# Deep Convolutional Neural Networks for Anomaly Event Classification on Distributed Systems

Jiechao Cheng, Rui Ren

[jetrobert19@gmail.com](mailto:jetrobert19@gmail.com), [renrui@ict.ac.cn](mailto:renrui@ict.ac.cn)

## Abstract

The increasing popularity of servers usage has brought a plenty anomaly operation events, which have threatened a vast collection of machines. Thereby, recognizing and categorizing the anomalous events is a much salient work for our systems, especially the ones generate the massive amount of data and harness it for technology value creation and business development. To assist in focusing on the detection and classification of anomaly events, and gaining critical insights from system log records, we propose a series of approaches with the state-of-the-art deep learning techniques in this paper. We employ typical CNN (Convolutional Neural Network) architectures and deepen them to build novel models for automated classification of anomalous system events detected from the distributed system logs, then measure our diverse array of classification algorithms with standard evaluation metrics. The results of our study reveals the advantages and potential capabilities of our proposed deep CNN models for classification tasks of anomaly events on real-world systems. Furthermore, our approach reaches at least 94% classification accuracy and highest accuracy is up to 98%.

**Index Terms**—Anomaly Detection, Event Classification, Log Analysis, Deep Learning, Convolutional neural network, Distributed System

## 1 Introduction

Logs automatically produced by modern heterogeneous operating systems record the application-specific operational states and temporal events at every period of time. Whilst lacking of structure, mining these logs usually facilitates root-cause analysis and troubleshooting for systems, and harnessing the massive amount of valuable log information is much beneficial for business strategies, even detecting and grouping abnormal operations on large-scale systems is favorable for further decision making. Abnormal behavior or pattern of log data often indicates the presence of the error in execution of the system. The ever-growing number of system services during the last decade have stimulated a flood of interest in log anomaly detection and classification. Anomaly detection related problems are addressed in a great deal of practical applications, including intrusion detection, fraud detection, aw well as system health monitoring. Anomalous events are also referred to as novelties, noise, exceptions etc. [1]. Anomaly discrimination as a common approach of log analysis [24], enables us to discover the suspicious operations on a server or detect an unauthorized access in a system by exploring the anomalous operation records. To cope with a complex scenario like real-time system management and reduce the influence of anomaly events affecting the end clients, it is of vital importance to effectively detect the occurrence of anomalies and classify them specifically for further refinement of system service.

Machine learning has been widely applied in a diverse array of anomaly detection and recognition techniques, include but not limited to density-based methods [3, 4, 5, 23], correlation-based anomaly detection [6], cluster analysis-based abnormal detection [7, 8], ensemble techniques [9, 10] etc. However, traditional data mining methods at least to some degree has been overwhelmed by the huge volume of log data. Motivated by the recent success of deep learning largely in field of image classification, natural language processing, speech recognition, etc., specifically, high accuracy performance exhibited by CNN in complex detection and classification tasks, consequently, we decide to explore a solution with deep learning algorithm to handle such complex scenarios for gaining relevant optimum results. Deep learning has been chosen due to its more promising effectiveness in discriminative feature learning and classification process for anomaly detection and categorization based on the large set of original features or much more relevant set of features. Alternatively, labeled data set are taken into consideration here for anomaly classification tasks and fed into the state-of-the-art deep CNN architecture.

Traditional rule-based monitoring is not scalable to deal with the complexities of modern systems, thus it’s so difficult for it to understand all of the patterns and detect anomalous behaviors. Rather than relying on manually set thresholds to detect anomalous behavior, we leverage deep learning techniques to learn complex patterns and detect anomalies efficiently.

We wanted to apply cutting-edge techniques to solve the problem by creating the model with no explicit feature creation or assumption about data distributions.

To classify types of anomalies at scale, we use different combinations of techniques starting with Deep CNN and ending with CNN sophisticated deep learning models.

We propose a simple yet efficient approach to detect and classify system operation anomalies using deep learning techniques. Deep learning algorithms build data-driven models from labeled data and make predictions on data which they can learn from.

Each type of Log contains distinguishing the type of information. To obtain favorable performances of anomaly classification, we propose a deep learning model based on well-known deep convolutional neural network. A convolutional neural network is a classification algorithm that classifies instances by … Deep learning algorithms are like appealing black-box solutions, it’s efficient but very challenging to understand the detailed reasons leading to a particular classification result. In addition, CNN explicitly show …, the learning algorithm automatically select the most discriminating features. Deep CNN…. Last but not least, previous work [] has shown that deep learning outperforms other machine learning algorithms for the sake of log anomaly classification.

By applying deep learning algorithms to our studies, we’ve gained key insights and achieved potentially high classification accuracy.

Our contributions are as follows.

In this paper we extend the anomaly classification research to deep learning, which applying complex architecture with non-linear spatial temporal transformations. Our deep CNN model obtain high classification accuracy with performance measurements of original large scale system log dataset.

The reminder of the paper is organized as follows: section 2 introduce the related works. In section 3, a brief description of CNN architecture is presented. In section 4 we provide the details of our experiments including dataset, the features we employ for classification and detals about parameters settings. In section 5 We discuss our results and analysis. Finally, Finally, we briefly summarize

our findings in the conclusion section.

The remainder of this article is organized as follows: Section 2 briefly reviews the related works. Section 3 reveals our deep learning approaches utilized in the rest of the paper. Section 4 describes the proposed anomaly classification experiments including the generation of the semantic datasets used for models training and evaluation purposes. Section 5 presents the discussions of the obtained anomaly classification results. Finally, the last section concludes out work.

## 2 Related Work

**Anomaly Event Classification.** Sequences and time series data usually need different algorithms to detect anomalies [13]. By using ... , P. Fiadino et al. [19] reported statistical detection and diagnosis of anomalies. Lazarevic et al. [14] compared LOF, k-NN, PCA and unsupervised SVM for intrusion detection. Ding et al. [15] studied SVDD, a k-NN classifier, k-means and a GMM for detecting anomalies. Amer et al. [16] proposed One-class Support Vector Machines for anomaly detection. The local density cluster-based outlier factor (LDCOF) [18] detect anomalies by estimating the clusters’ densities assuming a spherical distribution of the cluster members. Sub-space clustering approaches [20, 27] have also been used in anomaly identification and classification. M. Gupta et al. [22] applied anomaly detection for temporal data.

Combining with multiple anomaly detection algorithms, outlier ensembles boost their joint detection performance [17].

**Deep Learning.** Deep learning algorithms, like CNN, are widely used to categorize data with supervised methods.

Deep learning algorithm takes meta data as an input and process the data through a number of layers of the non-linear transformation of the input data to compute the output. It automatically grasps the relevant features required for the solution of the problem and reduces the burden on the programmer to select the features explicitly.

There has been considerable amount of research about anomaly classification in system log in recent years.

There are some machine learning approaches in anomaly detection and classification, for instance SVM, random forest... Among the machine learning techniques, random forest in widely considered as the recent anomaly classification researches.

It is commonly accepted that deep learning algorithms are well-suited for classification with higher accuracy than other previous techniques.

In this article we apply deep CNN approaches in large scale system anomaly classification.

## 3 Methodology

## 3.1 Convolutional Neural Networks

Convolutional networks have proven very useful in the field of image and video recognition.

Convolutional neural network is well-known for modeling spatial matrix data such as images data. Shallow CNN model did not do well in large scale matrix data.

The lowest layer in the CNN is responsible for the collection of raw data such as images, videos, text, etc. Each neuron of the lowest layer will store the information and pass the information further to the next layer of neurons and so on. So, we can conclude that as the data moves from lowest layer to highest layer more abstracted information is collected.

We briefly introduce baseline CNN architecture and its problem. Then we describe our several deep CNN models to address anomaly classification tasks for large scale system.

First at all, net’s input nodes receive a numeric array, which is then proceeded through the so-called hidden layers until an output or decision of each node is determined by the activation function of that node given an input or set of inputs. At the output layer, the network’s decision about the input is compared to the expected results, and the difference between the network’s guess and the ground-truth labels is utilized to correct the activation thresholds repeatedly to converge on the expected outputs.

Different from conventional feed forward neural networks, CNN have back propagation network .. Assuming that the input vector, the hidden vector and the output vector denoted by X, H and Y respectively. Given that X = (x1,x2,..., xn). ..

Where w is weight matrix, b is a bias vector.

activation function, optimization algorithm, and cost function..

The activation function determines whether and to what extent a signal should be sent to connected nodes. A frequently used activation is just a basic step function that is 0 if its input is less than some threshold and 1 if its input is greater than the threshold. he optimization algorithm determines how the network learns, and more accurately how weights are modified after determining the error. The most common optimization algorithm used is stochastic gradient descent. A cost function is a measure of error, which evaluates how well the neural network performed when making decisions about a given training sample, compared to the expected results.

The more data a deep learning algorithm is trained on, the more accurate it will be.

# 3.2 Deep Convolutional Neural Networks

The traditional CNN consists of one or two hidden layers and this type of structure of the neural network is not suitable for the computation of larger networks. Therefore, a deep stack of the layer of neurons is developed. Deep CNN automatically extracts the features required for the solution of the problem.

Ren et al. [26] show that adding both convolutional and connected layers to pretrained networks can improve performance.

In Deep CNN, each hidden layer is responsible for training the unique set of features based on the output of the previous layer. As the number of hidden layers increases, the complexity and abstraction of data also increase.

Deep CNN ... Figure 3 shows our Deep CNN model, the value computation are described in the following equations.

Q is the logistic sigmoid function,

We can solve the ...problems by using deep CNN.

We applied semantic data, derived from real system operation traces as suggested in []. The data in such format allows analyze the real-time system server operation with a large number of operational anomalies efficiently, moreover, it protects the sensitive information of system services. The procedure of generating semantic data and preprocessing are illustrated as following.

A label can be used as a result indicating whether an instance is an anomaly or not.

Let’s review discriminative algorithms from the perspective of application to finding various types of anomalies. The most suitable type of neural network working with time series is convolutional neural network, if properly built, it allows you to model the most sophisticated dependencies.

The batch size and epoch are 1000, 400 respectively in our CNN architecture. We use relu activation in the fully connected layer and softmax activation in the output layer, Adam gradient decent for the optimizer. The loss function is MSE (mean squared error).

SGD (stochastic gradient descent) gives us the direction of less error, and the learning rate determines how big of a step is taken in that direction. If the learning rate is too high, you may overshoot the error minimum; if it is too low, your training will take forever. This is a hyper-parameter you may need to adjust.

When compared across huge data sets and hyper parameters, different methods have little systematic advantages over another in the measurement performance [11].

Hyper-parameter are crucial for model initialization, unsuitable hyper-parameter settings are not good for model performance. Greff, Klaus, et al. [21] reported that the learning rate and hidden layer size play an important role in the model performance.

the parameters of the algorithm.

Thus, we implement a serials of experiments with different hyper-parameters, i.e., the learning rate and hidden layer size and numbers for algorithm. For the learning rate we pick a value from the set {0.0001, 0.001, 0.01, 0.1}. The possible values for the hidden layer size are {16, 32, 64, 128}.

To find the optimal hyper-parameter values for our models, we do need to run the algorithm with different combination of parameter values.

## 4 Experiments

We performed an extensive set of experiments to assess the effectiveness of our model using several metrics, our data source, and model architectures.

**4.1 Dataset**

Previous work [25] proposed to track the source code to discover the regular layout of log data.

In supervised learning, removing the noise data from the data set often results in a statistically significant increase in accuracy [2].

In fact, many practical anomaly detection problems often require a preprocessing in order to generate the appropriate data to handle with. The final step before the anomaly detection algorithm can be applied is normalization. In practical applications, the min-max normalization is often used, every feature is normalized into a [0, 1] interval, so we apply it in our experiment.

The semantic transformation from a raw unstructured anomaly categorization task to a structured anomaly type classification task requires a solid background knowledge of the data set, which features and instances are so different from the original raw data, namely the generation of a data view [12].

Since the original dataset does not contain any labels, it is more difficult to validate our results. Therefore, we have labeled datasets manually, and apply them to check whether the models we build will be able to detect some typical anomalies in our raw log dataset. The dataset will be subdivided into buckets of single event streaming.

There are 6 month-long data sets, each containing 140 numeric attributes and over 1,000,000 instances. The instances are the events recorded per application per thread, and the class label indicates what anomalous operation system took place.

The datasets that were generated and that are used for this part of the experiments can be found in table 3.

There are three main types of anomaly events, which are depicted in the following graphs.

The enormous amount of log data that has to be processed is an another challenge that has briefly been discussed above as well.

Using a larger set of log data as input for a deep learning model is not always the best choice, as this increases the dimension of the input data, introducing sparsity issues. Therefore, it may negatively impact classification results. Meanwhile, irrelevant or redundant features will bring more noise to the overall process, thus models will obtain inferior performances. We need to prepare, normalize, and vectorize the logs into a numeric array. Log files need to be serialized into the same format that the model trained on,

Log messages on large scale system are used in our experiment to measure the performance of deep CNN classifiers. The original gathered dataset is event-wise log text, so we preprocess the message field property of log records to numeric feature vector, as the input fed to the neural network. The features we selected in our experiments are shown in Table 1. They enable representing the semantic distance between system events and the topic of anomalous behaviour.

This setup started gathering logs from multiple applications within the distributed systems at the beginning of March 2017. Therefore, plenty of log data from different production servers is available for our experiments.

There are 1,000,000 system events that has been gathered from applications on the servers in the dataset and each log message has 140 numerically event-wise features, our models classify them into 13 different classes according to their characteristics. Table 1 reveals the categories of anomalous instances. Internet anomaly instance indicates ...

We here apply 10 percent of original system log dataset for training and test, because of large scale of them. The data ratio of database anomaly instances is bigger than others presented in figure 4, thus database anomalous instances are able to recognized easily, and it will be unfair to the overall model training and evaluation.

We total include 14 different classes of anomalies, each one is assigned a class.

We construct a fully labeled dataset spanning a period of a couple of months with consecutive days. The construction of the semantic data set is conceived with the objective of fundamentally maintaining the underlying structural characteristics of the raw temporal operation data as much as possible. The transformation procedure is described as follows. The first step of the construction procedure consists of manually labeling. Then, we transform the textual information into structured representation.

We divide this vector into m blocks, each one corresponding to a two-minute interval….

The data set gained in this way retains certain features of real log data. It keeps the time-series variations of system operation, also, it maintains the differentiation among a variety of anomaly types.

In order to categorize the anomalies, we take the distributions of anomalies across the types described in table 1.

During several months of trials, we successfully group anomalies with high accuracy. We aim at modeling...

We normalize all the features to an interval from zero to one by mapping a feature value x to …. The input vector consists of 140 features and output vector is comprised of 13 anomaly classifications. As a result, the dimension of input and output is 140 and 13, respectively.

We ca analyze the classification performances for messages aggregated per class.

**4.2 Settings**

Training the neural net is the step that will take the most time and hardware. Running training on GPUs will lead to a significant decrease in training time. We describe our detailed experiments settings in this section, including optimal hyper-parameter values for our deep CNN models to obtain best performance. Our experiment run environment configuration is listed as below:

CPU: Intel Xeon E5-2630 2.4 GHz

GPU: Nvidia Tesla M40

RAM: 64GB

OS: Ubuntu 16.04

In order to evaluate our model, we established various CNN models with different convolutional layers, the size of hidden layer, learning rate and dropout probability.

**4.3 Evaluation Metrics**

Training data is used to recognize the anomaly types within the system. Now, evaluate the performance of the model using test dataset.

This is the process of supervised learning i.e. log data patterns can be defined in advance.

To make a fair comparison between the anomaly performances of classification to determine which model performs best, we specifically extract a confusion matrix with the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) counts

from the results of an algorithm. Then, we can computer four standard evaluation metrics (best accuracy, precision, recall and f1-score) shown in equations 1, 2, 3 and 4 respectively.

Accuracy denotes.., Precision signifies..., Recall ..,F1-score ... Equations of the metrics based on confusion matrix are presented as follow:

**4.4 Results and Discussion**

Using more features increases the dimensionality of the feature space, usually bring in undesirable effects like sparsity, and some redundant or irrelevant features may diminish performance of models in classification.

In this section we describe the proposed anomaly classification approach based on deep learning, focusing on the principal features as input. Deep learning is widely employed lately as it is very efficient in a large number of scenarios, especially for huge amount of high-dimensional datasets.

Besides convolutional neural network approach, we consider deep learning approach in our work. In addition, ... It is clear that the selection of features for classification tasks plays a major role in its empirical performance. CNN approach generally consider the temporal analysis of certain features, it employs a powerful ... to build appealing ...

Convolutional neural network is composed of multiple layers of neurons, each of them generally represented by a non-linear function [], every neural employs an activation function that maps the weighted inputs to the output that is passed to the following layer. The weights, originally set to random values, are iteratively adjusted during the training phase.

Global accuracy, recall and precision, F1-score are consider here to evaluate the performance of our deep learning classification models. Global accuracy Ai indicates the… Recall Ri means, … Precision Pi is… These four standard metrics are widely used for performance evaluation in classification tasks. Accuracy measures…, precision measures…, recall measures…, whereas F1-score measures..

Figure 2 depicts the performance comparison of the 2 classifiers in the classification of all 14 anomalous types. To decrease the influence the bias might bring in to all the evaluations, we employ 8-fold cross-validation with different random splits of the dataset, which indicates that we train and test our models in 8 different train/test dataset combinations.

There are no particular bias for both classifiers. The CNN classifier shows a slightly higher variance in the results, which might suggests that the model is slightly less robust and prone to leading to over-fitting problems.

CNN models provide great insights about

Figure 5 depicts the trends of accuracy, precision, recall and f1-score when the learning rate is increased. We can get best accuracy at learning rate is 0.0001. The recall obtains best recall when we set learning rate 0.001.

Figure 6 shows the effects of hidden layer size on model classification performance. With the size growing, accuracy.., precision..,

After optimizing both models, initial results showed CNN had much greater accuracy. The confusion matrices below allows visualization of performance for the CNN (fig 4)

models by showing percentage of total test data’s predicted label as compared to its ground-truth label.

Using the same CNN network with tuned regularization parameter and learning rate, the greatest accuracy of 98% was achieved for network classification. However, all 6 classes data on the same model yielded average 92.1% accuracy.

We evaluate our proposed deep learning approaches in this section by comparing anomalous instances classification performances achieved by all methods.

In the testing phase, we test the 20,000 events and repeat 10 times to calculate the average performance of CNN classifiers.

Not only the occurrence event is categorized in our experiment, but also the whole duration is classified here. Note that CNN is meant to be applied in the temporal dataset.

There are big differences between the models shown in Table 5. Based on those results, we can see that the model E has the best performance overall, Another interesting trend we can observe is the fact that

Some of the models provide more insights about the performance of models if we look at the more detailed results in table 6. Model B’s F1-score actually comes close to the F1-score of the model C

CNN achieves almost perfect classification performance in both cases, even slightly surpassing the CNN classifier. Figure 2 presents the classification results achieved by CNN on all features. We can conclude that CNN offers an accuracy comparable or even slightly better that that achieved by CNN in all anomalous types.

The classification performance obtained of type database anomalies is slightly worse than that of type file.

It should also be taken into account that there is a certain level of subjectiveness to the manual classification process, which may influence the results. This is caused by the fact that there are no strict

rules that determine whether something is an anomaly. Model E seems to give the best performance of the all models.

## 5 Conclusion

In this paper we have proposed a deep convolutional neural network approach for anomaly event classification on distributed systems, offering a very powerful and straightforward technique to categorize anomalous instances. We generated dataset by extracting semantic information from large scale system log text. We take a series studies to find the proper learning rate and hidden layer size. We achieved an overall accuracy of 98.01%, which shows the potentiality of deep CNN for anomaly classification tasks on distributed systems.

Deep CNN model B is the most successful with 92.6% accuracy

In a nutshell, we believe this appealing approach is capable of providing high insights for understanding system server operations without disclosing any business sensitive information. By depending on deep learning techniques, we have shown the classification performance of the labeled anomalies in an efficient fashion. In general, CNN approach outperforms the CNN method. We will explore better deep learning model for anomaly classification in the future work.

This area is still on-going research, and it requires a lot of work to build the model for the time series. Should you succeed, you may achieve outstanding performance results in terms of accuracy.

In this paper, we have made a comparison between different models to anomaly event classification. We evaluated the performance of these models when applied to an anomaly event task for application log data analysis. It turned out that the relatively model E gave the best performance. However,

The obvious but useful extension to this work would be to extend the experiments to more algorithms to see if there are algorithms available that work even better. With more data could also be useful since that could add the possibility of researching changing trends in the data and how the different algorithms cope with these changes.

However it also faces many challenges.

There are still a few things that need to be done to get this automatic classification running in a production environment.

The novelty of our work lies in the automatic anomaly classification of system events and states from streaming operational logs. Through the evaluation of ..;

## Acknowledgments

This work was supported in part by Institute of Computing Technology, Chinese Academy of Sciences.

## References

[1] Victoria J. Hodge, Jim Austin. A Survey of Outlier Detection Methodologies. Artificial Intelligence Review. 2004; 22(2):85-126. doi:10.1007/s10462-004-4304-y.

[2] Smith M. R., Martinez T. Improving classification accuracy by identifying and removing instances that should be misclassified. The 2011 International Joint Conference on Neural Networks. 2011; p. 2690. ISBN 978-1-4244-9635-8. doi:10.1109/IJCNN.2011.6033571.

[3] Knorr E. M., Ng R. T., Tucakov, V. Distance-based outliers: Algorithms and applications. The VLDB Journal the International Journal on Very Large Data Bases. 2000; 8(3-4):237-253. doi:10.1007/s007780050006.

[4] Breunig M. M., Kriegel H.-P., Ng R. T., Sander, J. LOF: Identifying Density-based Local Outliers. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD. 2000; pp. 93–104. ISBN 1-58113-217-4. doi:10.1145/335191.335388.

[5] Schubert E., Zimek A., Kriegel H. P. Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection. Data Mining and Knowledge Discovery. 2012; 28: 190–237.

doi:10.1007/s10618-012-0300-z.

[6] Kriegel H. P., Kroger P., Schubert E., Zimek A. Outlier Detection in Arbitrarily Oriented Subspaces. 2012 IEEE 12th International Conference on Data Mining. 2012; p.379. ISBN 978-1-4673-4649-8. doi:10.1109/ICDM.2012.21.

[7] He Z., Xu X., Deng S. Discovering cluster-based local outliers. Pattern Recognition Letters. 2003; 24(9-10):1641-1650. doi:10.1016/S0167-8655(03)00003-5.

[8] Campello R. J. G. B., Moulavi D., Zimek A., Sander J. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. ACM Transactions on Knowledge Discovery from Data. 2015; 10(1):5:1-51. doi:10.1145/2733381.

[9] Zimek A., Campello R. J. G. B., Sander J. R. Ensembles for unsupervised outlier detection. ACM SIGKDD Explorations Newsletter. 2014; 15:11–22. doi:10.1145/2594473.2594476.

[10] Zimek A., Campello R. J. G. B., Sander J. R. Data perturbation for outlier detection ensembles. Proceedings of the 26th International Conference on Scientific and Statistical Database Management-SSDBM. 2014; p.1. ISBN 978-1-4503-2722-0. doi:10.1145/2618243.2618257.

[11] Campos Guilherme O., Zimek Arthur, Sander Jörg, Campello Ricardo J. G. B., Micenková Barbora, Schubert Erich, Assent Ira, Houle Michael E. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Mining and Knowledge Discovery. 2016; 30(4):891. ISSN 1384-5810. doi:10.1007/s10618-015-0444-8.

[12] Goldstein M. In: Markus Hofmann RK, editor. Anomaly Detection. Data Mining and Knowledge Discovery Series. Chapman and Hall/CRC. 2013; p.367-394.

[13] Aggarwal CC. Outlier Analysis. Springer-Verlag. NewYork. 2013.

[14] Lazarevic A, Ertoz L, Kumar V, Ozgur A, Srivastava J. A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. In Proceedings of the Third SIAM International Conference on Data Mining. vol. 3. Siam; 2003. p. 25–36.

[15] Ding X, Li Y, Belatreche A, Maguire LP. An Experimental Evaluation of Novelty Detection Methods. Neurocomputing. 2014; 135:313–327. doi:10.1016/j.neucom.2013.12.002

[16] Amer M, Goldstein M, Abdennadher S. Enhancing One-class Support Vector Machines for Unsupervised Anomaly Detection. In: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description (ODD’13). New York, NY, USA: ACM Press. 2013; p. 8-15.

[17] Zimek A, Campello RJGB, Sander J. Ensembles for Unsupervised Outlier Detection: Challenges and Research Questions a Position Paper. SIGKDD Explor Newsl. 2014; 15(1):11-22. doi: 10.1145/2594473.2594476.

[18] Amer M, Goldstein M. Nearest-Neighbor and Clustering based Anomaly Detection Algorithms for RapidMiner. In: Simon Fischer IM, editor. Proceedings of the 3rd RapidMiner Community Meeting and Conferernce (RCOMM 2012). Shaker Verlag GmbH; 2012. p. 1-12.

[19] P. Fiadino et al. RCATool - A Framework for Detecting and Diagnosing Anomalies in Cellular Networks. ITC. 2015.

[20] P. Casas et al. MINETRAC: Mining Flows for Unsupervised Analysis & Semi-Supervised Classification. ITC. 2011.

[21] Greff, Klaus, et al, LSTM: A Search Space Odyssey, arXiv preprint arXiv:1503.04069, 2015.

[22] M. Gupta, J. Gao, C. Aggarwal, and J. Han. Outlier detection for temporal data. Synthesis Lectures on Data Mining and Knowledge Discovery 5.1. Morgan & Claypool Publishers; 2014. pp. 1–129.

[23] E. Schubert, A. Zimek, and H.-P. Kriegel. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. Data Mining and Knowledge Discovery 28.1. Springer; 2014. pp. 190-237.

[24] T. Kimura, K. Ishibashi, T. Mori, H. Sawada, T. Toyono, K. Nishimatsu, A. Watanabe, A. Shimoda, and K. Shiomoto. Spatio-temporal factorization of log data for understanding network events. 2014 IEEE Conference on Computer Communications. INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014; 2014. pp. 610-618.

[25] R. Vaarandi. A data clustering algorithm for mining patterns from event logs,” in In Proceedings of the 2003 IEEE Workshop on IP Operations and Management (IPOM), 2013; pp. 118-126.

[26] J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. CoRR, abs/1412.3128, 2014.5.

[27] M. Henrion, D. J. Hand, A. Gandy, and D. J. Mortlock. CASOS: A subspace method for anomaly detection in high dimensional astronomical databases. Statistical Analysis and Data Mining, 6(1):53–72, 2013.