Indonesian Journal of Electrical Engineering and Computer Science
Vol. 10, No. 1, April 2018, pp. 343~347
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v10.i1.pp343-347

Anomaly Detection in Log Records

Raghav Rastogi, Shreyansh Nahata, Poonam Ghuli, Pratiba D, Dr. Shobha G
Department of Computer Science and Engineering, R.V. College of Engineering, Bengaluru, India

Article Info

Article history:

Received Jan 2, 2018
Revised Mar 9, 2018
Accepted Mar 24, 2018

Keywords:

Anomaly detection
Log analysis
Log records
Neural network

ABSTRACT

In recent times complex software systems are continuously generating application and server logs for the events which had occurred in the past. These generated logs can be utilized for anomaly and intrusion detection. These log files can be used for detecting certain types of abnormalities or exceptions such as spikes in HTTP requests, number of exceptions raised in logs, etc. These types of events recorded in the log files are generally used for anomaly prediction and analysis in future. The proposed prototype for anomaly detection assumes that the log records are uploaded as input using a standard apache log format. Next, a prototype is developed to get the number of HTTP requests for outlier detection. Then anomalies in number of HTTP requests are detected using three techniques namely InterQuartileRange method, Moving averages and Median Absolute deviation. Once the outliers are detected, these outliers are removed from the current dataset. This output is given as input to the Multilayer Perceptron model to predict the number of HTTP requests at the next timestamp. This paper presents a web based model to automate the process of anomaly detection in log files.

Corresponding Author:

Raghav Rastogi,
Department of Computer Science and Engineering,
R.V. College of Engineering, Bengaluru 59, India.

1. INTRODUCTION

A large number of communication logs are generated by corporate systems. These are used to monitor the internet traffic that has been flowing towards a particular source. They produce a large number of logs which are then collected and stored. These logs are scraped to extract the particular point of interest and the associated timestamp which is converted to timeseries dataset. Such type of data is then used by developers and programmers to find any anomalies within the extracted time series dataset. Such anomalies cannot be detected manually. There is a requirement to automate the anomaly detection for which large and complex systems are built. These systems scan the datasets for any anomalies involving any suspicious or interesting part. A large number of traffic anomalies including attacks, large file transfers, flash crowds and outages occur fairly frequently. Large enterprise networks have dedicated security operations for continuously monitoring the network traffic in order to detect, identify and take action on the anomalies that occur in log files. On a smaller scale, these operations are handled by network administrators who can also simultaneously work on other day-to-day maintenance operations and planning activities. Although there has been a recent growth in network monitoring [1], log analysis [2] and intrusion detection systems [3-4], but it is still challenging to correctly detect various types of anomalies. This paper discusses the realization of web based framework to detect outliers in terms of number of HTTP requests for the input IP address using three algorithms namely Inter-Quartile-Range, Moving Averages, Moving median absolute deviation. Thereafter a multilayer perceptron model is built to predict the outliers for a particular time period. This paper focus on three main stages in log analysis namely outlier detection, outlier removal and outlier prediction for the given log file.

2. STATE OF ART DEVELOPMENT

In the last decade lot of work has been carried out on the outlier detection from log records. Recently anomaly detection based systems are also used for cyber-intrusion detection [5]. Anomalies can even be generated in payment security domain. All the payment transaction done generate certain type of log files through which anomalies can be found such as risk factor, fraud detection etc [6].

There are different tools available in market which helps us to detect anomaly. One such tool is loggly [7-8]. Loggly's anomaly detection framework helps in finding fluctuations in event frequency. Anomalies may be of various types such as a spike in number of requests etc. The trend chart plotted by loggly's anomaly detection framework allows choosing any field on which the analysis has to be done. It shows the fluctuations in the frequencies of the choosen fileds. It also shows changes with reapect to current timestamp, the background timestamp and brings the field on top of list with most fluctuations.

Another existing tool is Nagios [9] which also alerts the development team about the various anomalies. Nagios is able to detect different anomalies such as memory usage, disk space usage, port connectivity, whether a process is running or not, etc. For all the anomalies detected it sends out the alert via notification system. In this type of system generally alerts are generated only when a particular service is down. It cannot predict which service may go down in future.

Every day, large amount of enterprise data is generated in the form of time-stamped logs from network devices, security appliances, servers, endpoints, applications, users and so on. The required knowledge to efficiently manage and secure IT infrastructures is hidden in this data, but it's impractical for humans to extract this information. Prelert [10] is an anomaly detection engine. It analyzes the data and detects anomalies. Then relate them together and provides the information about advanced threat activity and any problems related to IT operations. But Prelert is not an open source tool, it is an enterprise application.

2.1. Contribution towards Framework for Anomaly detectIon in Log Records

The contributions towards the development of framework for anomaly detection in log files are listed below:

- The proposed tool provides easy to use web interface where users need to upload the log files.
- For anomaly detection, configurable parameters are provided in the web interface such as window size for median, absolute deviation and moving averages.
- Provision is given to select the IP address for which a user wants to detect and predict anomalies using drop-down menu.
- The prediction models are saved to disk which reduces memory usage. It consists of activation fucntion used, input dimension to neural network etc.
- The proposed tool supports web based interface.

3. PROPOSED WORK

The developed framework for log analysis is currently being used by Cisco; it is developed as a web based application with the following design goals:

- It is light weight framework by supporting minimum dependencies.
- It makes use of micro framework for server side processing.
- It also involves the use of Keras [11], a high-level neural networks API, written in Python and capable of running on top of either TensorFlow or Theano.
- The tool also supports the plotting of anomaly detections as well as predictions with the help of google charts API.

The end user must upload the log file, by processing in the background with the help of Regular Expressions, all the IP addresses contained in the log file are populated in the drop-down menu in the web interface. The user must select the IP for which he wants the outliers to be plotted. Currently the outliers are detected in terms of number of HTTP requests for that IP address. Then user must select the algorithms or methods for detecting the outliers from Inter-Quartile-Range, Moving Averages, Moving median absolute deviation along with their tuning parameters from web interface respectively. After the outlier detection phase the user has the option to train the Multilayer perceptron model for epochs provided by user from web-interface. Once the model is trained, it is persisted to the disk for future reference. Then the user is provided an option to predict the outliers for a particular time period, which is given as input by user through web-interface. This paper realizes the log analysis in three stages. First stage focus on outlier detection, second stage focus on removal of outliers, third stage deals with outlier prediction.

3.1. Outlier Detection

The main objective of this module is to detect the outliers in terms of number of HTTP requests. This module makes use of three algorithms namely Interquartile range, Moving averages and Median Absolute deviation. The Interquartile range (IQR) is a measure of variability. It divides the given dataset into quartiles. It is a measure of statistical depression. These are used to divide the dataset into 4 equal parts called quartiles. The values used to divide each quartile are called the first, second, and third quartiles; and they are denoted by Q1, Q2, and Q3, respectively. Q1 is the "middle" value in the first half of the data set. Q2 is the median value in the given data set. Q3 is the "middle" value in the second half of the data set. To calculate the interquartile one has to compute the difference between Q3 and Q1. For this algorithm the parameter named alpha needs to be selected in the range from 0 to 10. Alpha parameter is required for making the upper and lower quartile normalized to simplify the calculation.

In order to detect outliers in the given one dimensional data the data points are marked and are calculated based on its standard deviations. But the presence of outliers causes a profound effect on the mean and standard deviations and thus the direct use of such a naïve technique is not possible. The median absolute deviation (MAD) is one of the solutions to handle the variability of a univariate sample of quantitative data. For instance, it can also refer to the population parameter. This is estimated by the median absolute deviation which is computed from a sample. For this algorithm window size needs to be adjusted between 0 and 1000.

In the moving average method which is also referred as rolling average or running average, analysis of data points is carried out by generating a series of averages on different subsets of the complete data set. It is a type of finite impulse response filter. Some variations include: simple, and cumulative, or weighted forms (described below).For instance, given a series of numbers and a fixed window size, the first element of the moving average is obtained by taking the average of the initial fixed window size of the number series. Then in the further steps the new average is calculated by moving the window forward. Thereby, the first element is excluded while calculating and this is continued for all such elements. By using one of the methods outliers are plotted with help of google charts API.

3.2. Removal of Outlier

- For the removal of outliers, first the mean of the given dataset is computed. While computing the mean exclude the points marked as anomalous in the previous outlier detection step.
- The mean is calculated using the data collected either from inter-quartile methodology or moving average approach.
- The selection of the median or average can be given as an option to the system which normalizes all the outlier values in correspondence to the given average.
- The outlier data points are then modified and normalized using these values. After this the mean and deviation of the data is recalculated and kept for the analytics purposes.

3.3 Outlier Prediction

The main objective of this module is to predict the outliers with respect to number of HTTP requests corresponding to a given IP address. In prediction module the Multilayer perceptron model is used with the help of keras library. A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. In times series data, the sequence of values is important. First the dataset is split into training and testing dataset which is in the ratio of 7:3. Now the next step is to fit the multilayer perceptron model to the training data. The optimizer being used is RMSPROP (Root Mean Square). After the fitting of data, from test data the values are predicted and plotted using google charts API.

The algorithm takes in a look-back value which defines the number of points used in predicting the value at any point t. The training is done on the dataset and new values can then be forecasted by the system. For training, the user can input the number of epochs (iterations) the model has to run upon to improve efficiency. The loss values for each epoch are also calculated as root mean squared error values and gives an idea to the user about the accuracy of the model on that dataset.

4. RESULTS AND ANALYSIS

The main objective of the discussed tool is to analyze the log files and detect and predict outliers in a very simple way. The visualization of outlier detection and prediction are shown using google charts API. The tool supports web enabled single page application. The advantage of using the current paradigm is that it does not allow the web pages to be rendered from server side. The REST API's are developed for the prototype which sends response as JSON object. The view layer is decoupled from model layer. The view layer only consumes the response in JSON from model layer.

The model is trained on the extracted time series dataset. The x-axis of the graph shown in Figure 1 indicates the increasing timestamp range and the y-axis indicates the number of request hits at that timestamp. The blue region shows the training part of the data and the green part of the graph is used to find the accuracy of the given model by treating it as test data. The proposed model uses RMSprop optimizer as it provides better results by keeping balance between the speed and accuracy of model training. RMSprop divides the learning rate by an exponentially decaying average of squared gradients.

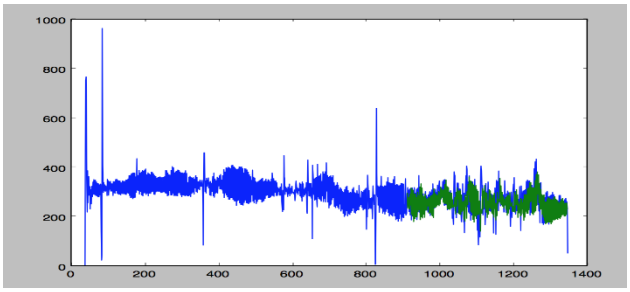

Figure 1. Sample Graph for Predicted Outliers Plotted in Green Color

After the model is trained, it is saved and can be further used for prediction. A sample graph for predicted values is plotted in red color as shown in Figure 2. The x-axis of the graph indicates the increasing timestamp range and the y-axis indicates the number of request hits at that timestamp. The user can input the number of HTTP requests to be forecasted and a graph is displayed in which the green dots gives the original request hits value and the red dots provides the predicted request hits. By using this model the root mean square error was reduced to significant level.

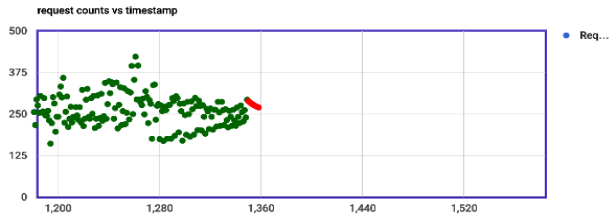

Figure 2. Sample Graph with Predicted Values Plotted in Red Color

5. CONCLUSION

This paper summarizes the design and development of framework for detecting and predicting anomalies in log records in apache format. The current tool provides a simple user interface. The tool successfully plots the outliers detected using three methods namely interquartile ranges, moving averages or median absolute deviation along with the adjustment of the tuning parameters. The tool also gives the user to train the model for a particular number of epochs. After training the model is persisted to disk which reduces the training of model repeatedly. This significantly reduces the CPU utilization time. The predicted values are visualized using google charts API for a given time period provided by user through web interface. The major limitation of this framework is that it supports log files in standard apache format.

ACKNOWLEDGEMENTS

This proposed web enabled framework for anamoly detection is based on the work supported by Cisco India. We thank Cisco Systems for providing an opportunity and support to develop this tool.

REFERENCES

[1] Basavaraj, G. M. "Crowd Anomaly Detection Using Motion Based Spatio-Temporal Feature Analysis." Indonesian Journal of Electrical Engineering and Computer Science, vol 7(3), pp. 737-747, 2017.
[2] Raghav Rastogi, Akash S, Shobha G, Poonam Ghuli, Pratiba D, Ankit Singh, "Design and development of generic web based framework for log analysis", in the proceedings of IEEE *Region 10 Conference (TENCON)*, 2016 pp. 232-236.
[3] Lei, L., Network intrusion detection system based on optimized Fuzzy rules algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, vol 12(4), pp. 2816-2825, 2014.
[4] Liu, L., Wan, P., Wang, Y., & Liu, S., "Clustering and hybrid genetic algorithm based intrusion detection strategy". *Indonesian Journal of Electrical Engineering and Computer Science*, vol 12(1), pp. 762-770, 2014.
[5] Combining Filtering and Statistical Methods for Anomaly Detection. Augustin Soule LIP6-UPMC Kave Salamatian ´ LIP6-UPMC Nina Taft Intel Research.
[6] Shen, Junyuan, and Jidong Wang. "Network intrusion detection by artificial immune system." *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2011, pp. 4716-4720.
[7] AppDynamics White Paper, "A Modern Approach to Monitoring Performance in Production", 2014.
[8] https://www.loggly.com/docs/anomaly-detection/
[9] Josephsen, David. Building a monitoring infrastructure with Nagios. Prentice Hall PTR, 2007.
[10] http://info.prelert.com/
[11] https://keras.io/