Abnormal Detection to Big Data Using Deep Neural Networks

Jian Zheng¹*, Zhaoni Li², Jiang Li³ and Hongling Liu⁴

¹, ³, ⁴ Chongqing Aerospace Polytechnic, 401331, Chongqing, P.R.China
² College of Computer, Qinghai Normal University, Xining, 810008, P.R.China

*Corresponding author email: zhengjian.002@163.com

Abstract. It is difficult to detect the anomalies in big data using traditional methods due to big data has the characteristics of mass and disorder. For the common methods, they divide big data into several small samples, then analyze these divided small samples. However, this manner increases the complexity of segmentation algorithms, moreover, it is difficult to control the risk of data segmentation. To address this, here proposes a neural network approach based on Vapnik risk model. Firstly, the sample data is randomly divided into small data blocks. Then, a neural network learns these divided small sample data blocks. To reduce the risks in the process of data segmentation, the Vapnik risk model is used to supervise data segmentation. Finally, the proposed method is verify on the historical electricity price data of Mountain View, California. The results show that our method is effectiveness.

Keywords: Abnormal detection, big data, deep neural networks.

1. Introduction

It is a major challenge to detect abnormal behavior in a high-speed network environment and massive amounts of data in the field of big data security. Using traditional detection methods, such as misuse detection, is difficult to be used for the security detection of big data. Researchers integrate various machine learning methods to improve the detection accuracy, meanwhile, to reduce the dimensionality of the sample data to increase the detection speed. For instance, the fragmentation (segmentation) mechanism of data stream, whose core idea is to divide high-speed and massive data into different parts and assigned to distributed independent detection nodes for parallel processing.

Data segmentation is the key point to achieve distributed parallel detection of data. Since the quality of segmentation algorithms directly affect the detection results, the ideal segmentation algorithm should be consider as following factors: (1) The data block size is moderate and the amount of information is sufficient. The segmentation algorithm requires the amount of information contained in the data block as support. The division of the data block size can be determined by the amount of information. (2) The size of the data block obtained by each detection node is as the same as possible. Moreover, the segmentation algorithm is as simple and efficient as possible. There are many existing distributed parallel detection mechanisms, such as the fragmentation mechanism, the hash-based packet distribution mechanism, et al.

The fragmentation mechanism is to preferentially allocate all data blocks connected at a time to a detection node with a shorter queue length. This mechanism does not consider whether the divided data blocks contain the amount of information to support node detection, when the amount of connected data is too large, this may cause congestion in some detection nodes, also increases the load of the detection nodes. Before the data blocks are distributed to each detection node, the data blocks are pre-treated by pre-screening and local caching methods to reduce the burden on the processing nodes. For instance, in [1], the advance screening and local caching are used to pre-treat data blocks. However, this not only increases the complexity of the segmentation algorithm, but also difficultly implement and requires a
certain hardware environment to support. Using a hash-based packet distribution mechanism can reduce the burden of the load balancing module [2], but this manner does not give a basis for the division of data block sizes. The segmentation algorithm inevitably destroys the integrity of the data to a certain extent. Hence, the existing segmentation mechanisms need to be improved to reduce the error ratio caused by distributed parallel detection.

Neural network can establish a classification model with learning ability [3], and analyze a large amount of complex data [4], as well as have the ability to model complex problems and high tolerance to noise data [5]. In various neural network models, there are more applications and researches. The traditional network models have standard BP algorithms[6], the improved BP algorithm with adaptive adjustment of additional momentum and learning rate, Levenberg-Marquardt (LM) optimization method, etc. The disadvantages of the standard BP algorithm is slow convergence and falls into local minimum. The improved BP algorithm has slow training speed and low learning efficiency, but it converges faster than the standard BP algorithm. The LM method is also easy to fall into local minimum, moreover, it requires a lot of storage space to solve complex problems. In order to make up for the shortcomings of these above mentioned methods, it is necessary to proposed a new method, which requires the training target to be classified with high accuracy and not easy to fall into local minimization. Indeed, evolutionary strategies have good global search capabilities [7], which can improve search efficiency.

We summarize the main contributions of this work as follows:

(1) The two-hidden layers neural network is designed, which is used to detect anomalies in big data.

(2) This manner of consisting of neural networks and the Vapnik risk model can reduce the risk of data segmentation.

2. Method

2.1. Calculation Data Segmentation Risk

Vapnik-Chervonenkis proposed a machine learning method based on the theoretical framework of statistical learning [8], which not only considers the experience risk of training samples, but also takes into account the generalization ability of the algorithm. Empirical risk is a decision function, which is a function that minimizes empirical risk from a fixed set of functions. Vapnik proved that for any \( \delta \in (0,1] \) and any function \( f \) in the set of candidate functions, the following inequality holds at least with a probability of \( 1 - \delta \):

\[
R(f) \leq R_{\text{emp}}(f) + \frac{8}{\sqrt{l}}(h(\ln \frac{2l}{h} + 1) + \ln \frac{4}{\delta})
\]

where, \( h \) represents the VC dimension, and \( R(f) \) is the expected risk, which is the real risk that exists objectively. \( R_{\text{emp}}(f) + \frac{8}{\sqrt{l}}(h(\ln \frac{2l}{h} + 1) + \ln \frac{4}{\delta}) \) is structural risk. \( R_{\text{emp}}(f) \) is empirical risk. \( \frac{8}{\sqrt{l}}(h(\ln \frac{2l}{h} + 1) + \ln \frac{4}{\delta}) \) is confidence range, which is the increasing function of \( h \). The larger the \( \frac{8}{\sqrt{l}}(h(\ln \frac{2l}{h} + 1) + \ln \frac{4}{\delta}) \) is, the worse the generalization ability is. Therefore, structural risk is an upper bound of expected risk. When \( h \) is larger, the experience risk is smaller, the confidence range becomes larger, and the generalization ability becomes worse. Vapnik proposes the principle of structural risk minimization: select the subset with the smallest structural risk, in this The smallest subset solves the function that minimizes the empirical risk. This smallest function is the optimal function. In order to prevent the loss of segmented data and ensure the integrity of the data, the principle of structural risk minimization is adopted in the distributed learning process.

2.2. Model Description

A two-hidden layer BP neural network is proposed, which is formulated in detail as follows.

Input layer. The layer receives the original input data.

Hidden layers. The layers include the encoding hidden-layers and the decoding hidden-layers, which are
designed as following, respectively.
For the encoding hidden-layers, the input \( H_i'(in) \) and the output \( h_i'(out) \) of the \( i \)-th hidden layer in the \( l \)-th iteration are calculated using Eq(2) and Eq(3), respectively, having
\[
\begin{align*}
    h_i'(l) & = \text{act}_l \left( (w_i(l)H_i'(in) + b_i(l)) \right) \quad (2) \\
    H_i'(in) & = h_i'(l)(out) \quad (3)
\end{align*}
\]
For the decoding hidden-layer, correspondingly, the corresponding output of the \( j \)-th hidden layer in the \( l \)-th iteration are calculated, as following
\[
\begin{align*}
    h_j'(l) & = \text{act}_l \left( (w_j(l)H_j'(in) + b_j(l)) \right) \quad (4) \\
    H_j'(in) & = h_j'(l)(out) \quad (5)
\end{align*}
\]
where, the items \( \text{act} \) is activation functions. \( w \) and \( b \) are the weights and bias of each hidden layer.
Output layer. The layer is used to output results.
Cost function. The cost function, denoted as \( J(w,b) \), consists of the mean square error (mse) and the error function \( \phi(y,f_w(x,w)) \) in Eq (6). For the solution of the error function, the approximate value can be obtained by Taylor series expansion.
\[
J(w,b) = \text{mse} + \phi(y,f_w(x,w)) \quad (6)
\]
where, \( w \) and \( b \) are the weights and bias of each hidden layer.

3. Experimental Simulation
(1) Dataset description
The experimental test uses historical electricity price information from 2010 to 2015 in Mountain View, California [9]. Because electricity prices change in real time, the sample data is processed first. The sample data is roughly divided into five categories: industrial electricity (denoted as \( I \)), agricultural and commercial electricity (denoted as \( A \)), and residential electricity (denoted as \( R \)), Municipal electricity (denoted as \( G \)), other electricity (denoted as \( O \)). Each type of electricity price includes three attributes: peak period (denoted as \( pt \)), valley period (denoted as \( bt \)), and flat period (denoted as \( at \)). Each of these three attributes includes two sub-attributes: summer (denoted as \( sr \)), winter (denoted as \( wr \)). We set the one-hour electricity price as a data sample, and the one-hour electricity price of these five types of electricity is expressed as \( I^{\frac{1}{r}}, A^{\frac{2}{r}}, R^{\frac{3}{r}}, G^{\frac{4}{r}}, O^{\frac{5}{r}} \), where, \( \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5 \in [0, 23] \). The sample data contains 24 numeric attributes \{0, 1, 2, ..., 23\} and 5 symbol attributes \{\( pt, bt, at, sr, wr \}\).
(2) Benchmark test
In order to verify the effectiveness of the algorithm, the algorithm is first benchmarked. The test sample is selected from the test data of the "3rd International Knowledge Discovery and Data Mining Tools Competition" [9]. 2000 data were randomly selected as training samples from the test data. The test data contained 23 target classifications. Each sample data contained 41 attributes, including 7 symbol attributes and 34 continuous attributes. In addition, we also compared the method in [6]. In the whole test process, when the accuracy of the two algorithms is the highest, the number of training times and the number of neurons are listed in Table 1.

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Algorithm} & \text{Training steps} & \text{Accuracy} & \text{Neurons scale} \\
\hline
\text{Our method} & 380–525 & 81%–88% & 117–207 \\
\text{Method in [7]} & 688–1003 & 73%–78% & 73–165 \\
\hline
\end{array}
\]

(3) Testing results
In order to verify the performance of this algorithm, 200,000 pieces of data are selected as training samples. The learning results are shown in Fig.1. The learning results of the 16th to 37th blocks are relatively stable, and the stable knowledge will be stored in the corresponding neurons.
Figure 1. The number of types of learning behaviors for each data sample.

Figure 2. Comparison of prediction rate and actual detection rate.

The sample is used to predict the accuracy of the theoretical model before testing, and the prediction result is compared with the experimental simulation result. The comparison result is shown in Fig. 2. It can be seen from Fig. 2 that the sample attributes are 1-8, and the actual detection accuracy is lower than the predicted accuracy. The sample attributes are 9-25, and the predicted results are closer to the experimental results, and the error is relatively small. It can be seen from the learning result in Figure 2 that when the learning result is relatively stable, the knowledge is stored in the corresponding neuron. At this time, the accuracy of the experimental detection is close to the accuracy of the model prediction.

4. Conclusion

In this paper, we proposed a detection method of consisting of neural networks and the Vapnik-Chervonenkis model, to reduce the risk of segmentation data. Finally, the effectiveness of the proposed method is verified through benchmark tests and simulation experiments. Compared with competing methods, our method outperforms them in terms of detection precision. In the future work, we will continue to explore some new methods in regard to anomaly detection in big data.
Acknowledgement
This research was financially supported by Project of science and Technology Department of Qinghai Province (2016-NN-161).

References
[1] Lai H, Cai S, Huang H, et al. A parallel intrusion detection system for high-speed networks[C]. In: Jakobsson M, ed. Proc. of the Applied Cryptography and Network Security: 2nd Int’l Conf. Berlin, Heidelberg: Springer-Verlag, 2004.pp.439–451.
[2] Charitakis I, Anagnostakis KG, Markatos E. An active traffic splitter architecture for intrusion detection[C]. In: Kotsis G, ed. Proc. of the 11th IEEE/ACM Int’l Symp. on Modeling, Analysis and Simulation of Computer Telecommunications Systems. Washington: IEEE Computer Society, 2003. pp.238–241.
[3] Shang Lin, Wang Jin-Gen, Yao Wang-shu, et al. A Classification Approach Based on Evolutionary Neural Networks [J]. Journal of Software, 2005, 16(9). pp.1577–1583.
[4] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, et al. A deep learning approach for network intrusion detection system[C]// In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), 2016:21-26.
[5] Ye Yuan, Guangxu Xun, Fenglong Ma, et al. Muvan: A multi-view attention network for multivariate temporal data[C]// In 2018 IEEE International Conference on Data Mining (ICDM), 2018:717-726.
[6] Guo Chong-hui, Tang Huan-wen. Global convergence properties of evolution strategies[J]. Mathematica Numerica Sinica, 2001, 23(1), pp.105–110.
[7] V.Vapnik, Statistical Learning Theory[C]. New York: John Wiley and Sons, 1998.
[8] California ISO[OL]. 2010. http://www.caiso.com/
[9] KDDCUP99 data [DB/OL].(1999). http://kdd.ics.uci.edu/databases/kddcup99/