Anomaly Detection in Industrial Control Networks Using Hybrid LDA - Autoencoder Based Models

Hua Zhang[1, a], Shixiang Zhu[1, b], Jun Zhao[2], Minghui Gao[3], Zheng Shou[2], and Ye Liang[3]

[1]Institute of Network Technology, Beijing University of Posts and Telecommunications, China;

[2]State Grid Liao Ning Electric Power Supply Co. Ltd, China;

[3]Beijing Kedong Electric Power Control System Co. Ltd, NARI Group Corporation, China.

[a]zhanghua_288@bupt.edu.cn, [b]meowoodie@outlook.com.

Abstract. This paper introduces a hybrid model that combines Latent Dirichlet Allocation (LDA) model with autoencoder to detect anomalies in Industrial Control Networks. The autoencoder provides a low-dimensional embedding for the input data, whose subsequent distribution is captured by the LDA model. The autoencoder thus acts as a trainable feature extractor while the LDA model captures the group structure of the data. This new approach potentially completes the strength of signature-based and anomaly-based methods.

Keywords: Anomaly detection; Industrial Control Networks; LDA; Autoencoder.

1. Introduction

Anomaly detection continue to be a significant problem in the communities of Industrial Control Networks (ICNs) [1]. With the rapidly increasing connectivity of control systems to open networks for decentralized management and remote control, ICNs become more and more vulnerable to network attacks nowadays [2-4]. There are a plenty of Network Intrusion Detection Systems (NIDSs) have been widely proposed, implemented, and even deployed, which can be divided into two basic main theories [5, 6]. One is signature based (or rule based) theory, which detects anomaly, attacks, or intrusion by observing network activities in the system and applying a set of rules that determines whether a given pattern of network activity is anomalous or not [5, 7, 8]. The other one is anomaly based theory, which compares some specific features of the activity with an predefined baseline, and the baseline will identify which packet in the network is normal data [6, 9, 10].

Unfortunately, on the one hand, traditional Network Intrusion Detection System (NIDS) makes significant use of pattern matching to identify malicious behaviour and may not detect zero-day exploits in which a new attack is employed. On the other hand, it would cost a tremendous manpower to build an effective NIDS because of labelling and modeling manually each kind of malicious behaviour [6].

To address these limitations, we have made use of techniques from the fields of Deep Learning and Natural Language Processing, specifically, the autoencoders and the Latent Dirichlet Allocation (LDA). Probabilistic graphical models like LDA are a natural way to represent the high level structure of a signal. The effectiveness of using probabilistic graphical models in anomaly detection has been proven in many previous works [11-14]. Equally, autoencoders have proven effective at automatically learning good feature representations from the raw signal [15].

This paper proposes a novel hybrid model that combines LDA with autoencoders and presents a joint unsupervised learning algorithm which means costs of labor would be greatly reduced. We hope to use deep learning techniques to encode the raw features of ICNs' network packets into low-dimensional or tight embeddings without losing any important information, and use LDA to find out latent normal and malicious behaviours in each network packet and each fragment of network traffic.

2. The Hybrid Model

Our work is based on ideas inspired by Natural Language Processing (NLP). In order to use LDA for anomaly detection, we need to map the anomaly detection problem into the topic modeling problem. Fortunately, the network traffics share a great deal in common with text corpora. In analogy to "documents" in corpus, network data consists of segments of network traffics. Furthermore, we liken the feature of a packet in a segment of network traffics to a "word" in a document. Most importantly, we treat the network behaviours of specific traffics as "topics" of documents, like some normal behaviours in ICN: "reading coil ", "writing coil", or some anomalous behaviours: "DDoS attack" and so on. By training LDA model, we hope to attach these "topic" hidden variables to every segment of traffics or even every packet in the network. This would help us to identify those anomalous segments or packets whose "topics" represents malicious behaviours in the network.

In this case, the combination of all possible feature vectors of a packet is regarded as "vocabulary", which has an enormous amount, need to be reduced in order to make the training of LDA more efficient and the result of LDA more accurate. To this end, we need to find out a tighter and lower-dimensional expression of feature vectors as "vocabulary". Here, we consider an autoencoder with one hidden layer, as shown in Fig. 2. An arbitrary number of extra hidden layers (encoders and decoders) could be inserted in the autoencoder if more a complex transformation is preferred.

A general LDA is combined with a simple autoencoder to form a hybrid model by encoding raw feature of network packets and finding latent network behaviours. The architecture of this hybrid model as shown in Fig. 1.

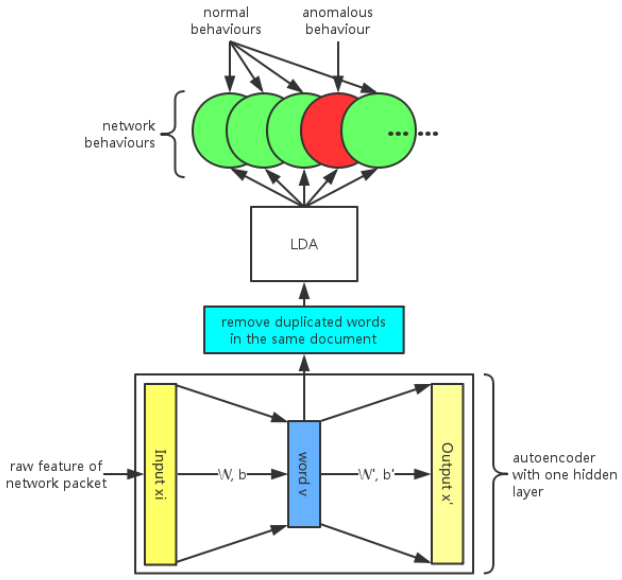

Fig. 1. The hybrid model that combines LDA and autoencoder.

2.1 Encoding raw feature with autoencoder

Given a network packet in ICNs, we first need to extract raw feature from it. Commonly, the raw feature of network packet is a high-dimensional real-valued vector. Take data set KDD-CUP-99, which was the most wildly used data set for the evaluation of anomaly detection methods, as an example. The each feature vector of the data set contains 41 real-valued features and is labeled as either normal or an attack, with exactly one specific attack type. And in ICNs, it would contains more feature due to the industrial protocol (the application layer) of the packet contains additional information. Thus, it means the combination of all possible feature vectors, i.e. "vocabulary", of the ICNs packet has an enormous amount, much more than the amount of the words in any kind of language. It raises two problems:

1) An important deficiency in network data is the huge number of redundant records. About 78% and 75% of the records in the KDD train and test data set are duplicated and similar, respectively [18]. This large amount of redundant records in the train set will cause learning algorithms to be biased towards the more frequent records, and thus prevent it from learning unfrequent records which are

usually more harmful to networks such as U2R attacks. The existence of these repeated records in the test set, on the other hand, will cause the evaluation results to be biased by the methods which have better detection rates on the frequent records.

2) In order to get a reasonable result by training our model, we need a huge number of data to cover the most of the possible combinations of feature vectors, which means the training process will be inefficiency. Otherwise, we cannot get an accuracy result at all because the inadequate training data doesn't stand in for a real data population.

Therefore, in this paper, we use a simple autoencoder with only one hidden layer to encode every raw feature vector x_i in a specific segment of the traffic into a low-dimensional binary "*word*" v_i, and remove those duplicated words in the same segment of the traffic to form a "*document*", see Fig. 3.

2.2 Using LDA to find anomaly

As to every single raw feature vector $\mathbf{x_i}$, the autoencoder will transform $\mathbf{x_i}$ into a binary word vector $\mathbf{v_i}$. Then we divide these word vectors in the whole network traffic into segments, which are input to the LDA model, according to the borders of sessions in ICNs protocols. In terms of the probabilistic graphical model representation of LDA shown in Fig. 2, we use the idea of LDA to assume that network data being processed was generated as follows:

* Pick $\mathbf{N_j}$, the number of the "words" that segment of network traffic ("document") will have in the ICNs.

* Pick $\boldsymbol{\theta_j}$, the distribution of network behaviours ("topics") in this segment of network traffic, from a Dirichlet distribution parametrized by $\boldsymbol{\alpha}$.

* Generate each "word" $\mathbf{v_i}, \mathbf{0 \leq i \leq N}$ in this segment of network traffic by:

- Pick a network behaviour $\mathbf{z_{j,i}}$ ("topic") from multinomial distribution parametrized by $\boldsymbol{\theta_j}$;

- Pick a "word" $\mathbf{v_i}$ given the distribution of "words" for the chosen topic $\boldsymbol{\theta_j}$

Assuming this generative model for a collection of documents, LDA then tries to backtrack from the documents to find a set of network behaviours ("topics") that are likely to have generated the collection. The remainder of the algorithm details including inference of LDA can be found in [16].

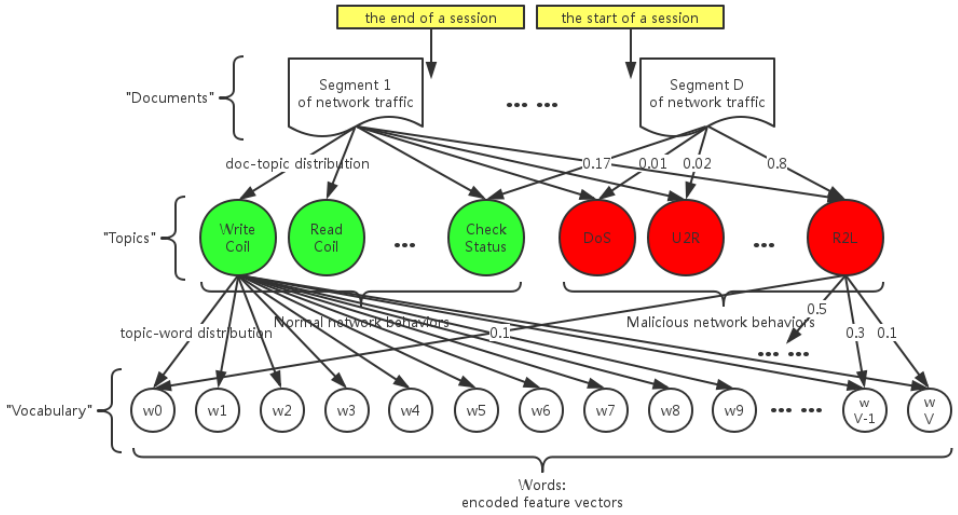

Fig. 2. The example of LDA for anomaly detection in ICNs.

To illustrate the above process, consider the following scenario shown in Fig. 4. We assume that the some possible network behaviours ("topics"), normal behaviours such as "write coil", "read coil", "check status", and malicious behaviours such as "DoS" (Denial of Service Attack), "U2R" (User to Root Attack), "R2L" (Remote to Local Attack), are monitored and detected by our model, as shown in the middle layer in Fig. 4. The collection of those behaviours types form the encoded feature vectors ("vocabulary"). In the figure, each "document" represents a segment of network traffic which is divided by the borders of sessions; each "word" in the "document" represents an encoded feature vector which is processed by autoencoder that mentioned before.

At runtime, a new segment of network traffic D, i.e. a "document", associated with latent network behaviours or activities over a predefined period of time, is collected. However, LDA is an

unsupervised learning method that maximizes the probability of word assignments to one of K fixed topics. We need to extract the meanings of the behaviours (correlate the behaviours to the fixed topics) by interpreting manually the top N probability encoded feature vectors ("words") for a given topic, which means LDA will not output the meaning of topics, rather it will organise words by topic to be interpreted by the user. Back to the example that is shown in Fig. 4, based on the encoded feature vectors ("words") contained in the "document", its "topics" distribution is calculated as follows: “Check Status” (0.17), “DoS” (0.01), "U2R" (0.02) and "R2L" (0.8). The obtained probability distribution indicates that the observed behaviour corresponds in a high probability to a malicious behaviour, i.e. Remote to Local Attack, (rather than legitimate activity); hence an alert should be raised.

Generally, we define, in the network behaviour distribution of a segment of network traffic ("doc-topic" distribution), if a probability on a malicious behaviour is higher than a predefined threshold, an alert should be issued; if the probabilities on normal behaviours are fairly high and the probabilities on all malicious behaviours are lower than their thresholds, the behaviour reflected by the segment of network traffic can be regarded as normal; if all probabilities on all network behaviours are low, it means that the behaviour reflected by the segment of network traffic is beyond the current topics, then further investigation needs to be conducted.

3. Experiments and Results

The data analyzed here is a fragment of real Modbus network data that we captured in ICNs. The first half data was used for training the hybrid model, and the last half data was analyzed to detect anomalies. The raw feature vector we defined has 47 features. Except those 41 features that have been defined in KDD-CUP-99, we added 6 more features that contain the information of Modbus ADU (Application Data Unit) in each packet. Table 1 shows these 6 features that we mentioned above.

Table 1. Features of Modbus.

Feature Name	Description	Value
Transaction Identifier	Identification of a MODBUS Request/Response transaction	0 - 65535
Protocol Identifier	0 = MODBUS protocol	0 - 65535
Length	Number of following bytes	0 - 65535
Unit Identifier	Identification of a remote slave connected on a serial line or on other buses.	0 - 255
function code	indicate to the server which kind of action to perform	0 - 255
data	depended on Length field	real number

We used the simple autoencoder with 47 real-valued input/output nodes and 10 binary hidden nodes to encode the raw feature vector into "word", i.e. a binary vector with length 10, then LDA was run on these "words", with $N = 5$. And we predefined 12 network behaviours, which means set the fixed topics size as 12. It is really easy to decipher the clear meanings of the topics that LDA generated because the raw features to which encoded feature vectors ("words") correspond contains specific Modbus fields that have clear meanings. For examples, if the field of function code in Modbus PDU (Protocol Data Field) is bigger than 255, it means that this packet very likely contains an anomaly behaviour, which is buffer overflow. Or if the field of function code in Modbus PDU is 16, it means this packet indicates to write multiple. Table 2 shows part of the predefined network behaviours and their judgement standards.

Table 2. Network behaviours and their judgement standards.

Topic No.	Network Behaviour	Standards
0	Write Multiple Coils	function code=15 in Modbus
1	Read Coils	function code=01 in Modbus
2	DoS	makes some computing or memory resource too busy or too full
3	Buffer Overflow	the value of any field is overflowed
4	Probing Attack	access to those sensitive devices(IP or unit Id)
…	…	… …

The "vocabulary" list is first generated based on the training data by the autoencoder. For each packet in the traffic, the raw feature vector is determined by extracting 46 features from every packet and then forming these features to a binary "word" vector. Then the network traffic can be converted into documents. For each host in ICNs, the network traffic is divided into multiple sessions, and we turn each session into a "document". A LDA model is trained for each host, for the reason that different host may be dedicated to different purposes in ICNs, thus the "topic" distributions could be totally different among hosts. By using a single LDA model for each host, the detection accuracy could be greatly improved.

Attacks are detected during the test phase using likelihood according to the probabilistic graph where β_k is learned in the training phase and θ_d in the test document. For each host, the lowest likelihood of its training documents is used as threshold. The test document whose likelihood is lower than the threshold is labeled as an attack. Given 25000 documents for all the 5 hosts, our method raises 130 documents as anomaly, of which 79 are true positives and 62 are false positives. Since one instance of attack may contain several documents, there are 79 attacks detected in all. Each host has an average of 3.27 false positive documents.

We also compare the efficiency of our method against some other traditional NIDS, including NGIDES and osPCA. NGIDES (Next-Generation Intrusion Detection Expert System) [7], which is a typical of signature-based NIDS, was developed by Stanford Research Institute (SRI). And osPCA (oversampling Principal Component Analysis) [9], as a mainstream anomaly-based method, has also been widely implemented in various application scenarios. In order to show the effectiveness of the hybrid model more directly, in Fig. 3, we show the receiver operating characteristic (ROC) curve of each method, which is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. It can be seen that there is a substantial improvement going from the proposed method to the other two methods since the area under the curve (AUC) of the proposed method (LDA-autoencoder) is much larger than that of other methods.

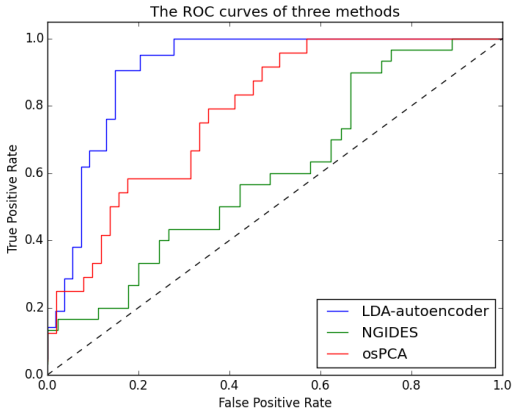

Fig. 3. The ROC curves of three methods

4. Summary

In this paper, we propose a hybrid model that combines LDA with autoencoder to knowledge discovery from big data of network traffic for anomaly detection. We have shown that the model learned using deep learning and NLP techniques have a promising performance that is better than those traditional algorithm used in NIDS. More importantly, the topics identified by LDA catch the latent semantics of a segment of the network traffic in ICNs, which gives an accuracy measurement on the probabilities of containing each possible network behaviour that we concern. As future work, we would consider more factors that affect ICNs network traffic in the probabilistic graph, and make some changes on the topic model, instead of using LDA directly, in order to use this hybrid model in ICNs more satisfactorily.

Acknowledgements

This paper is supported by the project: State Gird Science and Technology Project of 2016 - Application and Research on the Key Technology of Security Protection Framework in Coordination Applying in the Smart Grid Dispatching Control System.

References

[1] B. Galloway and G. P. Hancke, "Introduction to Industrial Control Networks," IEEE Commun. Surv. Tutorials, vol. 15, no. 2, pp. 860–880, 2013.
[2] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber-attacks on SCADA systems," Proc. - 2011 IEEE Int. Conf. Internet Things Cyber, Phys. Soc. Comput. iThings/ CPSCom 2011, pp. 380–388, 2011.
[3] M. Line, A. Zand, G. Stringhini, and R. Kemmerer, "Targeted Attacks against Industrial Control Systems: Is the Power Industry Prepared?" Proc. ACM Work. Smart Energy Grid Secur., pp. 13–22, 2014.
[4] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in SCADA networks," Comput. Secur., vol. 25, no. 7, pp. 498–506, 2006.
[5] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Comput. Networks, vol. 51, no. 12, pp. 3448–3470, 2007.
[6] C. Tsang and S. Kwong, "Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction," Ind. Technol. 2005. ICIT 2005. …, pp. 51–56, 2005.
[7] D. Anderson, T. Frivold, and A. Valdes, "Next-generation Intrusion Detection Expert System (NIDES): A summary," SRI Int., no. May 1995, p. 47, 1995.
[8] P. a Porras and P. G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," Proc. 20th NIST-{NCSC} Natl. Inf. Syst. Secur. Conf., pp. 353–365, 1997.
[9] Y. J. Lee, Y. R. Yeh, and Y. C. F. Wang, "Anomaly detection via online oversampling principal component analysis," IEEE Trans. Knowl. Data Eng., vol. 25, no. 7, pp. 1460–1470, 2013.
[10] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Comput. {&} Secur., vol. 28, no. 1–2, pp. 18–28, 2009.
[11] B. D. Newton, "Using Latent Dirichlet Allocation," no. December, pp. 1–4, 2012.
[12] H. Jingwei, Z. Kalbarczyk, and D. M. Nicol, "Knowledge Discovery from Big Data for Intrusion Detection Using LDA," Big Data (BigData Congr. 2014 IEEE Int. Congr., pp. 760–761, 2014.
[13] C. Cramer and L. Carin, "Bayesian topic models for describing computer network behaviors," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., pp. 1888–1891, 2011.
[14] X. Cao, B. Chen, and H. Li, "Packet Header Anomaly Detection Using Bayesian Topic Models," pp. 1–12, 2016.
[15] P. Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures," ICML Unsupervised Transf. Learn., pp. 37–50, 2012.
[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," J. Mach. Learn. Res., vol. 3, pp. 993–1022, 2015.
[17] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," J. Mach. Learn. Res., vol. 11, no. 3, pp. 3371–3408, 2010.
[18] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009, no. Cisda, pp. 1–6, 2009.