Intrusion Detection Classification Algorithm Based on Fuzzy SVM Model (Postprint)

Authors: Wang Sheng, Zhigang Jin

Date: 2018-12-13T00:00:00+00:00

Abstract

To address the issues of limited training samples and low classification accuracy in intrusion detection classification, a multi-level classification mechanism based on fuzzy support vector machines is proposed. This classification mechanism first trains a fuzzy SVM model to perform coarse-grained classification of data into normal and attack categories, then utilizes the DBSCAN algorithm to generate a fine-grained model for automatic clustering of attack subsets, thereby subdividing the relevant data to obtain specific fine-grained attack classes. In the mechanism design, the computation of the membership function is optimized, procedures for data standardization and normalization are devised, and an efficient classifier is trained. Experimental results indicate that for network traffic datasets prevalent in network intrusion detection data, which are characterized by outlier interference, high noise levels, and a large proportion of negative samples, the proposed algorithm maintains high classification accuracy while requiring relatively short computational time for the classification process.

Full Text

Preamble

IDS Classification Algorithm Based on Fuzzy SVM Models

Wang Sheng[1], Jin Zhigang[2] [1]. Northern Institute of Electronic Equipment of China, Beijing 100191, China [2]. School of Electronic & Information Engineering, Tianjin University, Tianjin 300072, China

Abstract: To address the challenges of limited training samples and low classification accuracy in intrusion detection, this paper proposes a multi-level classification mechanism based on fuzzy support vector machines (FSVM). The mechanism first employs a trained fuzzy SVM model to coarsely partition data

into normal and attack categories, then applies the DBSCAN algorithm to generate a refined subdivision model for automatic clustering of attack subsets, thereby subdividing relevant data into specific attack classes. In designing this mechanism, we optimized the membership function calculation, designed data standardization and normalization procedures, and trained an efficient classifier. Experiments demonstrate that for network service datasets characterized by frequent outlier interference, high noise levels, and a large proportion of negative samples commonly found in network intrusion detection data, the proposed algorithm achieves high classification accuracy while requiring relatively short computation time.

Keywords: fuzzy; SVM; IDS; classification

0 Introduction

The growing number of network users has exacerbated cybersecurity concerns. According to the 41st CNNIC report, by the end of 2017, the number of internet users in China approached 800 million, with many utilizing high-value applications such as online finance [1]. In response to cybersecurity threats including data breaches, online fraud, and denial-of-service attacks, passive defense and active detection technologies have become research hotspots.

Intrusion detection represents a crucial active cybersecurity measure and a key component of security defense, typically involving packet analysis and identification to enable appropriate defensive actions. As network behavior grows increasingly complex, flexible and efficient identification and classification algorithms are required for data analysis.

Support vector machine (SVM) models can be applied to state estimation and classification for network intrusion detection [2]. Reference [3] proposes a combined principal component analysis (PCA) and SVM mechanism for intrusion detection classification. This method first extracts key components from network data via PCA, then optimizes SVM parameters using particle swarm algorithms. While improving classification and attack prediction accuracy, the approach suffers from high complexity due to its use of three machine learning algorithms (PCA, particle swarm, and SVM), and its actual classification performance depends on expert experience in the PCA model. To enhance intrusion detection accuracy, reference [4] introduces a dual-SVM model. Although classification accuracy improves, computational speed decreases significantly, and the dual-SVM model becomes more sensitive to parameter selection and optimization, making real-time automatic parameter selection difficult.

To address data noise issues, some researchers have introduced fuzzy membership to reduce SVM model sensitivity to noise points [5]. However, this model only employs fuzzy processing with a single-level SVM, making it difficult to efficiently perform multi-class classification on attack data that represents a very

small proportion of cybersecurity datasets. To maintain classification accuracy while adapting to data noise, other studies have employed alternative machine learning models and algorithms, primarily including artificial immune models [6] and K-means clustering models. Meanwhile, research on multi-class models for intrusion detection has also been extensive, with existing methods mainly focusing on direct application of multi-class SVM models to original datasets, such as the TWSVM model using a partial binary tree multi-class algorithm [7], an adaptive threshold multi-class SVM model [8], and a fast multi-class sampling SVM model [9].

Analysis indicates that support vector machine models represent the mainstream approach in intrusion detection classification research. Building upon the basic SVM model and considering dataset characteristics and practical classification requirements, designing appropriate parameters and integrating additional methods can effectively facilitate intrusion detection training and classification.

Related research shows that SVM models are highly sensitive to kernel function selection and parameter tuning, causing parameter optimization in multi-class algorithms to become increasingly complex as the number of categories grows. For network intrusion detection, training data exhibits uneven category distribution, with normal samples far outnumbering attack samples. This makes direct training of single-level SVM on raw data difficult for model parameter optimization, resulting in SVM classifiers with significantly lower accuracy for attack sample classification compared to normal sample classification. Therefore, to ensure multi-class discrimination accuracy in intrusion detection, this paper designs a novel mechanism combining fuzzy SVM binary classification with multi-value regression fine-classification models. The approach first trains a fuzzy SVM for binary classification of the initial dataset, then applies a DBSCAN fine-classification model to the "attack data" for multi-class automatic clustering discrimination to determine specific attack categories.

1 Model Design

1.1 Overall Model Architecture

The model first preprocesses data packets, then uses a coarse classifier for discrimination. Data classified as "normal" requires no further processing. The "attack data" subset is classified into specific attack types using a fine classifier.

The intrusion detection classification model consists of two levels: the coarse classifier performs initial screening of raw data to extract data of interest, while the fine classifier trains on data initially identified as attacks for multi-class subdivision. As shown in Figure 1 [Figure 1: see original paper], during preprocessing, the raw data is duplicated. The first copy treats normal data as positive samples and attack data as negative samples, used solely for fuzzy SVM coarse classification. Normal data is removed to construct the second training set, which after training yields parameters for both the intrusion detection binary classification model and the regression model multi-class classification. These

coarse and fine classification training parameters are then unified and passed to the predictor for formal hierarchical multi-class discrimination.

1.2 Fuzzy Classification Model Construction

Traditional SVM models are susceptible to noise points and outliers, leading to unstable classification. Since intrusion detection datasets typically contain significant noise, SVM models cannot be applied directly. Adding fuzzy membership attributes to data points through fuzzy processing can effectively mitigate the impact of outliers and noise on classification by fuzzifying the penalty parameter in SVM using membership functions. An efficient coarse classification model for intrusion detection data is obtained through optimized SVM parameter training and membership function adjustment.

For a sample i with feature vector and label, we add a function representing its . The optimal classification for the extended fuzzy SVM model is equivalent to solving equation (1) [10]. ; The SVM kernel function is expressed as where: is the slack variable, C is a fixed constant. To achieve better optimization results, a smaller value can be selected to reduce impact on optimization outcomes.

Using Lagrangian relaxation methods, equation (1) is equivalent to solving the extremum of equation (2).

The binary classification problem for intrusion detection data based on fuzzy SVM models thus transforms into optimizing the fuzzy membership function through dataset training. The following specifically describes the process of optimizing the membership function through sample training to obtain coarse classification model parameters.

To obtain the optimized function , the training set labels are divided into two subsets: normal and attack. For the normal subset , is set to 1; for the attack subset , is set to -1.

Based on calculating the distance between sample i and subset centers, is computed. The centers of positive and negative samples are denoted as and respectively. For each sample i, distances to both positive and negative sample centers are calculated. Taking positive samples as an example, , and similarly for negative samples d-. The membership function is further defined as .

For all samples, the LIBSVM software is used to train and optimize the membership function , with the resulting model parameters applied to the classifier in Section 1.3.

1.3 Attack Data Multi-class Subdivision Based on DBSCAN

DBSCAN (Density-based spatial clustering of applications with noise) is a density-based clustering algorithm [11]. DBSCAN categorizes data points into three types based on density, determined by whether the number of points within a circular neighborhood exceeds *MinPts*. Points with no fewer than

MinPts points in their neighborhood are core points; points with fewer than *MinPts* points within radius *Eps* but falling within a core point' s neighborhood are border points; remaining points are classified as noise. The algorithm' s main steps are:

a) Create new cluster. For object p in the unprocessed subset, if p is unprocessed, examine its neighborhood. If the neighborhood contains no fewer than *MinPts* points, create new cluster C and add all points to candidate set N.

b) Update candidate set. For any unprocessed object q in candidate set N, examine its neighborhood. If it contains at least *MinPts* objects, add these objects to N; if q does not belong to any current cluster, add q to C.

c) Check if N is empty. If not empty, repeat step b).

d) Check if all objects have been labeled. If objects remain, return to step c); otherwise, processing ends.

In summary, the proposed classification model achieves multi-class automatic classification of attack data containing outliers and substantial noise with low overall dataset proportion through integrated application of fuzzy SVM-based binary classification followed by DBSCAN clustering on data subsets identified as attacks, further dividing attack data into different subcategories.

2 Two-level Intrusion Detection Algorithm Based on Fuzzy SVM

To further classify data already coarsely identified as attacks, a one-vs-rest multi-class classification model is employed by constructing the required number of classifiers for training. For the prediction classifier, when determining which attack category a network packet belongs to, the probability of the packet belonging to each category is calculated, with the category having the highest probability selected as the packet' s classification. Unlike direct DBSCAN classification alone, which only uses density information and yields unstable classification numbers affected by data source and quality [12], the two-level classification algorithm designed in this paper significantly reduces noise point impact through fuzzy SVM processing, enabling DBSCAN to automatically generate clusters and obtain stable category numbers. For the KDDCup99 dataset used in this paper, which can be divided into four categories, only four trainers need to be constructed. The basic trainer flow is shown in Figure 2 [Figure 2: see original paper].

The DBSCAN algorithm pseudocode is as follows:

```
Input: Data object set D, radius Eps, density threshold MinPts
Output: Clusters C

DBSCAN(D, Eps, MinPts)
```

```
begin
    init C=0; // Initialize cluster count to 0
    for each unvisited point p in D
        mark p as visited; // Mark p as visited
        N = getNeighbours(p, Eps);
        if sizeOf(N) < MinPts then
            mark p as Noise; // If sizeOf(N) < MinPts, mark p as noise
        else
            C = next cluster; // Create new cluster C
            ExpandCluster(p, N, C, Eps, MinPts);
        end if
    end for
```

The key step ExpandCluster algorithm pseudocode is as follows:

```
ExpandCluster(p, N, C, Eps, MinPts)
    add p to cluster C; // First add core point to C
    for each point p' in N
        mark p' as visited;
        N' = getNeighbours(p', Eps); // Check radius for all points in N
        if sizeOf(N') >= MinPts then
            N = N + N'; // If >= MinPts, expand N
        end if
        if p' is not member of any cluster
            add p' to cluster C; // Add p' to cluster C
        end if
    end for
end ExpandCluster
```

3 Experiments

3.1 Data Preprocessing

Raw data directly collected by network intrusion detection systems consists of binary data streams from the network, requiring protocol parsing and format conversion for classification. First, the libpcap library function parses binary streams into fields such as IP addresses, ports, characters, and hexadecimal values. The converted data contains discrete and continuous fields, with discrete fields further divided into discrete character and discrete numeric types. Different preprocessing methods are applied to relevant fields to transform them into processed data acceptable to SVM models with minimal noise and errors. The preprocessing flow includes three steps: discrete character data processing, data standardization and normalization, and data format transformation.

As shown in Figure 3 [Figure 3: see original paper], discrete character data processing converts character data into numeric data to facilitate distance calculations during DBSCAN algorithm training. Data standardization and normalization address issues where large disparities in original data value ranges

may cause "large numbers dominating small numbers," data processing overflow, or inconsistent weighting. Normalization eliminates the impact of measurement units on model training, making training results more dependent on data characteristics themselves, thereby improving clustering model parameter optimization and classification prediction accuracy. Data format conversion further transforms numerically processed and normalized data into LIBSVM-supported format for direct input and training of fuzzy SVM models. The LIBSVM format is widely used in common classification algorithms, generally adopting the format {label 1:(value)1 2:(value)2 ···i:(value)i ···n:(value)n}, where label is the category label, index i is the ith field number, and (value)i is the numeric value of the ith field.

The KDDCup99 dataset is used for intrusion detection system data.

3.1.1 Character Data Numericalization The KDDCup99 dataset contains 42 fields, with 41 being network packet feature attribute fields and one being the data record label. To avoid issues where character data cannot directly calculate distances and where excessively large or small data values affect mean and distance calculations, data fields are processed according to the methods in Table 1 .

Table 1 Quantizing of Chars

TCP Connection Basic Features (9)	TCP Connection Content Features (13)	Continuous Data
Protocol type replacement	Flag converted to decimal	Service uses port
Retain original values		

For example, for the protocol_type field with discrete values including 'TCP' , 'UDP' , 'ICMP' , etc., 'TCP' is represented by 11, 'UDP' by 12, and 'ICMP' by 20.

3.1.2 Normalization Processing Further standardization and normalization are applied to numerically processed data to normalize training set data with the same feature attributes. A data record with label in the training set is denoted as , where i represents the data record index and j represents the feature number. Standardization and normalization are performed by calculating the mean and variance of feature j. The data is processed into standard data using the mean and variance of the same feature:

The normalized processing formula is as follows:

3.1.3 Data Format Conversion The numerically processed and normalized dataset is converted to LIBSVM format for SVM processing. The KDDCup99

dataset used in this paper has five label types: NORMAL, DOS, R2L, U2R, and PROBING. NORMAL represents normal, non-aggressive packets; DOS represents denial-of-service attacks; R2L represents unauthorized access from remote hosts; U2R represents unauthorized local superuser privilege access; PROBING represents port monitoring or scanning attacks. For model training and prediction classification needs, two training sets are generated during data format conversion: the first set labels NORMAL data as -1 and other four data types as 0; the second set excludes NORMAL data while retaining other data types.

After the above numerical processing, standardization/normalization, and data format conversion, the practical dataset is obtained.

3.2 Experimental Design

To train and test the proposed attack classification algorithm, a data collection and security classification experimental environment was built (Figure 4 [Figure 4: see original paper]), with the KDD CUP99 dataset driving the IDS Informer attack simulation software to generate various network attack packets, while clients generate normal network traffic. To improve experimental efficiency, 20% of 500,000 data records (100,000 total training samples of normal and attack behaviors) were uniformly sampled from the KDD CUP99 collection for training. After training the coarse and fine two-level classifiers, real business packet captures obtained from the prototype experimental network were used for prediction and classification discrimination. Table 2 shows the category distribution of the training set and the test set categories obtained from the experimental network.

To verify the classification accuracy and time efficiency of the proposed algorithm, comparative experiments were first conducted using only the optimized DBSCAN classifier against Naive Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF) classifiers. As shown in Figure 5 [Figure 5: see original paper], DBSCAN achieves the highest accuracy at 85%, while other classifiers achieve approximately 70% accuracy.

In terms of training time, although Naive Bayes training time is far less than the other three classifiers, its classification accuracy is too low, and it is overly sensitive to data noise with unstable classification type numbers. DBSCAN training time is similar to Random Forest and Support Vector Machine classifiers, providing stable classification numbers and high accuracy.

Since attack data accounts for a very low proportion in real network data with significant noise, direct DBSCAN classification yields low efficiency and excessively long training times. Therefore, in further Experiment 2, using the training and test set distributions shown in Table 2, a fuzzy SVM binary classifier was first used as the coarse classifier, followed by DBSCAN fine classification on attack data. Experimental results are shown in Table 3 . The results demonstrate that the fuzzy SVM-DBSCAN combined two-level classifier improves accuracy by over 10% compared to using DBSCAN alone, with only approximately 30%

increase in model training time. The combined coarse-fine two-level classifier is more suitable for real network scenarios requiring timely classification of actual business data and coordinated intrusion detection responses.

The above experiments used ordinary computer configurations. If the algorithms are compiled and optimized to run on high-configuration dedicated servers with multi-threaded execution, training speed is expected to improve by 6-8 times, reducing training time to 0.2-0.25 seconds. Furthermore, migrating the algorithms to dedicated acceleration hardware could improve training speed by at least an order of magnitude, reducing training time to 20-25 milliseconds, which fully satisfies IDS system operational requirements.

Table 2 Datasets of Training and Testing

Category	Training Set	Test Set
normal		
probe		

Table 3 Results of Classifiers

Classifier	Classification Accuracy	Training Time (s)
Fuzzy SVM-DBSCAN Combined		
DBSCAN Only		

4 Conclusion

Based on analysis of intrusion detection and classification models, this paper designed a coarse-fine joint classification model for multi-outlier noisy data, using fuzzy SVM classification as the foundation and applying DBSCAN models for clustering-based fine discrimination on data coarsely classified as attacks. Experimental results demonstrate that under conditions of uneven training sample distribution and limited training sample sizes, the DBSCAN algorithm achieves higher accuracy and shorter training time than other multi-class classification models. The two-level intrusion detection algorithm combining fuzzy SVM and DBSCAN achieves high classification accuracy while maintaining fast training speed, making it suitable for intrusion detection scenarios with uneven distributions of normal and attack data.

References

[1] China Internet Network Information Center. 41st Statistical Report on Internet Development in China [EB/OL]. [2018-03-15] http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201803/t

[2] Xiao Min, Han Jijun, Xiao Debao, et al. Review of IDS research based on clustering[J]. Journal of Computer Applications, 2008, 28(s1): 34-38.

[3] Kotpalliwar M V, Wajgi R. Classification of attacks using support vector machine (SVM) on KDDCUP' 99 IDS database[C]//Proc of International Conference on Communication Systems and Network Technologies. Piscataway, NJ: IEEE Press, 2015: 1-5.

[4] Wang Hao, Hua Jixue, Fan Xiaoshi. A Dual-SVM model based IDS scheme[J]. Journal of Shandong University: Engineering Edition, 2013, 43(6): 53-56.

[5] Liu Jia, Fang Ning, Xie Yongjun, et al. Multi-scale feature-based fuzzy-support vector machine classification using radar range profiles[J]. IET Radar Sonar Navigation, 2015, 10(2): 370-378.

[6] Zhang Ling, Bai Zhongying, Luo Shoushan, et al. Integrated IDS model based on rough set and artificial immunization[J]. Journal of Communications, 2013, 34(9): 166-176.

[7] Xie Juanying, Zhang Bingquan, Wang Wanzi. Dual-SVM based partial binary classification algorithm[J]. Journal of Nanjing University: Natural Science Edition, 2011, 47(4): 354-363.

[8] Wan Shibiao, Mak Manwai, Kung Sunyuan. Adaptive thresholding for multi-label SVM classification with application to protein subcellular localization prediction[C]//Proc of IEEE International Conference on Acoustics. Piscataway, NJ: IEEE Press, 2013: 3547-3551.

[9] Chen Jingnan, Liu Chenlin. Fast multi-class sample reduction for speeding up support vector machines[C]//Proc of IEEE International Workshop on Machine Learning for Signal Processing. Piscataway, NJ: IEEE Press, 2011: 1-6.

[10] Li Shengtun, Chen C C. A regularized monotonic fuzzy support vector machine model for data mining with prior knowledge[J]. IEEE Trans on Fuzzy Systems, 2015, 23(5): 1713-1727.

[11] Wang Zhigang. Coastline detection algorithm of SAR image based on superpixel[D]. Dalian: Dalian Maritime University, 2017.

[12] Zhang Minling, Wu Lei. Lift: multi-label learning with label-specific features[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2015, 37(1): 107-120.

Note: Figure translations are in progress. See original paper for figures.