An improved bayesian network intrusion detection algorithm based on deep learning

Dong Zhao LIU,Zu Hong FENG

School of computer science and Engineering, North Minzu University, Yinchuan, Ningxia 750021, China
452002125@qq.com

Abstract. Based on the preprocessing of training data sets, this paper adopts the convolution neural network technology to realize the reduction target of the redundant attribute of the data set. An improved Bayesian network intrusion detection algorithm based on deep learning is proposed. In this algorithm, sliding window technique and relative Euclidean distance are defined, and the structure updating and parameter learning of Bayesian networks are adaptively carried out on the basis of computing mutual information between attributes. Experimental results show that the new algorithm can effectively improve the computational efficiency and the accuracy of intrusion detection.

1. Introduction

With the popularization of computer and various intelligent terminal devices, network traffic is increasing and network security problems emerge. How to improve the security performance of the network, enhanced monitoring of network traffic and the identification of abnormal traffic have become a problem for people to deal with. At present some of the intrusion detection technology can more accurately identify the known attacks, but when relevance between testing data set and training data set go down or appear a large number of unknown attacks, precision fell sharply and the result is not ideal. Therefore, how to establish effective, adaptive and expansible intrusion detection model is an important research topic in intrusion detection field[1]. This paper put forward a new kind of improved bayes network (Improved bayes network,IBN) intrusion detection algorithm DLIBN based on deep learning (Deep Learning,DL) algorithms. This paper introduces the widely used deep learning and the classical sliding window method, which effectively reduces the dimensionality of the data set, improves the network scalability, and improves the detection efficiency and accuracy.

2. Deep learning

The development of deep learning stems from the development of biological neural networks [2][3]. Through machine learning algorithms, machine devices can learn from the mass of data to the internal links, so as to classify or identify. Machine learning is modeled and classified by abstracting features extracted from large amounts of data. Therefore, the accuracy of the final classification is largely determined by the choice of features. Traditional artificial feature selection is time-consuming and laborious, and it can not guarantee the quality of feature selection. Deep learning [4] can automatically learn the feature expression from a large amount of data, and the efficiency and quality are much higher than manual selection.

Convolution neural network (CNN) [5] is a kind of deep learning, which belongs to feed-forward
deep network model. The convolution neural network consists of multiple single-layer convolution neural networks, mainly consisting of convolution layer, nonlinear transform and subsampled layer. CNN adopts local connection and weight sharing, each layer has two two-dimensional planes, and each two-dimensional plane has many independent neurons. The neurons in each two-dimensional plane are connected only to some neurons in the upper layer and are responsible for extracting local features. Each convolution layer and the subsampled layer has a plurality of two-dimensional sampling feature plane, plane characteristics of each layer are sharing the weight parameter, characteristic plane of different extracting specific characteristics, enhance the expression and characteristics in greatly reducing the amount of calculation and improve the network generalization ability.

![Figure 1 CNN model](image)

As shown in figure 1. In the convolution layer, the characteristic image of the upper layer is convolution with the given convolution kernel, and the result is output after the activation function, and becomes the input of the next layer. In one layer, different features can be extracted by using different convolution kernels, and good feature expression can be obtained. The convolution layer and the subsampled layer appear alternately, Generally, the convolution layer is calculated as:

\[
X^l_j = \sigma(\sum_{m_j} X^{l-1} \ast Kernel^l_j + b_l)
\]  

Where \( l \) represents the network layer number, and Kernel represents the convolution kernel, each feature graph corresponds to a different convolution kernel, and \( m_j \) represents an option for the feature graph, with a trainable bias parameter \( b_l \) per layer.

The nonlinear transformation takes the data after convolution operation as input and performs nonlinear mapping. The formula of nonlinear operation function used in this paper is:

\[
R = \frac{1}{1 + e^{-x}}
\]

The subsampled function is simple, responsible for the independent operation of each feature map, and the average pool or maximum pooling method is adopted to reduce the data size. Pooling operations minimize the resolution of the feature map and reduce the computational complexity while retaining the original feature map information. The calculation formula of the subsampled layer is:

\[
X^l_j = \sigma(g_{m_j}(X^{l-1}) + b_j)
\]

Among them, \( g(z) \) means the subsampled of the information \( z \), which can be the average or maximum value in the region.

The output layer is the last layer, which is connected to the full connection layer or the subsampled layer. It is a highly abstract expression of the input layer data.

The trained convolution neural networks often employ back-propagation and supervised training methods. The input layer of the convolution neural network is \( X \), and the output layer is \( O \). The output feature \( O \) is compared with the ideal feature \( T \), and the error \( E \) is transferred back to each node of the upper layer, and the weight update is carried out according to the weight formula. In supervised training, the network error decreases as the number of iterations increases, until convergence or stabilization.

For an arbitrary layer of convolution neural networks \( L \), the update formula[6] of the weight \( w_{ij} \) between the \( i \)th input characteristic \( X_i \) and the \( j \)th output characteristic \( Y_j \) is:

\[
\Delta W_{ij} = \alpha \delta_j X_i
\]

When the \( L \) layer is the last layer, in equation (4):

\[
\Delta W_{ij} = \alpha \delta_j X_i
\]
\[ \delta_j = (T_j - Y_j)h'_L(X_j) \]  
(5)

Among them, \(T_j\) is the jth expected characteristic; and \(h'(x)\) is the derivative of nonlinear mapping function; \(j = 1, 2, \cdots, N_L\).

When the L layer is not the last layer, in equation (4):

\[ \delta_j = h'_L(X_j)\sum_{m=1}^{N_{L+1}} \delta_m \omega_{jm} \]  
(6)

Among them, \(N_{L+1}\) is the number of characteristic graphs of layer \(L+1\); \(\omega_{jm}\) is the weight between the jth output of the L layer and the mth output of the \(L+1\) layer.

3. Bayesian network

Naive bayes is a method that uses a priori probability and sample information to compute the posterior probability and divides the sample into the largest posterior probability class. The principle is based on the bayes formula and the simple conditional hypothesis.

According to the bayes formula, suppose \(A_1, A_2, A_3, \cdots, A_n\) is a set of mutually exclusive events, and the event B can only occur with one of the events happened at the same time, equation (7) is set up:

\[ p(A_i / B) = \frac{p(B / A_i)(A_i)}{p(B)} = \frac{\sum_{i=1}^{n} p(B / A_i)(A_i)}{p(B)} \]  
(7)

Suppose \(x = \{a_1, a_2, \cdots, a_n\}\) is an item to be classified, in which \(a\) is the attribute of the sample \(x\). There are class sets \(c = \{y_1, y_2, \cdots, y_n\}\). Finding a training sample set, according to the training sample set, the conditional probability or prior probability of each characteristic attribute is calculated according to the training sample set.

\[ p(a_1/y_1), p(a_2/y_1), \cdots, p(a_n/y_1); p(a_1/y_2), p(a_2/y_2), \cdots, p(a_n/y_2); \cdots; p(a_1/y_n), p(a_2/y_n), \cdots, p(a_n/y_n) \]

Suppose that the attributes of each attribute are conditionally independent, according to bayes' theorem:

\[ p(y_i / x) = \frac{p(x / y_i)p(y_i)}{p(x)} \]  
(8)

In formula (8), since the denominator \(p(x)\) is fixed, it is only necessary to compute the maximum value in the molecule. Because the attributes of each attribute are independent of each other, so:

\[ p(x / y_i)p(y_i) = p(y_i)\prod_{j=1}^{m} p(a_j / y_i) \]  
(9)

According to the calculations, if \(p(y_k / x) = \max \{p(y_1 / x), \cdots, p(y_n / x)\}\), then sample \(x_{yk}\).

4. This algorithm

This paper uses the convolutional neural network to reduce the attributes of the training data set. Then, the bayesian network is constructed by using the mutual information construction algorithm[7], and the parameter learning is carried out. Finally, the network is updated and tested according to the sliding window and the relative Euclidean distance.
As shown in figure 2. The input layer is 6*6, a total of 36 attributes, and the output layer contains 8 attributes. The middle layer consists of a roll layer and a subsampled layer. Among them, the convolution layer uses two different 3*3 convolution kernels, and the subsampled layer uses average pooling method to reduce the resolution.

The sliding window was first applied to the transport layer and data link layer in the OSI hierarchical protocol for flow control. This paper introduces sliding window for real-time update training set. Because the training set is updated, parameter learning alone can not satisfy the expression of bayesian networks. Therefore, the relative Euclidean distance is defined to measure the change of mutual information when the training set is updated. The relative Euclidean distance formula is as follows:

$$q_{tk} = \frac{d_{tk}}{m_k} = \frac{\sum_{i,j}(I_{w_j}^{tj} - I_{w_k}^{tj})^2}{\sum_{i,j}(I_{w_j}^{tj})^2}$$  \hspace{1cm} (10)$$

Among them, $d_{tk}$ represents the Euclidean distance of the mutual information between window $\omega_t$ and $\omega_k$; $m_k$ represents the Euclidean distance of the mutual information under the $\omega_k$ window; $\omega_k$ represents the current window, and the initial state is $\omega_1$; $\omega_t$ represents the tth sliding window; $I_{w_j}^{tj}$ represents the mutual information between column i and j in the current window $(i \neq j)$. When the relative euclidean distance $q_{tk}$ is greater than the threshold $\varepsilon$, it is shown that the current bayesian network structure can not accurately reflect the situation of the data set and needs to update the bayesian network structure.

The algorithm steps are as follows:

**Step 1** The training set $T_0$ is pretreated (for example, discretization and numerical value) by using the method of document [7], and the data set $T_1$ is obtained;

**Step 2** The data set $T_1$ is reduced by using the convolution neural network, and the data set $T_2$ is obtained;

**Step 3** Based on data set $T_2$, the bayesian network structure generation algorithm of [7] was used to construct bayesian network and conduct parameter learning.

**Step 4** The test set is classified according to sliding window mechanism and relative Euclidean distance. The concrete steps are as follows:

Connect the test set to the tail of the training set; Set four pointers, $P_1, P_2, P_3, P_4$, respectively points to the header and tail of the training set, the header and tail of the test set.

(1)The data set $T$ between pointer $P_1$ and pointer $P_2$ is used as training set to construct bayesian network and parameter learning. Order $t=1$, the initial windows is $\omega_1$.

(2)Check the sample which the pointer $P_3$ point to.

(3)Repeat (2), until $p_3=p_4$ or $p_3=p_2+N+1$, where N is the size of the sliding window for the experimental setup; $t=t+1$.

(4)If $p_3=p_4$, go to (5); or $p_2=p_2+N$, calculates the euclidean distance between the current window and the initial window, if the euclidean distance is greater than the threshold $\varepsilon$, go to (1), or go to (2).
(5) Test finished, calculate the correct rate.

5. Experimental results and analysis

5.1. Intrusion detection data set
This experiment uses the KDD Cpu 1999 data set, which originates from a real network environment and collects 9 weeks of network connection data. Among them, the training data set collected in the first two weeks contains about 5 million network connections, and the test data set for the next two weeks contains about 2 million network connections. Each connection has 41 characteristic attributes and 1 tag attributes. Each tag is labeled normal or abnormal, and the exception type includes a total of 4 categories, 39 small attack types, 22 of which appear in the training set, and the 17 in the test set.

5.2. Experimental results and analysis
The experimental data in this paper are derived from the kddcpu.data-10-percent data set and the corrected data set of KDD Cpu 1999 data set. From the kddcpu.data-10-percent dataset, 40 thousand random data are selected as training sets T0; Extract 20 thousand data from the corrected data set as the test set M1; From the corrected data set, 10 thousand unknown attack data are filtered and selected as the test set M2. After data preprocessing, artificial delete seventh, ninth, eleventh, fourteenth, fifteenth properties which are irrelevant to classify, and reduce the attribute of the original data set to 8 attributes with the convolutional neural network, with a total of 9 attribute attribute. Finally, uses the improved bayes network to model and classify.

5.2.1. Test results and analysis of data set M1
This experiment was completed in Matlab2015b. In this experiment, the sliding window N is 500, 1000 and 2000, and the relative euclidean distance threshold $\epsilon$ is 0.15, 0.2 and 0.25. The total 9 cases are compared. The classification accuracy is shown in table 1. The experimental data show that when the N is 1000 and epsilon is 0.15, the detection effect is the best. In this process, the bayesian network has undergone 3 changes, and its structure is shown in figure 3.

| Parameter | Type% |
|-----------|-------|
| Normal    |    |
| DOS       |    |
| R2L       |    |
| U2R       |    |
| Probe     |    |

Table 1 Comparison of correct rates under different parameters

| Parameter | Type% |
|-----------|-------|
| Normal    |    |
| DOS       |    |
| R2L       |    |
| U2R       |    |
| Probe     |    |

5

5
Figure 3 Bias network structure change diagram

The algorithm (DLIBN) is compared with several other intrusion detection algorithms, as shown in table 2. In addition to the Normal classification accuracy rate is slightly lower than the literature [8], the other four are higher than the other three algorithms. Among them, the detection rate of R2L is much higher than the other three algorithms, but still far lower than the other four types of detection accuracy. This is because the frequency of R2L in the data set is very low, leading to deep learning can not learn its abstract features.

Table 2  Accuracy comparison table

| algorithm          | Record type/% |
|--------------------|---------------|
|                    | Normal | DOS | R2L | U2R | Probe |
| Literature [8]     | 99.06  | 95.86 | 12.74 | 99  | 78.54 |
| Literature [9]     | 98.23  | 93.21 | 7.50  | 25.00 | 94.80 |
| Literature [10]    | 96.87  | 6.89  | 8.77  | 70.62 |
| DLIBN              | 99.00  | 96.30 | 62.50 | 99.02 | 95.11 |

5.2.2. Test results and analysis of data set M2
The attack type data in training set T0 and the test set M2 are all classified as unknown attack UA and labeled the same attribute. Then Bayesian networks are constructed and parameter learning is performed. The classification results are shown in table 3. The experimental results show that the accuracy of classification is independent of $\varepsilon$. When N is 1000, the classification effect is the best. This algorithm has the ability to detect unknown attacks, but it is still low.

Table 3  Unknown attack accuracy comparison

| parameter | type | parameter | type | parameter | type |
|-----------|------|-----------|------|-----------|------|
| 500       | 0.15 | 80.14     | 1000 | 0.15      | 82.57| 2000 | 0.15 | 81.32 |
| 500       | 0.2  | 80.14     | 1000 | 0.2       | 82.57| 2000 | 0.2  | 81.32 |
| 500       | 0.25 | 80.14     | 1000 | 0.25      | 82.57| 2000 | 0.25 | 81.32 |
6. Conclusion

Based on the KDD CPU 1999 data set, this paper uses deep learning to reduce the dimension of the data set, and introduces sliding window and euclidean distance to update the bayesian network in real time, so as to improve the performance of the algorithm. Experimental results show that the improved algorithm can effectively reduce the computational complexity and improve the accuracy of classification, and has a certain detection ability for unknown attacks. Although deep learning has the ability to extract high-level abstract features from multidimensional data, it is still difficult to learn complex logic behavior in high-level application behavior. It is necessary to further study and study how to detect unknown attacks or abnormal data according to the characteristic behavior of application layer.

References

[1] FENG Zu-hong,LI Jing. Bayesian network intrusion detection technology based on principal component analysis and sliding window[J]. Modern Electronics Technique, 2012,35(19):73-75.

[2] Hinton G,Salakhutdinov R. Reducing the dimensionality of data with neural networks[J]. Science,2006,313(5786):504-507.

[3] Hubel D H, Weisel T N. Receptive fields, binocular interaction and functional architecture in cat's visual cortex [J]. The Journal of Physiology, 1962, 160:106-154.

[4] Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position[J]. Biological Cybernetics, 1980, 36(4):193-202.

[5] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[6] HINTON G E. How neural networks learn from experience [J]. Scientific America, 1992, 267(3):145-151.

[7] WANG Yue, TAN Shu-qiu. Bayesian Network Structural Learning Algorithm Based on Mutual Information [J]. Computer Engineering, 2011, 37(7):62-64.

[8] WEN Jiao, WANG Weiping. Intrusion Detection Method Based on An Improved Bayesian Algorithm [J]. Computer Engineering, 2006, 32(12):160-162.

[9] ZHAO Jian-hua, LI Wei-hua. Application of Supervised SOM Neural Network in Intrusion Detection [J]. Computer Engineering, 2012, 38(12):110-111.

[10] DAI Hong. Application of Support Vector Machine in Intrusion Detection [J]. Computer Engineering, 2012, 38(4):143-145.