Deep unsupervised system log monitoring*

Hubert Nourtel, Christophe Cerisara, and Samuel Cruz-Lara

Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France
{hubert.nourtel,cerisara,samuel.cruz-lara}@loria.fr

Abstract. This work proposes a new unsupervised deep generative model for system logs. It is designed to be generic and may be used in various downstream anomaly detection tasks, such as system failure or intrusion detection. It is based on the (reasonable) assumption that most log lines follow rather fixed syntactic structures, which enables us to replace the costly traditional convolutional and recurrent architectures by a much faster component: a deep averaging network. Our model still exploits a standard recurrent model with attention to capture the dependencies between successive log lines. We experimentally validate the proposed generative model on a real dataset obtained from a state-of-the-art High Performance Computing cluster and show the effectiveness of the proposed approach in modeling the "normal" behaviour of the system.

Keywords: Anomaly detection · System log · HPC · Deep learning.

1 Introduction

Massive quantity of system logs are produced every second, and analyzing them manually is out of question. However, they contain valuable information related to the status of the system, risks of failures, potential intrusions and attacks, or other types of anomalies that should be detected in advance. A generic approach to predict most of these events is to train a generative model that is able to predict future log lines. When trained on a sufficiently large corpus, the generative model shall capture the "normal" behaviour of the system, and deviations from these predicted logs may be tagged as anomalies. This approach presents several advantages, especially the facts that it does not require any (costly) manual annotation, that it is generic and can be used in various domains and tasks.

We focus in this work on proposing a new deep generative model dedicated to system logs. In a future work, this model will be used to predict system and application failures in advance, by identifying early anomalies that may lead to a process crash. Compared to the state-of-the-art [4], the design of our model is based on two observations: first, system log lines often have a much less variable syntactic structure across words than natural language text; second, massive quantities of logs are continuously generated, which can only be treated with fast inference algorithms. Both observations lead us to propose a new deep architecture that replaces the traditional convolutional and recurrent processing

* Supported by the ITEA 3 PAPUD 16037, the OLKi and CPER LCHN projects.

within line by a deep averaging component, which is at the same time simpler, faster and powerful, as shown in the recent deep learning literature. Furthermore, we argue that the main drawback of this architecture, which makes the modeling of relative word positions more difficult, is not an issue with this type of data, thanks to the fact that system log lines have much less variability in the structures linking words. We thus reserve the more costly recurrent processing to capture cross-lines dependencies, and simplify the modeling of within-line word sequences.

2 Related Works

The literature about unsupervised deep learning methods mainly focuses on representation learning [24, 29] and on deep clustering, with a few additional papers that depart from these mainstream paradigms [23, 20, 9]. Generative models dominate the field, because of their capability to capture the hidden structures within observations, which constitute the only known information in the purely unsupervised setting.

Deep Belief Networks (DBNs) [14] are one of the first successful deep representation learning models. DBNs are formed by a stack of Restricted Boltzmann machines (RBMs) [13], which learn features one level at a time. This greedy layerwise training is finally used to initialize a deep supervised or a deep generative model like Deep Boltzmann Machines (DBMs) [33]. Nowadays, thanks to recent advances in the field [2], much simpler networks are used to learn good representations of the data, such as the class of Autoencoders (AEs) [22, 30, 38, 31]. Notable models of this class are Variational Autoencoders (VAEs) [21], which are bayesian networks with an autoencoder architecture. These generative models, which try to maximize a lower bound of the data likelihood, can perform efficient inference on large datasets. The hidden layer of these models capture the most salient features of the data [10].

Deep clustering is usually performed on the observations (input space) [25], but- may also be applied on the latent (intermediary) representation space [16, 7, 41, 6, 18, 42, 26]. The options for the clustering loss are numerous: k-means loss [40], cluster assignment hardening [39], locality-preserving loss [16], cluster classification loss [15] or agglomerative clustering loss [41] to cite a few.

A special type of unsupervised methods, which is of particular interest in our work, concern the training of models on positive examples only, or on a dataset mainly composed of positive examples plus a minority of negative examples, without any label. These methods are often referred to as *anomaly detection* approaches, or *one-class* unsupervised classifiers. Indeed, the positive class is the normal class, i.e., the class of samples that occur when the system is running correctly, or when the observed entities behave normally. Every sample that deviates from this normal behaviour is considered as belonging to the negative class, i.e., an anomaly. By definition, there are many observed positive samples, and only a few negative ones, and we further do not know where these negative samples occur in the training corpus. The methods that handle such a context include the

one-class SVM [34], which projects all positive samples into a high-dimensional space and computes an hyperplane that is as close as possible from these samples and separates them from the origin, which is assumed to contain all negative samples. This approach is extended in [36] by replacing the hyperplane with an hyper-sphere, and then in [32] by introducing deep neural architectures in these models. Later on, [5] transposes the original one-class SVM model completely into a deep neural architecture, and [28] projects a similar neural architectures as a supervised model by generating pseudo-labels for negative samples.

Other approaches based on deep neural networks include variational autoencoders [27]. [4] further exploits successfully a recurrent neural network on the difficult LANL dataset for anomaly detection in system logs. Deep architectures are also used on other logs datasets, such as [3, 37, 1]. Other recent non-deep approaches for anomaly detection in logs include [12, 11, 19, 35]. A review of the field can be found in [8].

3 Proposed Model

We propose a deep generative model, which generates the next log line based on the previously observed log lines. Such a generative model may be used in several applications, such as systems anomaly detection and intrusion detection, but in this work, we focus on the evaluation of the generative model itself, independently of the application.

The proposed model adopts a hierarchical structure, with a lower level dedicated to the modeling of a single line of text, while the upper level captures dependencies across multiple lines. Conversely to most other works [4], we have decided to not use a recurrent neural network at the lower level, but to rather model word sequences through a Deep Averaging Network [17]. This choice is first motivated by complexity considerations: indeed, recurrent networks are among the slowest types of basic neural architectures, which is the main reason why they are nearly never used in unsupervised generative models that have to be trained on very large corpora, such as word embeddings, which either exploit a fast single layer network (Word-to-Vec), or a small convolutional network (Collobert&Weston embeddings), or yet fast transformer networks (BERT, GPT...). Given the amount of system log lines that are generated every second, we have thus decided to base our model on the Deep Averaging Network, which is another type of extremely fast neural architecture that has already proven to be also very powerful in many applications [17].

Figure 1 shows the first step of the model: this step takes as input a log line tokenized into words. Each word is encoded into an embedding, and is then smoothed through a temporal convolution filter, which outputs a sequence of temporal vectors with the same size as the words embeddings. Then, a dimension-wise max-pooling operation is realized to reduce this sequence of vectors into a single vector: this is similar to a Deep Averaging Network, which, despite its name, can be performed either with an averaging or a max operator.

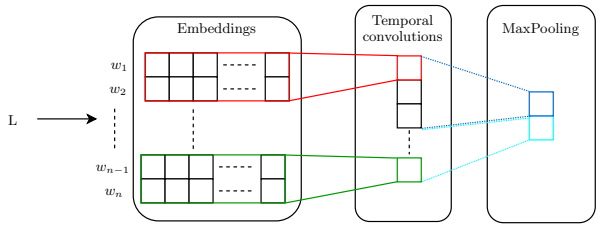

Fig. 1. Predictive model: first step

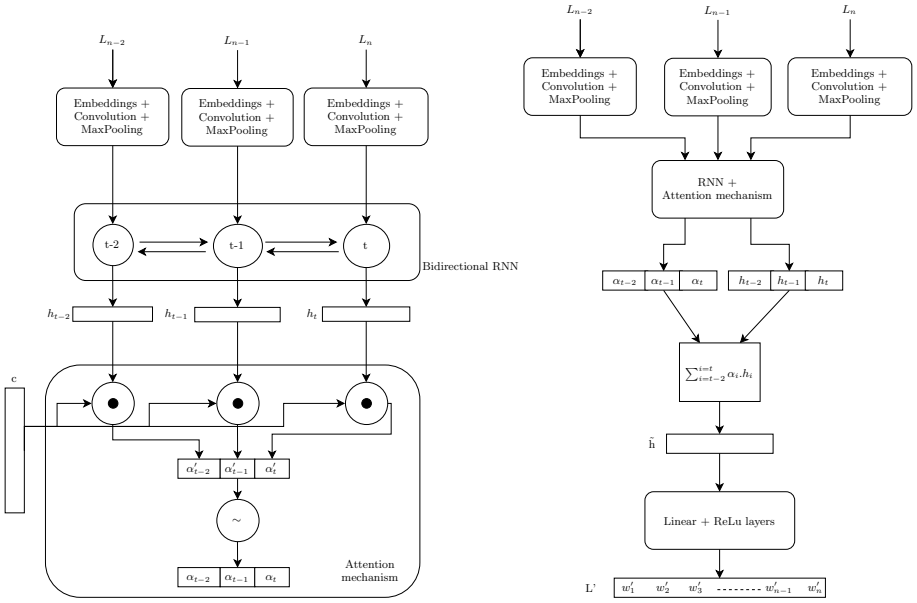

Fig. 2. Predictive model: second and third steps

Figure 2 shows (left) how the line embeddings produced at step 1 are passed to a bidirectional Long-Short Term Memory (LSTM) network with attention: the embeddings of successive lines are passed one after the other into the LSTM, which extracts the most relevant information from these embeddings and cumulates this information into its hidden vector h_t. The LSTM outputs one vector h_t per log line. Then, another parameter vector c is learnt, which role is to weight each h_t through an attention vector α:

$$\alpha_t = \frac{e^{c^T \cdot h_t}}{\sum_\tau e^{c^T \cdot h_\tau}}$$

The right column in Figure 2 explicits how the hidden states are combined: $\tilde{h} = \sum_t \alpha_t h_t$. The summary embedding $\tilde{h}$ is finally passed to a standard feed-

forward neuronal classifier that transforms this vector into a "sequence" of T predicted words: the output dimension of this multi-classification layer is thus $T \times V$ where V is the size of the vocabulary. T softmax operations are applied on this output to obtain word probabilities.

4 Experimental Validation

4.1 Data

We evaluate our model on the Bull-ATOS HPC logs files dataset, which contains anonymized system logs produced by the Deutsches Klimarechenzentrum Supercomputer, ranked #73 in 06/2019 in the TOP500 Supercomputer list. These system logs have been recorded during the execution of real production applications. Every log line contains the following fields : *Timestamp (in seconds) / Node id / User id / Severity / Message*. The training dataset is composed of 318,426 files with 214,379,053 lines; a separate test dataset of 12 files with 5,396 lines is used for validation. An example of sequence of logs is:

```
1527154392 10002 su info pam_unix(su:session): session opened for user b364103 by (uid=0)
1527154392 10002 su info pam_unix(su:session): session closed for user b364103
1527154393 10002 smartd info Device: /dev/sda [SAT], SMART Usage Attribute: 194 Temperature_Celsius changed from 56 to 55
1527154482 10002 pam_slurm info access granted for user root (uid=0)
1527154482 10002 sshd info Accepted publickey for root from 10.50.4.3 port 38260 ssh2
1527154482 10002 sshd info pam_unix(sshd:session): session opened for user root by (uid=0)
```

4.2 Experimental Setup

Every log line is tokenized into a sequence of words, by splitting the line with whitespaces. Then, the length of every words sequence is set to 15 words, after cutting or padding, to make parallel processing easier. All characters are transformed into lower case, and every word that contains one or more digits is replaced by a joker word. Finally, we remove successive lines containing exactly the same words in the same order. The vocabulary contains every word that occurs at least 10 times. Rare words are mapped to the special UNK word. The final vocabulary contains 2,989 words.

Hyper-parameters are set based on reasonable values given in the literature and on a few preliminary experiments: The ADAM optimizer is used with a learning rate of 0.0001 and a batch size of 128. Word embeddings have 100 dimensions. The loss is the cross-entropy between the predicted words and the gold words observed in the following line.

4.3 Results

We compare our generative model in terms of word accuracy, i.e., ratio of predicted words that are correct in all 15-length words sequences, with three baselines in Figure 3.

Our proposed model outperforms every baseline by a large margin. Furthermore, using two lines of context significantly increases its performances as compared to observing only the previous log line, although using more than two lines does not seem to bring further improvements.

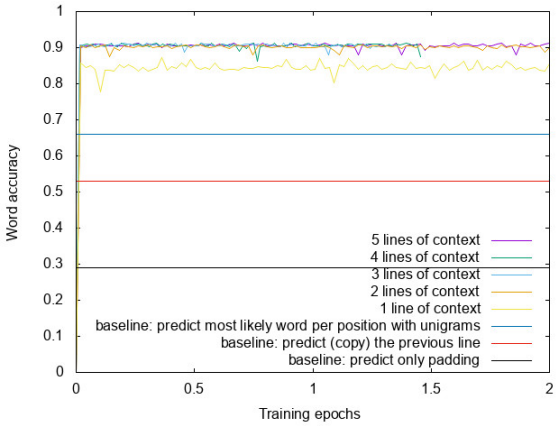

Fig. 3. Prediction accuracy on the Bull dataset over two epochs.

5 Conclusion

We have proposed a deep generative model for predicting system logs. The originality of our model lies in the combination of a fast but powerful component to merge individual word embeddings: the Deep Averaging Network, with a more standard recurrent architecture with attention to model the relation between successive lines. Such a generative model may be used to predict anomalies, system failures or detect intrusions when the proportion of such events is too rare to allow for supervised training. We focus in this work on evaluating the generative capabilities of our proposed model, and experimentally show that it is able to capture correlations both within and across lines to help predict the next log line. In a future work, we plan to exploit attention to build semantically-related chains of events and use the resulting model for anomaly detection.

References

1. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE **2**, 1–18 (2015)
2. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. arXiv:1206.5533 (2012)
3. Bontemps, L., McDermott, J., Le-Khac, N.A., et al.: Collective anomaly detection based on long short-term memory recurrent neural networks. In: International Conference on Future Data and Security Engineering. pp. 141–152. Springer (2016)
4. Brown, A., Tuor, A., Hutchinson, B., Nichols, N.: Recurrent neural network attention mechanisms for interpretable system log anomaly detection. arXiv:1803.04967 (2018)
5. Chalapathy, R., Menon, A.K., Chawla, S.: Anomaly detection using one-class neural networks. arXiv:1802.06360 (2018)

6. Chang, J., Wang, L., Meng, G., Xiang, S., Pan, C.: Deep adaptive image clustering. In: Proc. CVPR. pp. 5879–5887. Honolulu, Hawaii (2017)
7. Dilokthanakul, N., Mediano, P.A.M., Garnelo, M., Lee, M.C.H., Salimbeni, H., Arulkumaran, K., Shanahan, M.: Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv:1611.02648 (2016)
8. Goldstein, M., Uchida, S.: A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PloS one **11**(4), e0152173 (2016)
9. Golts, A., Freedman, D., Elad, M.: Deep energy: Using energy functions for unsupervised training of dnns. arXiv:1805.12355 (2018)
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
11. Gutflaish, E., Kontorovich, A., Sabato, S., Biller, O., Sofer, O.: Temporal anomaly detection: calibrating the surprise. arXiv:1705.10085 (2017)
12. Harada, Y., Yamagata, Y., Mizuno, O., Choi, E.H.: Log-based anomaly detection of cps using a statistical method. In: 2017 8th International Workshop on Empirical Software Engineering in Practice (IWESEP). pp. 1–6. IEEE (2017)
13. Hinton, G.E.: A practical guide to training restricted boltzmann machines. In: Montavon, G., Orr, G.B., Mller, K.R. (eds.) Neural Networks: Tricks of the Trade (2nd ed.), Lecture Notes in Computer Science, vol. 7700, pp. 599–619. Springer (2012)
14. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (Jul 2006)
15. Hsu, C.C., Lin, C.W.: Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. IEEE Transactions on Multimedia **20**, 421–429 (2018)
16. Huang, P., Huang, Y., Wang, W., Wang, L.: Deep embedding network for clustering. In: 22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014. pp. 1532–1537 (2014)
17. Iyyer, M., Manjunatha, V., Boyd-Graber, J., Daumé III, H.: Deep unordered composition rivals syntactic methods for text classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). vol. 1, pp. 1681–1691 (2015)
18. Jiang, Z., Zheng, Y., Tan, H., Tang, B., Zhou, H.: Variational deep embedding: An unsupervised and generative approach to clustering. In: Proc. IJCAI. pp. 1965–1972. Melbourne, Australia (Aug 2017)
19. Juan, D.C., Shah, N., Tang, M., Qian, Z., Marculescu, D., Faloutsos, C.: M3a: Model, metamodel, and anomaly detection in web searches. arXiv:1606.05978 (2016)
20. Kilinc, O., Uysal, I.: Learning latent representations in neural networks for clustering through pseudo supervision and graph-based activity regularization. In: Proc. ICLR (2018)
21. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv:1312.6114 (2013)
22. Lecun, Y.: Modeles connexionnistes de l'apprentissage (connectionist learning models). Ph.D. thesis, Universite P. et M. Curie (Paris 6), Paris, France (1987)
23. Metz, L., Maheswaranathan, N., Cheung, B., Sohl-Dickstein, J.: Learning unsupervised learning rules. arXiv:1804.00222 (2018)
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proc. NIPS. pp. 3111–3119 (2013)

25. Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., Long, J.: A survey of clustering with deep learning: From the perspective of network architecture. IEEE Access **6**, 39501–39514 (2018)
26. Mukherjee, S., Asnani, H., Lin, E., Kannan, S.: Clustergan : Latent space clustering in generative adversarial networks. arXiv:1809.03627 (Oct 2018)
27. Nguyen, Q.P., Lim, K.W., Divakaran, D.M., Low, K.H., Chan, M.C.: Gee: A gradient-based explainable variational autoencoder for network anomaly detection. arXiv:1903.06661 (2019)
28. Oza, P., Patel, V.M.: One-class convolutional neural network. IEEE Signal Processing Letters **26**(2), 277–281 (2018)
29. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434 (2015)
30. Ranzato, M.A., Poultney, C., Chopra, S., Cun, Y.L.: Efficient learning of sparse representations with an energy-based model. In: Schölkopf, B., Platt, J.C., Hoffman, T. (eds.) Advances in Neural Information Processing Systems 19, pp. 1137–1144. MIT Press (2007)
31. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contracting auto-encoders: Explicit invariance during feature extraction. In: Proc. ICML (2011)
32. Ruff, L., Görnitz, N., Deecke, L., Siddiqui, S.A., Vandermeulen, R., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International Conference on Machine Learning. pp. 4390–4399 (2018)
33. Salakhutdinov, R., Hinton, G.: Deep boltzmann machines. In: van Dyk, D., Welling, M. (eds.) Proc. of the Twelth International Conference on Artificial Intelligence and Statistics. vol. 5, pp. 448–455. Clearwater Beach, Florida USA (Apr 2009)
34. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural computation **13**(7), 1443–1471 (2001)
35. Sun, L., Versteeg, S., Boztas, S., Rao, A.: Detecting anomalous user behavior using an extended isolation forest algorithm: an enterprise case study. arXiv:1609.06676 (2016)
36. Tax, D.M., Duin, R.P.: Support vector data description. Machine learning **54**(1), 45–66 (2004)
37. Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., Robinson, S.: Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. In: Workshops at the Thirty-First AAAI Conference on Artificial Intelligence (2017)
38. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proc. ICML. pp. 1096–1103 (Jan 2008)
39. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: Proc. ICML. pp. 478–487. New York (2016)
40. Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In: Proc. ICML. pp. 3861–3870. Sydney, Australia (Aug 2017)
41. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 5147–5156 (2016)
42. Yang, T., Arvanitidis, G., Fu, D., Li, X., Hauberg, S.: Geodesic clustering in deep generative models. arXiv:1809.04747 (Sep 2018)