IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 16, 2019, accepted August 25, 2019, date of publication September 4, 2019, date of current version September 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939490

Network Security Situation Prediction Based on MR-SVM

JINGJING HU, DONGYAN MA, CHEN LIU, ZHIYU SHI, HUAIZHI YAN, AND CHANGZHEN HU
School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Jingjing Hu (hujingjing@bit.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800700, and in part by the National Natural Science Foundation of China under Grant 61772070.

ABSTRACT The support vector machine (SVM) is verified to be effective for predicting cyber security situations, however, the long training time of the prediction model is a drawback to its use. To address this, a cyber security situation prediction model based on MapReduce and the SVM is proposed. The base classifier for this model uses an SVM, and parameter optimization is performed by the Cuckoo Search (CS) to determine the optimal parameters of the SVM. Considering the problem of time cost when a data set is large, we choose to use MapReduce to perform distributed training on SVMs to improve training speed. ''Map'' is used to map distributed training network security situation data, and ''Reduce'' merges and sorts the prediction results. Experimental results show that the proposed prediction model has improved the accuracy and decreased the training time cost compared to the traditional model.

INDEX TERMS MapReduce, SVM, cuckoo search, network security situation prediction.

I. INTRODUCTION

Cyber security situation prediction plays a vital role in the field of network security. It can predict the network environment, improve the security of the network environment, and prevent impending network security incidents [1], [2]. However, there exist many network data attributes and huge amounts of data. Every day, massive amounts of data are generated, which poses a huge challenge to the use of algorithms [3], [4]. The amount of data increases the training time of the machine learning algorithm, and reduces its efficiency. The space-time cost of the algorithm has a profound impact on the establishment of the network security prediction model. On the premise of ensuring the accuracy, reducing the training time and improving the forecasting efficiency become an important issue of security situation prediction.

The support vector machine (SVM) classifies data sets via the VC dimension theory and the structural risk minimization theory based on statistical learning [5], [6]. SVMs exhibit good performance in processing high-dimensional numbers and small sample data sets [7], [8]. However, for data sets with large sample sizes, the SVM processing speed is slower than other machine learning algorithms, which is adverse for predicting network security situation with huge amounts of data [9]. Therefore, the parallelization method is proposed to train the SVM and this paper uses MapReduce to parallelize the SVMs. MapReduce passes the data fragment to the mapper function for parallel processing, and then uses the reduce function to obtain the final result.

In order to improve the prediction accuracy of the SVM, the Cuckoo Search (CS) algorithm is used to optimize the parameters [10], [11]. There are many other parameter optimization algorithms for SVM, including the grid search algorithm [12], [13] and the particle swarm optimization algorithm, [14] among others. Of the available algorithms, the CS has global convergence, can find the global optimal solution of parameters, has fewer control parameters, and has higher versatility and robustness.

The research contributions of this paper can be summarized as follows:

1) Use the SVM to classify the network dataset and establish a network security situation prediction model.
2) Use CS algorithm to optimize the parameters and improve the classification efficiency of the SVM.
3) Use MapReduce to parallelize the SVM model using parameter optimization of the CS algorithm to improve time efficiency.

In summary, this paper proposes a cyber space security situation prediction model based on MapReduce and SVM (MR-SVM). The MapReduce method is used to parallelize the SVMs, and the effectiveness and prediction accuracy

The associate editor coordinating the review of this manuscript and approving it for publication was Zheli Liu.

of the method are quantitatively analyzed by experimentation. Compared to the results of a traditional SVM network security situation prediction method, the proposed prediction model has improved the accuracy and decreased the training time cost. The model helps predict the occurrence of network security incidents and early warnings, thereby reducing the losses caused by network insecurity.

The main research content of this paper is divided into eight sections. In section 2, we introduce recent technologies related to network security. In section 3, we establish the prediction model and outline the steps of establishing the prediction model proposed in this paper. In section 4, we introduce MapReduce distributed training and describe the parallel method used in the paper. In section 5, we introduce SVM classification and the SVM classification algorithm used in this paper. In section 6, we describe the SVM parameter selection process and algorithm. In section 7, we analyze the experimental results, which describes the data set selected by the experiment and the experimental verification of the proposed MR-SVM network security situation prediction models. Finally, we conclude the paper in Section 8.

II. RELATED WORK

Network security has a long history of research and many sensing technologies have been developed. However, many attack technologies are emerging, including white box attacks, gray box attacks, black box attacks, and improvements to these attacks [15]–[18]. In addition, there are many models and emerging technologies that apply to network security [19]–[23]. Security situation prediction has become an effective method for protecting network security, and has previously been applied in sensor networks [24]–[30] and other mobile networks [31], [32]. Many algorithms have been applied to the prediction of network security situation, including artificial neural networks, clustering algorithms, association analysis, and SVM [33]–[35]. The artificial neural network inputs the network security data set, calculates the characteristics of the data set in the hidden layer, and finally obtains the category information in the output layer, but its training time and complexity are high. The clustering algorithm obtains category information by clustering the feature vector of the network security data set, but the initial node selection is not easy. Association analysis obtains a network security situation prediction model by analyzing the relationship between each data point in the network, but it requires certain rules to support. In the network security situation prediction, the space formed by the network security data set can use the SVM to find the classification hyperplane, thereby determining the network security baseline and judging the network security situation [36]–[39]. The generalization ability of SVM is strong, however, the parameters of the SVM need to be tuned, and as the amount of data increases, the training time increases rapidly. To strengthen the network protection strength, we propose a network security situation prediction model. We choose the CS to optimize the parameters of SVM, and uses MapReduce for parallel optimization.

III. BUILDING A PREDICTIVE MODEL

The network security situation prediction process based on MR-SVM algorithm is presented in the following steps:

1) Obtain a network security data set, select a training set and testing set;
2) Upload the data set to HDFS, schedule it by MapReduce, and parallelize the SVM;
3) The data stored in HDFS is used as the data set of the SVM. The RBF kernel function is selected in the SVM. Define the SVM parameter value interval and step size, and apply the CS combined with the ten-fold cross-validation method for parameter optimization;
4) Use the parameters obtained in the third step to determine the SVM cyber security situation prediction model and test the model;
5) Determine whether the prediction result satisfies the termination condition. If it is true, obtain an optimized SVM prediction model; otherwise, return to the third step to continue optimizing the model. The termination condition is that the model prediction accuracy reaches a predetermined threshold or the number of cycles exceeds a preset maximum number of cycles;
6) The parallel SVM is reduced to obtain the cyber security situation prediction model.

In supervised machine learning, data sets are usually divided into training sets and testing sets. The training sets are used to optimize model parameters, and a high-precision network security situation prediction model is obtained. Testing sets are required to check whether the prediction model has the promotion ability. In this model, n training sets are needed; n data sets are obtained by sampling data sets containing massive data and n SVM models are trained by n data sets. Finally, the SVM prediction results are reduced. This results in the final cyber security situation prediction model. When using the training set to train the SVM classifier, the SVM parameter optimization method uses the CS combined with ten-fold cross-validation to obtain the SVM classifier with high classification accuracy.

The selection and parallelization of the basic classifier are the core of the network security situation prediction. The proposed MR-SVM network security situation prediction model consists of two parts. The first uses MapReduce for data parallelization, and the second uses SVM to perform classification predictions. The core algorithms of the model and the reasons for the algorithm selection are detailed in algorithm 1.

IV. MAPREDUCE DISTRIBUTED TRAINING

MapReduce is a programming model for the parallel computing of large data sets. [40], [41] ''Map'' and ''Reduce'' are its main functions. It is implemented to specify a ''Map'' function to map a set of key-value pairs into a new set of key-value pairs, and to specify the concurrent ''Reduce'' function to ensure that each of the mapped key-value pairs shares the same key group.

Algorithm 1 MapReduce-SVM of Network Security Situation Prediction

Input: Network dataset
Output: MR-SVM model

1: Preprocessing the dataset;
2: Upload the dataset to HDFS and schedule it by MapReduce;
3: Use SVM for training the processed dataset, and use the cuckoo search algorithm to optimize the SVM parameters;
4: **while** accuracy not satisfied **do**
5: Training the SVM model
6: Using the reduce function for reduction;

1) Once a MapReduce program starts, MRAppMaster will start first. After the MRAppMaster starts, the number of required maptask instances is calculated according to the number and size of the network security data sets uploaded at this time, and then the corresponding number of maptask processes is then started.
2) After the maptask process starts, data processing is performed according to the given data slice range. The main flow is:
   a) Use the specified input format to get the RecordReader to read a data set;
   b) Pass the input data set to the customer-defined *map*() method, perform SVM training, and collect the KV pairs output by the *map*() method into the cache;
   c) The KV in the cache is sorted according to the K partition and then overflows to the disk file;
3) After MRAppMaster monitors all maptask process tasks, it starts the reducetask process and tells the reducetask process what range of data to process.
4) After the reducetask process starts, several maptask output result files are obtained according to the location of the data to be processed as notified by MRAppMaster, and then maptask output result files are re-merged and sorted locally. Then, according to the KV of the same key as a group, the "Reduce" method is called to predict the prediction result. The result is reduced, the result KV of the operation output is collected, and the customer-specified output format is called to output the result data to the external storage.

The task of MapReduce on the map side primarily carries out the training process of the SVM. The Map task obtains the input data set from the fragmentation of the data block partition in the distributed file system HDFS, and then reads the input data set with the mapper function and uses the data set as the training data set of the SVM. The SVM uses the data set read by the map task to train. The CS algorithm is used to optimize the parameters required by the SVM, and the optimal parameters are found and then brought into the SVM model.

The improved SVM is trained, and then the test set of the data set in the file system is read to perform model verification on the SVM. Finally, the labeled SVM test data set is obtained. The data in the final data set is used as the key, and the identifier is used as the value to form an output file for the key-value pair. The Map side first writes the key value pair to the buffer. In the buffer, the map sorts the output data by key value and divides the data according to the corresponding reduce task. After the buffer data reaches the threshold, the buffer data is overwritten to the disk. The mapper function is shown in Figure 1.

After the completion of each map task, the end of the Reduce task copies the output file of the corresponding map to the local disk of the reduce end. The copy process also has a buffer for storing the map output file. When the buffer reaches the threshold, the input file is written to the disk and the key values are combined for initial merging. When the map task is completed, the reduce side combines the key values of all the input files, and then processes them in the Reducer function. The data key value pair is input into the Reducer function, then the input data is combined according to the key value, The same value of the key value is added; if the value is greater than 0, the value corresponding to the key value is reset to $+1$. Otherwise, the flag value that corresponds to the corresponding key value is reset to -1. Finally, the output file is written into the distributed file system HDFS, and the prediction result of the final obtained test data set is compared with the actual identification of the test data set to obtain the classification prediction accuracy and time efficiency information of the distributed SVM. The distributed parallel SVM is applied to the classification model of the network security dataset to predict the network security situation. The reducer function is shown in Figure 2.

V. SVM CLASSIFICATION

First, the work flow chart of the network security situation classification SVM is introduced, as shown in Figure 3. $X_i, i \in [1, n)$ is the network data feature vector. $X_i, i \in [1, n)$ is turned into high-dimensional data by the kernel

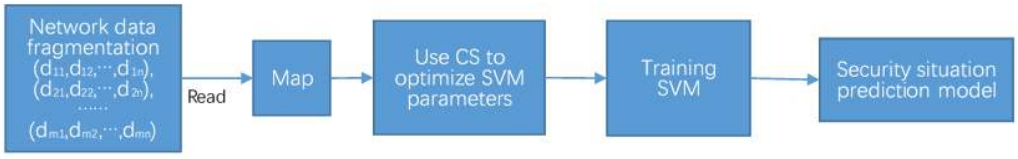


FIGURE 1. Mapper function of distributed SVM training.

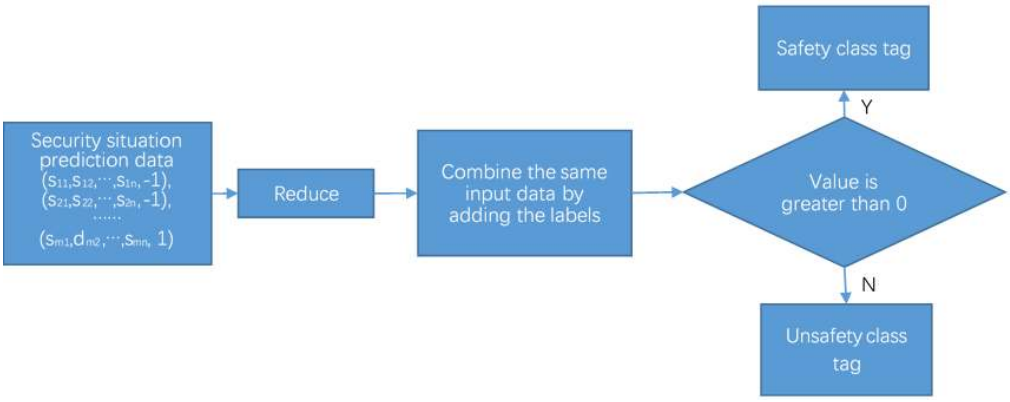


FIGURE 2. Reducer function of distributed SVM training.

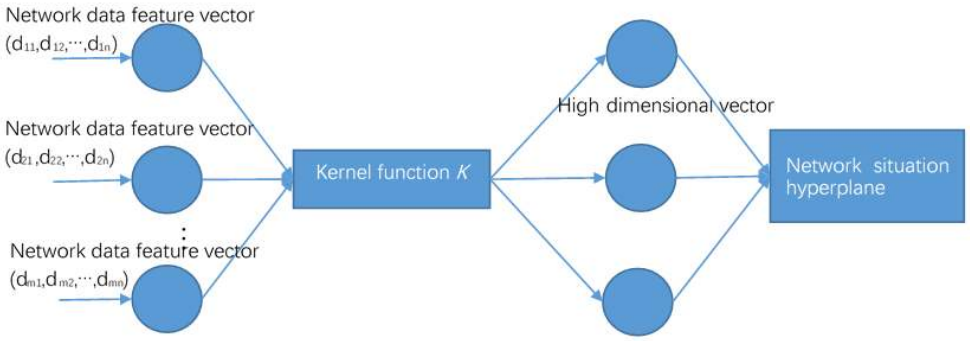


FIGURE 3. SVM work process of distributed SVM training.

function $K(x_i, x)$. In this paper, the kernel function $K(x_i, x)$ selects the radial basis kernel function and finally passes the decision function $sgn()$.

Using the network security training set to train the SVM, the two optimal parameters of the SVM are obtained, i.e., the classification hyperplane is determined, and the SVM model establishment process ends. The process of prediction is to input the test set into the trained SVM classifier and make a decision through the decision function. If the input data falls within the safe space determined by the optimal classification hyperplane function, the output result of the SVM classifier is marked as $+1$, i.e., the network connection corresponding to the data is determined to be safe. If the classification of the input data is in the unsafe space as determined by the hyperplane function, it is marked as -1 in the output result of the SVM classifier, i.e., the network connection corresponding to the data is determined to be unsafe.

The sample points of the network security data set usually have outliers. This situation is called approximate linear separability. If the original steps to search are followed, a classification hyperplane that can separate the two types of sample points cannot be found, and the problem cannot be solved. To solve the above problem, it is necessary to introduce slack variables in the classifier, as illustrated in Figure 4. $\xi_i(\xi_i \geq 0, i = 1, 2, \ldots, n)$ The slack variable is used to describe the outliers, and is a non-negative value. A penalty factor is introduced to evaluate this loss, indicating the degree of

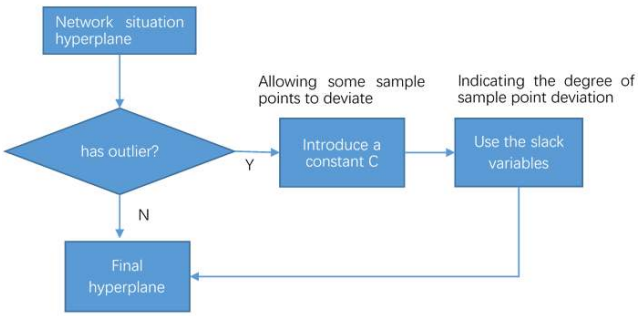


FIGURE 4. SVM that has outliers.

emphasis on outliers during training, and the outliers are also called loss points.

After introducing the penalty factor and the slack variable, the objective function is as shown in Eq. (1).

$$\Psi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{1}$$

For the actual data, the training process of the SVM is to solve the optimization problem of Eq. (2).

$$\begin{cases} min\{\frac{1}{2} \parallel \omega \parallel^2 + C \sum_{i=1}^{n} \xi_i\} \\ s.t. y_i(\omega^T \gamma(x_i) + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, n, \ \xi_i \geq 0 \end{cases} \tag{2}$$

,where C is the penalty coefficient, ξ_i is a relaxation variable, ω is a vector orthogonal to the classification hyperplane, b is a deviation term, and $\gamma(x)$ is a kernel function used by the SVM. In this model, $\gamma(x)$ is a radial basis kernel function. The result of solving the above optimization problem is the final decision function, as given by Eq.(3).

$$f(x) = sgn(\sum_{i=1}^{n} \lambda_i y_i K(x_i, x) + b) \tag{3}$$

The training process of the SVM is as follows:

1) Use the key-value pairs entered in the mapper function as the training set. Each key-value pair represents a piece of data in the training set, key represents the feature vector composed of each attribute of the data, and value represents the classification mark of the piece of data. The key value in the key-value pair constitutes the sample space of the SVM;
2) Find the optimal classification hyperplane in the input sample space. The optimal classification hyperplane distance is as far as possible from the positive and negative examples. That is, the classification hyperplane with the largest classification interval is found, and the optimization target is established according to the maximum interval principle. Simplify the operation by the Lagrangian method. Because the data set involved in this paper is linearly inseparable, the kernel method is used to map the sample space to the high-dimensional solution for optimization. The SVM of this paper uses the Gaussian kernel as the kernel function to enhance the generalization performance of the SVM. It introduces penalty factors and slack variables as two parameters of the SVM in the optimization problem;
3) Use the CS algorithm to optimize the two parameters in step 2, set the fitness function, initial population size, maximum iteration number, and maximum discovery probability as input to the CS algorithm. The fitness function is the classification accuracy of the SVM, so a set of SVM parameters obtained after reaching the maximum number of iterations can best fit the current SVM and improve its prediction accuracy.
4) Use the set of parameters obtained in step 3 to solve the SVM model, and obtain the current data set optimal SVM;
5) Using the SVM obtained in step 4 to predict the test set in the distributed file system HDFS, obtain the prediction identifier of each piece of data in the test set, and write the output file.

VI. SVM PARAMETER SELECTION

The Cuckoo Search (CS) is used for the SVM algorithm. It combines the ten-fold cross-validation method for parameter optimization, including penalty factor C and kernel function parameters g.

Normally, large cuckoos remove one or more eggs from a host before they lay their own eggs in the nest. In order not to be discovered by the host, the cuckoos ensure that the number of new eggs in the nest is equal or similar to the number that was removed. Once the cuckoo nestlings are hatched by the foster mother, the foster mother's own chicks are pushed out of the nest so that the foster nestlings are raised. This greatly increases the probability that the nestlings survive. In order to simulate the habit of the cuckoo, the CS algorithm assumes the following three ideal states:

1) Each cuckoo produces only one egg at a time, and randomly selects a nest to store;
2) During the nesting process, the best nest of eggs will be retained to the next generation;
3) The number of available nests is fixed, and the probability of finding foreign eggs in the nest is $P, P \in [0, 1]$. If a foreign bird is found, the owner of the bird re-establishes a bird's nest.

Under the assumptions of the above three ideal states, the updated formulas of the position and path of the CS are given by Eq.(4):

$$x_i^{t+1} = x_i^t + \alpha \oplus L(\lambda), \quad i = 1, 2, \ldots, n \tag{4}$$

In the equation, $x_i^{(t)}$ indicates the position of the $i-th$ bird's nest in the t-generation nest, $\oplus$ is point-to-point multiplication, and α is a step control amount that is used to control the search range of the step size, and its value obeys a normal distribution. Finally, $L(\lambda)$ is the Levi random search path, and the random step size is the Levi distribution.

The CS optimization range is $0.1 \sim 150$, the population size is 20, the maximum discovery probability P is 0.25, and the maximum iteration number is 20. The SVM classification accuracy rate is selected as the fitness function. The values of the parameters and the highest accuracy of the SVM are obtained, and the optimal classification hyperplane function is calculated. Finally, the SVM classifier is trained by the above steps. When the normalization operation is involved in the experiment, it is mapped to [0, 1].

The algorithm of the CS is described in Algorithm 2.

The experimental process of k-fold cross-validation is demonstrated in Algorithm 3.

The network security situation prediction model is verified by a ten-fold crossover method. The training set is divided into 10 subsets. Under the parameters determined by the CS algorithm, each subset is tested once. After 10 trials, 10 experiments are calculated to determine the average classification accuracy as a fitness function for evaluating the parameters of the group.

VII. EXPERIMENTAL SIMULATION AND RESULTS ANALYSIS

This section will verify the MR-SVM model. The KDD dataset is from an intrusion detection assessment project conducted by the US Department of Defense's Advanced Planning Agency (DARPA) at the MIT Lincoln Laboratory. The experimental data set was selected from the KDD dataset. Considering the characteristics of MR-SVM algorithms, the experiment is divided into the following two parts: (1) Parallelized SVM; the purpose of this part of the

Algorithm 2 Cuckoo Search Algorithm of MR-SVM

Input: the fitness function $f(X)$, the population size n, the maximum discovery probability P, the maximum iteration number MaxGeneration

Output: Optimal position X_i

1: Initialize the position of n nests, $X_i(i = 1, 2, 3, \ldots, n), X_i = (x_1, x_2)$;
2: **while** t<MaxGeneration||Minimum error requirement not met **do**
3: Select the correct rate of the SVM for the test set as the fitness function, and calculate the function value of each X_i;
4: Record the optimal function value and update other X_i;
5: Calculate the fitness function value for the updated X_i,compare it with the optimal function value, and if it is better, update the current optimal X_i;
6: Compare random numbers $r \in [0, 1]$ with P, if $r >$P, Then randomly change X_i^{t+1}, else stay X_i^{t+1}.Finally retain the best set of bird nest locations X_i;

Algorithm 3 k-Fold Cross-Validation of MR-SVM

Input: Network dataset

Output: Average of the accuracy of k prediction models

1: Divide the dataset into k subsets;
2: **while** model < k **do**
3: Select one of the subsets as the test set in sequence, and the remaining $k - 1$ subsets as the training sets;
4: Training to obtain network security situation prediction model;

experiment is to optimize the parameters for the two key parameters existing in the MR-SVM algorithm, train the SVM and perform reduction, and find the optimal network security situation prediction model under the experimental data set. (2) The purpose of the second part of the experiment is verifying the feasibility of the model. Verification is conducted by comparing the prediction results of the MR-SVM model with the prediction results of the traditional SVM model.

A. EXPERIMENTAL DATA SET

The experimental data uses the KDD (Data Mining and Knowledge Discovery) data set of an intrusion detection assessment project conducted by MIT Lincoln Laboratory [42]. It contains two data sets. A detailed description of the data characteristics is provided in Table 1. This experiment selects the ICMP protocol data in the KDD data set for testing.

In order to verify the performance of the proposed model, a 10-fold cross-validation (10-fold CV) was used in the experiment. First, the training set was randomly divided into 10 subsets. For each experiment, one of the subsets was used as the test set, and the remaining nine subsets were used as the training set. Each subset was used once as a test set, so a total of 10 validation experiments was performed. After the completion of 10 experiments, the average of 10 experiments of each indicator was used to evaluate the performance of the model.

This paper uses four indicators to evaluate the performance of the model, namely accuracy, precision, recall, and F-measure values (harmonic average of precision and recall) [43]. The recall rate indicates the ratio of the number of unsafe connections detected to the actual number of connections. The precision indicates the ratio of the number of unsafe connections to the actual number of connections. The F value is the harmonic average of the precision and recall, that has achieved a compromise between recall and precision. SVM parameter optimization is achieved with a CS combined with ten-fold cross-validation.

B. MODEL VALIDATION EXPERIMENT

C. EXPERIMENTAL RESULTS AND ANALYSIS

Based on the above selection parameters, the experimental results are the best prediction results of the MR-SVM model. The different SVM models are compared against the four evaluation indicators. LTSA is a nonlinear manifold learning method for the dimensionality reduction of high-dimensional data, and can maintain the inherent features and structure of the original data, which can improve the classification

TABLE 1. KDD dataset.

Basic feature	Specific feature	Time-based feature	Host-based feature
duration	hot	count	dst_host_count
protocol_type	num_failed_logins	srv_count	dst_host_srv_count
service	logged_in	serror_rate	dst_host_same_srv_rate
flag	num_compromised	srv_serror_rate	dst_host_diff_srv_rate
src_bytes	root_shell	rerror_rate	dst_host_same_src_port_rate
dst_bytes	su_attempted	srv_rerror_rate	dst_host_srv_diff_host_rate
land	num_root	same_srv_rate	dst_host_serror_rate
wrong_fragment	name_file_creations	diff_srv_rate	dst_host_srv_serror_rate
urgent	num_shells	srv_diff_host_rate	dst_host_rerror_rate
	name_access_files		dst_host_srv_rerror_rate
	name_outbound_cmds		
	is_hot_login		
	is_guest_login		

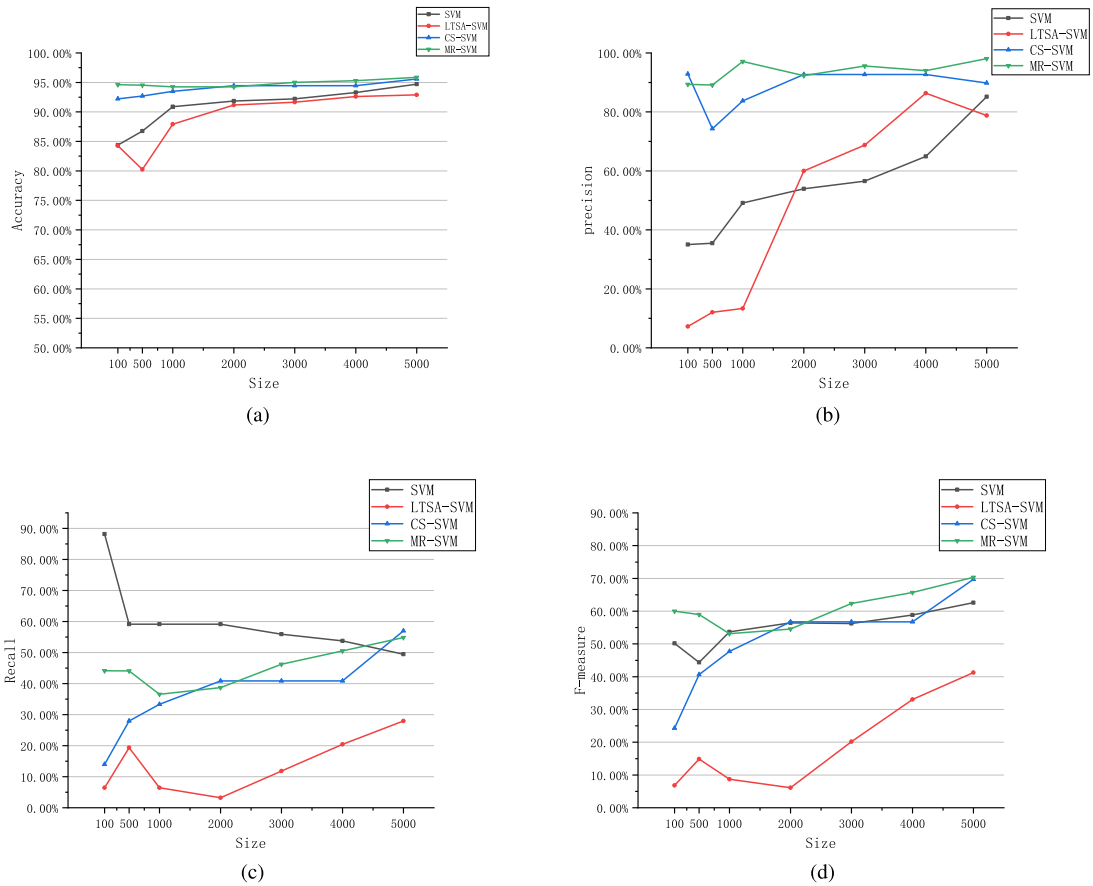


FIGURE 5. (a)The comparison of accuracy. (b)The comparison of precision. (c)The comparison of recall (d)The comparison of F-measure.

TABLE 2. The prediction results of SVM, CS-SVM, LTSA-SVM, and MR-SVM.

Algorithm	Accuracy	Precision	Recall	F-measure
SVM	94.73%	86.54%	48.39%	62.07%
LTSA-SVM	95.01%	80.60%	58.06%	67.5%
CS-SVM	95.97%	90.48%	61.29%	73.08%
MR-SVM	96.16%	88.41%	65.69%	74.31%

TABLE 3. The cost of time.

Algorithm	Time (ms)
SVM	1037
LTSA-SVM	54621
CS_ SVM	11078932
MR-SVM	1655501

efficiency of SVMs. The prediction results are provided in Table 2.

Table 3 provides a comparison of the time efficiency between CS_SVM and MR-SVM. The following Figure 5(a)-(d) describe the performance of SVM, CS-SVM, LTSA-SVM and MR-SVM under different size of data sets.

It is evident from the above figures that the SVM model using the CS algorithm is superior to the traditional SVM model in terms of all four indicators. Because LTSA is related to the dimension, the different data sets are chosen to have different dimensions, so the training results are not stable. The experimental results illustrate two aspects: on the one hand, when other conditions are the same, the parallelization of the SVM is more efficient than the traditional SVM model; on the other hand, the CS algorithm can be suitable to address the SVM parameter optimization problem, as it improves the four indicators.

The algorithm proposed in this paper can solve the problem of the binary classification of data sets with more data. The MapReduce method reduces the training cost of SVM by parallelizing the SVMs. The speedup of the model is 6.69 which is the ratio of time spent in traditional SVM and MR-SVM. The CS algorithm solves the problem of parameter optimization of the SVM, as it can find the optimal solution for this global problem.

VIII. CONCLUSION

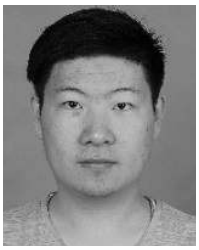
To address the problem of increasing training time costs resulting from a huge amount of network security data, this paper applies the MapReduce framework to network security situation prediction and uses the CS to optimize SVM parameters. This paper proposes an SVM network security situation prediction model MR-SVM. The model effectively reduces the training time of the SVM and improves the accuracy of network security situation prediction. Using the CS algorithm proposed in this paper, the prediction accuracy of the MR-SVM network security situation prediction model

increases. This paper selects the KDD data set for comparison experiments between the traditional SVM, the SVM using the CS algorithm for parameter optimization, the LTSA-SVM, and the MR-SVM. The experiments prove that the MR-SVM model can effectively solve the problem of the rapid increase of SVM training cost associated with increases of data volume. However, there are two nodes used in the experiment, and the training speed is limited. In the future work, multiple nodes can be established for training.

REFERENCES

[1] Z. Fan, Y. Xiao, A. Nayak, and C. Tan, "An improved network security situation assessment approach in software defined networks," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 2, pp. 295–309, Mar. 2019.
[2] Z. Guan, Y. Zhang, L. Wu, J. Wu, J. Li, J. Ma, and J. Hu, "APPA: An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *J. Netw. Comput. Appl.*, vol. 125, pp. 82–92, Jan. 2019.
[3] Y. Li, J. Hu, Z. Wu, C. Liu, F. Peng, and Y. Zhang, "Research on QoS service composition based on coevolutionary genetic algorithm," *Soft Comput.*, vol. 22, no. 23, pp. 7865–7874, 2018.
[4] J. Xiao, B. Zhang, and F. Luo, "Distribution network security situation awareness method based on security distance," *IEEE Access*, vol. 7, pp. 37855–37864, 2019.
[5] S. Ding, L. Cong, Q. Hu, H. Jia, and Z. Shi, "A multiway p-spectral clustering algorithm," *Knowl-Based Syst.*, vol. 164, pp. 371–377, Jan. 2019.
[6] J. He, Z. Zhang, M. Li, L. Zhu, and J. Hu, "Provable data integrity of cloud storage service with enhanced security in the Internet of Things," *IEEE Access*, vol. 7, pp. 6226–6239, 2018.
[7] S. Ding, N. Zhang, X. Zhang, and F. Wu, "Twin support vector machine: Theory, algorithm and applications," *Neural Comput. Appl.*, vol. 28, no. 11, pp. 3119–3130, Nov. 2017.
[8] H. Jingjing, C. Xiaolei, and Z. Changyou, "Proactive service selection based on acquaintance model and LS-SVM," *Neurocomputing*, vol. 211, pp. 60–65, Oct. 2016.
[9] X. Li, Y. Lu, S. Liu, and W. Nie, "Network security situation assessment method based on Markov game model," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 5, pp. 2414–2428, May 2018.
[10] S. Ding, Z. Zhu, and X. Zhang, "An overview on semi-supervised support vector machine," *Neural Comput. Appl.*, vol. 28, no. 5, pp. 969–978, May 2017.
[11] H. Zhu, X. Qi, F. Chen, L. Chen, and Z. Zhang, "Quantum-inspired cuckoo co-search algorithm for no-wait flow shop scheduling," *Appl. Intell.*, vol. 49, no. 2, pp. 791–803, Feb. 2019.
[12] P. C. Bhat, H. B. Prosper, S. Sekmen, and C. Stewart, "Optimizing event selection with the random grid search," *Comput. Phys. Commun.*, vol. 228, pp. 245–257, Jul. 2018.
[13] X. Kong, Y. Sun, R. Su, and X. Shi, "Real-time eutrophication status evaluation of coastal waters using support vector machine with grid search algorithm," *Mar. Pollut. Bull.*, vol. 119, no. 1, pp. 307–319, Jun. 2017.
[14] J. Vijayashree and H. P. Sultana, "A machine learning framework for feature selection in heart disease classification using improved particle swarm optimization with support vector machine classifier," *Program. Comput. Softw.*, vol. 44, no. 6, pp. 388–397, Nov. 2018.
[15] X. Gao, Y.-A. Tan, H. Jiang, Q. Zhang, and X. Kuang, "Boosting targeted black-box attacks via ensemble substitute training and linear augmentation," *Appl. Sci.*, vol. 9, no. 11, p. 2286, 2019.
[16] Q. Zhang, H. Gong, X. Zhang, C. Liang, and Y.-A. Tan, "A sensitive network jitter measurement for covert timing channels over interactive traffic," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3493–3509, Feb. 2019.
[17] X. Zhang, L. Zhu, X. Wang, C. Zhang, H. Zhu, and Y.-A. Tan, "A packet-reordering covert channel over VoLTE voice and video traffics," *J. Netw. Comput. Appl.*, vol. 126, pp. 29–38, Jan. 2019.
[18] Z. Qikun, L. Yongjiao, G. Yong, Z. Chuanyang, L. Xiangyang, and Z. Jun, "Group key agreement protocol based on privacy protection and attribute authentication," *IEEE Access*, vol. 7, pp. 87085–87096, 2019.
[19] Y. Li, S. Yao, K. Yang, Y.-A. Tan, and Q. Zhang, "A High-imperceptibility and histogram-shifting data hiding scheme for JPEG images," *IEEE Access*, vol. 7, pp. 73573–73582, 2019.
[20] Y. Xue, Y.-A. Tan, C. Liang, Y. Li, J. Zheng, and Q. Zhang, "RootAgency: A digital signature-based root privilege management agency for cloud terminal devices," *Inf. Sci.*, vol. 444, pp. 36–50, May 2018.
[21] Y.-A. Tan, Y. Xue, C. Liang, J. Zheng, Q. Zhang, J. Zheng, and Y. Li, "A root privilege management scheme with revocable authorization for Android devices," *J. Netw. Comput. Appl.*, vol. 107, no. 4, pp. 69–82, Apr. 2018.
[22] Z. Guan, Y. Zhang, L. Zhu, L. Wu, and S. Yu, "EFFECT: An efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid," *Sci. China Inf. Sci.*, vol. 62, no. 3, pp. 1–14, Mar. 2019.
[23] Z. Qikun, G. Yong, Z. Quanxin, W. Ruifang, and T. Yu-An, "A dynamic and cross-domain authentication asymmetric group key agreement in telemedicine application," *IEEE Access*, vol. 6, pp. 24064–24074, 2018.
[24] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 11–12, pp. 2314–2341, Sep. 2007.
[25] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Netw.*, vol. 5, no. 1, pp. 24–34, Jan. 2007.
[26] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.
[27] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "ASAP: An anonymous smart-parking and payment scheme in vehicular networks," *IEEE Trans. Depend. Sec. Comput.*, to be published.
[28] L. Zhu, X. Tang, M. Shen, X. Du, and M. Guizani, "Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 628–643, Mar. 2018.
[29] L. Zhu, Y. Wu, K. Gai, and K.-K. R. Choo, "Controllable and trustworthy blockchain-based cloud data management," *Future Gener. Comput. Syst.*, vol. 91, pp. 527–535, Feb. 2019.
[30] L. Zhu, C. Zhang, X. Xu, X. Du, R. Xu, K. Sharif, and M. Guizani, "Prif: A privacy-preserving interest-based forwarding scheme for social Internet of vehicles," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2457–2466, Aug. 2018.
[31] X. Zhang, C. Liang, Q. Zhang, Y. Li, J. Zhen, and Y.-A. Tan, "Building covert timing channels by packet rearrangement over mobile networks," *Inf. Sci.*, vols. 445–446, pp. 66–78, Jun. 2018.
[32] C. Liang, X. Wang, X. Zhang, Y. Zhang, K. Sharif, and Y.-A. Tan, "A payload-dependent packet rearranging covert channel for mobile VoIP traffic," *Inf. Sci.*, vol. 465, pp. 162–173, Oct. 2018.
[33] X. Du and H.-H. Chen, "Security in wireless sensor networks," *IEEE Wireless Commun. Mag.*, vol. 15, no. 4, pp. 60–66, Aug. 2008.
[34] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, and Y. Ma, "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 82–88, Jul. 2018.
[35] Q. Zhang, Y. Li, Q. Zhang, J. Yuan, R. Wang, Y. Gan, and Y. Tan, "A self-certified cross-cluster asymmetric group key agreement for wireless sensor networks," *Chin. J. Electron.*, vol. 28, no. 2, pp. 280–287, Mar. 2019.
[36] H. Hu, H. Zhang, Y. Liu, and Y. Wang, "Quantitative method for network security situation based on attack prediction," *Secur. Commun. Netw.*, vol. 2017, Jul. 2017, Art. no. 3407642.
[37] D. Zhao and J. Liu, "Study on network security situation awareness based on particle swarm optimization algorithm," *Comput. Ind. Eng.*, vol. 125, pp. 764–775, Nov. 2018.
[38] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu, "Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 940–953, Apr. 2018.
[39] C. Zhang, L. Zhu, C. Xu, X. Liu, and K. Sharif, "Reliable and privacy-preserving truth discovery for mobile crowdsensing systems," *IEEE Trans. Depend. Sec. Comput.*, to be published.
[40] Y. Madani, M. Erritali, and J. Bengourram, "Sentiment analysis using semantic similarity and Hadoop MapReduce," *Knowl. Inf. Syst.*, vol. 59, no. 2, pp. 413–436, May 2019.
[41] M. Bendre and R. Manthalkar, "Time series decomposition and predictive analytics using MapReduce framework," *Expert Syst. Appl.*, vol. 116, pp. 108–120, Feb. 2019.
[42] [Online]. Available: https://download.csdn.net/download/xiqianwei7030/10389510
[43] S. E. N. Fernandes and J. P. Papa, "Improving optimum-path forest learning using bag-of-classifiers and confidence measures," *Pattern. Anal. Appl.*, vol. 22, no. 2, pp. 703–716, May 2019.

JINGJING HU received the Ph.D. degree in computer science from the Beijing Institute of Technology, Beijing, China, where she is currently an Associate Professor with the School of Computer. Her research interests include in the areas of service computing, web intelligence, and information security.



DONGYAN MA is currently pursuing the Postgraduate degree with the School of Computer, Beijing Institute of Technology, China. His research interests include intelligent information networks and cyberspace security.

CHEN LIU is currently pursuing the Postgraduate degree with the School of Computer, Beijing Institute of Technology, China. His research interests include intelligent information networks, services computing, and information security.

ZHIYU SHI is currently a Professor with the School of Computer Science and Technology, Beijing Institute of Technology. He is also selected into the Program for New Century Excellent Talents in University from Ministry of Education, China. His research interests include the Internet of things, cloud computing security, and blockchain.

HUAIZHI YAN received the Ph.D. degree from the Beijing Institute of Technology, where he is currently an Associate Professor with the School of Computer. His current research interests include cyberspace security and software engineering.

CHANGZHEN HU received the Ph.D. degree from the Beijing Institute of Technology, where he is currently a Professor with the School of Computer. His current research interests include cyberspace security and intelligent security.