no. 2, **(2001),** pp. 415-425.
[17] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection", Wermter S, Riloff E, Scheler G, eds. Proc. 14th Joint Int. Conf. Artificial Intelligence. San Mateo, CA: Morgan Kaufmann, **(1995)**.

Authors

Xuesen Cai, he was born in Taonan of Jilin Province, China, in 1976. He received Master degree from Jilin University, China, in 2008. Now he is a PhD candidate in College of GeoExploration Science and Technology of Jilin University, China. At the same time, he is working for department of Computer Science and Technology of Changchun Normal University as a lecturer. His research interests include Data mining and Virtual Instrument Development.