A Unsupervised Learning Method of Anomaly Detection Using GRU

Zhaowei Qu, Lun Su, Xiaoru Wang, Shuqiang Zheng, Xiaomin Song, Xiaohui Song
Beijing University of Posts and Telecommunications, Beijing, China
{zwqu, shuqiang.zheng, wxr, songxiaomin, songxiaohui}@bupt.edu.cn, 199248ssll@163.com

***Abstract*—With the development of the Internet, malicious acts of illegal access to Internet services are also growing, detection and prevention of these acts become particularly important. In order to detect illegal access, existing research not only uses rule-based and statistical-based techniques, but also uses machine-learning techniques. In the field of security, these studies have been effective in detecting abnormal traffic. However, few studies focus on detecting behavior patterns based on malicious users. Malicious behavior from users tends to have a certain pattern. In this paper, we propose an improved neural network model that uses web log data to identify user anomalies (INNAD). Our experimental evaluation shows that our detection method is more effective than traditional lstm method and svm model in anomaly detection.**

***Index Terms*—Anomaly Detection; Illegal Access; Neural Network; GRU**

I. Introduction

The anomaly detection [1] based on machine learning technology brings new development and breakthrough for web against malicious attacks. Existing methods can be based on a large number of data for automated learning and training, and applied to the actual scene. There are currently four detection methods:

- Statistical learning model [2]: feature extraction and analysis for normal flow. It usually establishes a mathematical model by feature distribution statistics for a large number of samples, then detects abnormity by using statistical methods.
- Text analysis model [3]: Web anomaly detection is ultimately based on the analysis of log text. Therefore, we can draw on some ideas and methods in NLP to do text analysis and modeling. The more successful method is parameter value outlier detection based on hidden Markov model (HMM).
- Clustering model [4]: Normally, normal access is largely repetitive, while intrusions are rare. Therefore, using web access data for clustering analysis, we can identify the abnormal behavior of small clusters and perform intrusion detection.
- One-class classification model [5]: As the web intrusion of black samples is scarce, traditional supervised learning methods are difficult to train. An anomaly detection method based on white samples can be used for sample learning through unsupervised or unclassified models, and can construct a minimum model that can adequately express a white sample as a profile, thus enabling exception detection.

This work was supported by the National Natural Science Foundation of China (Grant No.61672108).

There are challenges to machine learning for web intrusion detection. One of the biggest challenges is the lack of tag data. Compared to a large number of normal access data, invasive samples are scarce and varied, which is a great obstacle to the learning and training of the model. The existing research focus is based on unsupervised methods for establishing a large number of normal log models. Traffic that does not conform to the model is abnormal traffic [7]. Existing models based on cyclic neural networks are often able to distinguish anomalous behavior from normal behavior. However, the number of neural networks has a great influence on the training time and the detection results, and the problem of gradient disappearance in the training process also affects the effectiveness of the method.

In this paper, we propose an improved auto-encoder model that can be quickly trained to fully identify the inner links of long sequences and then provide accurate and effective anomalies for test data. Our approach has a superior effect compared to multiple existing algorithms.

II. Related works

Anomaly detection has received significant focus in recent years. Many existing methods aim to detect abnormal behaviors through review text. However, these approaches are typically not adversarially robust: spammers [8] can carefully select their review texts to avoid detection. Even without knowledge of the detection system, they may mimic normal user reviews as closely as possible. Graph-based approaches detect groups of spammers, often by identifying unexpectedly dense regions of the graph of users and products.

Bryan Hooi et al. used to find dense subgraphs to detect anomalies [12]. Their contribution is to propose a measure of resistance to fraud, making the experimental results improved. However, with the development of web2.0, more and more types of user behavior continue to produce, only map structure of the abnormal model must not have universal, will soon be lagging behind.

The model proposed in this paper is to extract the user behavior sequence from the web log data. We learn the context of the link through a neural network model to determine whether the normal behavior log, whether or not there is a new user operation. We use the simple squared loss objective

2375-9356/18/$31.00 ©2018 IEEE
DOI 10.1109/BigComp.2018.00126

function as a starting point and focus on designing an encoder-decoder RNN architecture that can be used with any loss function. In Section 4 we can see that our experimental results have significant stability and accuracy compared to other methods.

III. Model description

In this section, we describe our improved Auto-Encoder Model. The model consists of two parts, encoder model and decoder model. The basic unit of our network is the GRU cell block which is a variant of LSTM. As shown in the Figure 1, the input of the model is a vector sequence (the feature vector of the web log data of the same user under the time series). The encoder model reads the input in sequence. For the input sequence, the encoder outputs an intermediate vector sequence as a series of features that represent the input. This intermediate vector sequence is processed as input to the decoder, and the decoder then minimizes the input by minimizing the reconstruction error. By iterating the process, we can identify the behavior patterns in the sequence data.

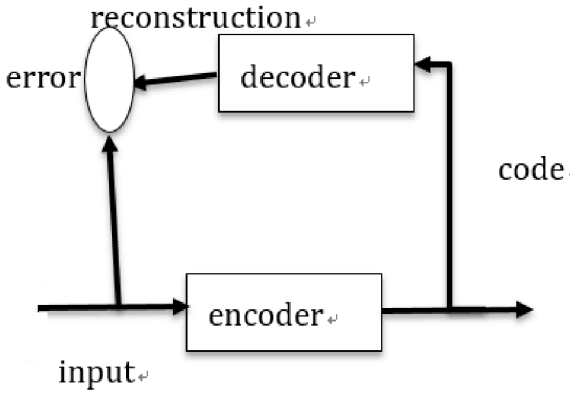

Fig. 1. Auto-Encoder Model

A. Encoder model

Our encoder model is to encode the input log data into an intermediate result that can represent the intrinsic relationship of the input data. The encoder is based on a GRU neural network.

1) Lstm:

Before describing the structure of the GRU, we should look at the structure of the LSTM, which is a variant of the RNN. The LSTM was developed by Hochreiter and Schmidhuber in 1997. As shown in the figure, there are three well-designed gate structures inside the LSTM model to remove or increase information to the cell state. For example, the gate formalized as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1,x_t}] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1,x_t}] + b_i\right) \tag{2}$$

$$\widetilde{C_t} = tanh(W_C \cdot [h_{t-1,x_t}] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{4}$$

$$o_t = \sigma\left(W_o\left[h_{t-1,x_t}\right] + h_t + b_o\right) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

It can ease the difficulty of learning long sequences of data when using backpropagation and can also suppress the vanishing and exploding gradient problems.

2) GRU:

GRU is a variant of LSTM and is proposed by Cho, et al. (2014). As shown in the figure, it combines the forget gate and the input gate to a single update gate. Formalized as follows:

$$z_t = \sigma\left(W_z \cdot [h_{t-1,x_t}]\right) \tag{7}$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1,x_t}]\right) \tag{8}$$

$$\widetilde{h_t} = tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \tag{10}$$

Also mixed with cell states and hidden states, and made some other changes. The final model is simpler than the standard LSTM model. It preserves the advantages of lstm's ability to learn the context of the context, and shortens the training time because the structure is smaller than the lstm. We use GRU as the unit of the encoder and encode the n-dimensional input into a 2 * m-dimensional output. The effect of this output is described in the next section.

B. Decoder model

Our decoder model is to restore the intermediate result of the encoder. The previous encoder-decoder is connected by an intermediate vector c, and we use a 2 * m-dimensional vector sequence as an intermediate result. The encoder needs to encode the input into one such sequence. When decoding, it is possible to fully and accurately use the information carried by the input sequence in order to generate each output. We use a special sampling method, which can learn the input sequence of key information, and the corresponding neglect of non-key information.

1) Sampling:

We treat the output of the encoder (the vector of 2m-dimensional) as the logarithm of the mean and variance of m gaussian distributions, respectively. According to the mean and variance of the encoder output, a random number z corresponding to the corresponding Gaussian distribution is generated. This random number z can be entered into the decoder of the GRU model, resulting in an n-dimensional

2) Optimization goals:

Combining the encoder and decoder, we can output $\hat{x}$ for the same dimension for each x $\in$ X. Our goal is to make $\hat{x}$ and x as close as possible. That is, through the encoded x, can be decoded as much as possible to restore the original information. Since x $\in$ [0,1], we use cross entropy to measure x and $\hat{x}$ differences:

$$xent = \sum_{i=1}^{n} -\left[x_i \cdot log(\hat{x_i}) + (1 - x_i) \cdot log(1 - \hat{x_i})\right] \tag{11}$$

The smaller the xent, the closer the x is to $\hat{x}$. In addition, we need to encode the output $z\_mean(\mu)$ and $z_log_var(log \quad \sigma^2)$of the encoder. Here is the use of KL divergence:

$$KL = -0.5 * (1 + log\sigma^2 - \mu^2 - \sigma^2) = \\ -0.5(1 + log\sigma^2 - \mu^2 - exp(log\sigma^2)) \tag{12}$$

The overall optimization goal (minimized) is:

$$loss = xent + KL \tag{13}$$

With the objective function, and all the operations from the input to the output are derivable, we can train the network through SGD or its improved method. Reparemenrization techniques is used here. Since $z \sim N(\mu, \sigma)$, we should sample from $N(\mu, \sigma)$. However, this sampling operation is not derivable for $\mu and \sigma$, resulting in a conventional gradient descent method (GD) that can not be used by error reversal. With reparemerization, we first sample ε from N (0,1), then $z = \sigma \cdot \varepsilon + \mu$. In this way, $z \sim N(\mu, \sigma)$, and, from the encoder output to z, only involves the linear operation, (ε for the neural network is only constant), therefore, can be used to optimize the GD.

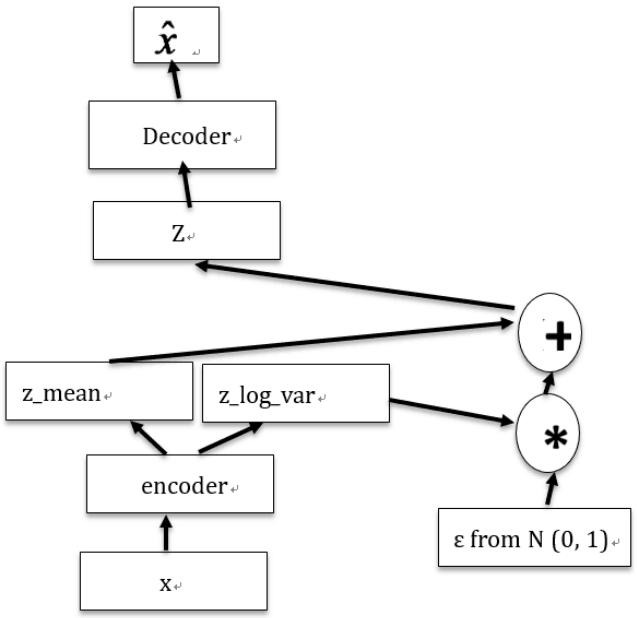

Fig. 2. Complete Detection Model

C. Alogrithm

According to the model described in the previous two sections, our complete detection model which is showed as Figure 2. The encoder receives input to train and divides the output into two parts. These two parts are sampled as input to the decoder.

IV. Experiment

A. Dataset

We use Baidu log data and Amazon log data to carry out our experiments.

TABLE I
NUMBER OF UNIQUE BROSWERS VISITED TO YAHOO JAPAN

Date	*Number of broswers*
April 29,2016	95,123,327
April 30,2016	92,878,076
May 1,2016	92,141,013
May 2,2016	92,050,854
May 3,2016	92,517,378
May 4,2016	93,257,615
May 5,2016	93,291,749

Table 1 shows the number of unique browser visited to Yahoo JAPAN from April 29, 2016 to May 5, 2016. Table 1 shows the details of our dataset. Our testing environment was a CentOS 7.2.1511 virtual server with a Intel Xeon 1.80GHz processor and 8 GB of memory. We choose the same ip log as a log sequence. We log the text data through the n-gram model into a vector form, and then we use the clustering method to reduce the characteristics. The number of feature will be reduced to 10.

B. Anomaly detection

We enter the reduced feature into the previously defined model and train the model parameters by minimizing the error function.

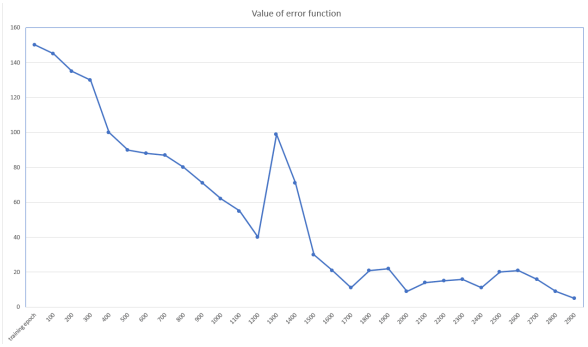

Fig. 3. Value of Error Function

Figure 3 shows the value of the error function as the number of times of training. The training took 21 hours and 51 minutes and 11 seconds.

The following figure is an experimental result of our method INNAD on two datasets:

As shown in figure 4, we can see that our methods have achieved very good results under the two data sets.

We compared the accuracy of our approach with the traditional lstm method and svm model in anomaly detection. The results are shown in the following figure 5:

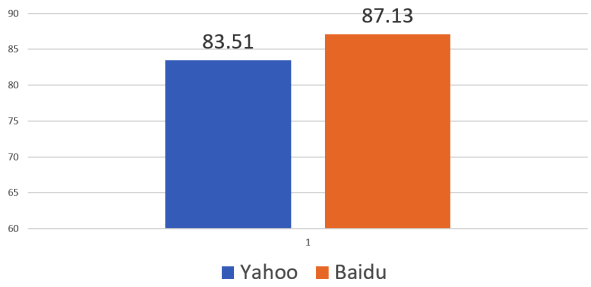

Fig. 4. Accuracy on Two Datasets

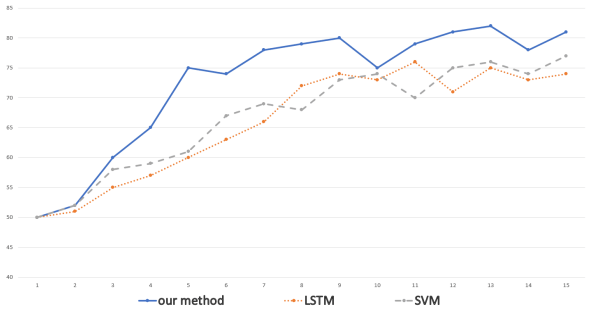

Fig. 5. Compared Results of Three Method

As shown in figure 5, the accuracy of our approach INNAD is higher than traditional lstm method and svm model in anomaly detection. So INNAD has better performance in the field of anomaly detection.

V. Conclusion

In this paper, we propose an exception detection model based on the automatic coding model, we can identify the abnormal behavior patterns in web log data. Our main contributions are as follows:

- Improved model: We first use improved automatic encoders to join the sampling layer to achieve our goal.
- Effectiveness: Our model has been successfully applied to the real world of web log data. It can conduct more accurate detection of abnormal behavior.
- Stability: Our model can learn the far and near behavior patterns to reach the greatest degree of prevention of fraud.

Our experimental evaluation shows that our detection method INNAD is more effective than traditional lstm method and svm model in anomaly detection.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No.61672108). We thank our teacher for providing us with technical support.

References

[1] Lazarevic, Aleksandar, et al. “A comparative study of anomaly detection schemes in network intrusion detection.” Proceedings of the 2003 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2003.

[2] Vapnik, Vladimir Naumovich, and Vlamimir Vapnik. “Statistical learning theory.” Vol. 1. New York: Wiley, 1998.

[3] Nord, Christiane. “Text analysis in translation: Theory, methodology, and didactic application of a model for translation-oriented text analysis.” No. 94. Rodopi, 2005.

[4] Pal, Nikhil R., Kuhu Pal, and James C. Bezdek. “A mixed c-means clustering model.” Fuzzy Systems, 1997., Proceedings of the Sixth IEEE International Conference on. Vol. 1. IEEE, 1997.

[5] Hempstalk, Kathryn, Eibe Frank, and Ian H. Witten. “One-class classification by combining density and class probability estimation.” Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2008.

[6] Bradley, Brenda J., Karen E. Chambers, and Linda Vigilant. “Accurate DNA-based sex identification of apes using non-invasive samples.” Conservation Genetics 2.2 (2001): 179-181.

[7] Kim, Myung-Sup, et al. “A flow-based method for abnormal network traffic detection.” Network operations and management symposium, 2004. NOMS 2004. IEEE/IFIP. Vol. 1. IEEE, 2004.

[8] Stringhini, Gianluca, Christopher Kruegel, and Giovanni Vigna. “Detecting spammers on social networks.” Proceedings of the 26th annual computer security applications conference. ACM, 2010.

[9] Gers, Felix A., Jrgen Schmidhuber, and Fred Cummins, “Learning to forget: Continual prediction with LSTM.” (1999): 850-855.

[10] Chung, Junyoung, et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling.” arXiv preprint arXiv:1412.3555 (2014). In

[11] Graves, Alex, Santiago Fernndez, and Jrgen Schmidhuber, “Bidirectional LSTM networks for improved phoneme classification and recognition.” Artificial Neural Networks: Formal Models and Their ApplicationsI-CANN 2005 (2005): 753-753.

[12] Hooi B, Shin K, Song H A, et al, “Graph-Based Fraud Detection in the Face of Camouflage[J].” Acm Transactions on Knowledge Discovery from Data, 2017, 11(4):1-26.