International Journal of Engineering & Technology, 7 (4.1) (2018) 47-50

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paperer

Machine Learning for the Communication Optimization in Distributed Systems

Zarina Kazhmaganbetova[1*], Shnar Imangaliyev[1], Altynbek Sharipbay[2]

[1]L.N. Gumilyov Eurasian National University
[2]Research Institute "Artificial Intelligence" of L.N. Gumilyov Eurasian National University
[*]Corresponding author E-mail:z.kazhmaganbetova@gmail.com

Abstract

The objective of the work that is presented in this paper was the problem of the communication optimization and detection of the issues of computing resources performance degradation [1, 2] with the usage of machine learning techniques. Computer networks transmit payload data and the meta-data from numerous sources towards vast number of destinations, especially in multi-tenant environments [3, 4]. Meta data describes the payload data and could be analyzed for anomalies detection in the communication patterns. Communication patterns depend on the payload itself and technical protocol used. The technical patterns are the research target as their analysis could spotlight the vulnerable behavior, for example: unusual traffic, extra load transported and etc.

There was a big data used to train model with a supervised machine learning. Dataset was collected from the network interfaces of the distributed application infrastructure. Machine Learning tools had been retained from the cloud services provider – Amazon Web Services. The stochastic gradient descent technique was utilized for the model training, so that it could represent the communication patterns in the system. The learning target parameter was a packet length, the regression was performed to understand the relationship between packet meta-data (timestamp, protocol, the source server) and its length. The root mean square error calculation was applied to evaluate the learning efficiency. After model was prepared using training dataset, the model was tested with the test dataset and then applied on the target dataset (dataset for prediction) to check whether it was capable to detect anomalies.

The experimental part showed the applicability of machine learning for the communication optimization in the distributed application environment. By means of the trained artificial intelligence model, it was possible to predict target parameters of traffic and computing resources usage with purpose to avoid service degradation. Additionally, one could reveal anomalies in the transferred traffic between application components. The application of techniques is envisioned in information security field and in the field of efficient network resources planning.

Further research could be in application machine learning techniques for more complicated distributed environments and enlarging the number of protocols to prepare communication patterns.

Keywords: *machine learning; artificial intelligence; data communication optimization; distributed network*.

1. Introduction

The machine learning (ML) is a part of the artificial intelligence theory, developed for the resolution of various tasks. The advantage of the machine learning techniques in their applicability for the analysis of big data. Moreover, large volumes of the training sets are necessary for effective training of models, so that it is subsequently possible to make predictions for the target parameter. Target parameter can be category (a line variable) or number (a numerical variable).

The machine learning tools are available at the Amazon Web Services (further – AWS), including the following options of supervised learning [5]:

1. binary classification – reference of a vector of values from set to a class by the principle of identification of belonging value to any category in the truth / false format;
2. classification by multiple categories – reference of a vector of values from set to one of categories in the list;
3. predictive analytics – carrying out the regression analysis of a vector of values for predicting the target numerical parameter.

As a mathematical basis, there are various methodological techniques (decision trees, neural networks, genetic algorithms, Bayesian network) that can be used. Many of them are available through artificial intelligence and machine learning tools at the AWS (Apache MXNet, TensorFlow, Caffe, Theano, Torch, Keras and CNTK).

2. Stochastic Gradient Descent

With purpose to perform the regression analysis using the AWS machine learning tools, method of stochastic gradient descent (Stochastic gradient descent) was applied [6]. When the training model in this method, the problem of minimization of square cost function (cost function) is solved with application of L2 of regularization (Tikhonov's ordering), which is necessary for the big amount of the data set, as follows in formula below (1):

$$f(\mathbf{w}) = 1/\mathrm{m} \sum_{i=1}^{m} \mathrm{E_i}(\mathbf{w})^2 + \lambda \sum_{j=1}^{n} \omega_j^2 \qquad (1)$$

w – values vector, which is estimated for the cost function minimization;
λ – "hyper parameter" or parameter of regularization.

While predicting the target parameter for a vector of input values w in a formula (1), the error is calculated as the difference between expected and real value in a point of xj(i), as follows in the formula (2):

$$\mathrm{E_i}(\mathbf{w}) = \sum_{j=1}^{n} \omega_j x_j^{(i)} - y^{(i)} \qquad (2)$$

Thus, we determine the gradient of the f function of through the formula (3):

$$\mathrm{d}f/\mathrm{d}\omega_j(\mathrm{w}) = (1/\mathrm{m} \sum_{i=1}^{m} 2\mathrm{E_i}(a) \cdot x_j^{(i)}) + \lambda 2\lambda\omega_j \qquad (3)$$

The method of gradient descent is applied for the solution, so that one performs iterative approach to a target value of parameter for the cost function minimization, as follows in the formula (4):

$$\omega_j = \omega_j - \eta\ (2\mathrm{E_i}(a) \cdot x_j + 2\lambda\omega_j) \qquad (4)$$

η – learning rate.

3. Approach to Machine Learning

Approach to machine learning was organised into three stages: dataset preparation, definition of algorithms for learning and model efficiency evaluation through root mean square error calculation, testbed preparation for experiments running.

3.1. Dataset Preparation

To implement machine learning in the format of regression analysis at the AWS, the following steps are required:

1. Placement of analyzed data - to analyze the data it is necessary to upload the initial dataset and the dataset for the usage in the prediction in the AWS environment using the S3 data storage and data management service;

2. Data Separation - the initial set must be divided into two sets, 70% of the set data will be used for training (hereinafter referred to as the training set), 30% for model testing (hereinafter - the test kit);

3. Data Stirring - the data in the dataset should be mixed to exclude rows of the same values;

4. Normalization of data - the data of the training set must be brought to the appropriate formats suitable for analysis;

5. Defining learning parameters - at this step, you need to setup the model parameters for the regularization and the number of iterations on one value vector;

6. Model training - the creation and training of a model on a training set, taking into account the selected training parameters;

7. Testing the model - assessing the effectiveness of the model using a set for testing.

3.2. Model Efficiency Evaluation

In accordance with (2), the model efficiency is evaluated using the root mean square error (RMSE) method according to the formula (5):

$$\mathrm{RMSE} = \ \surd\ (1/\mathrm{n} \sum (y_{target.i} - y_{predicted.i})^2) \qquad (5)$$

$y_{target.i}$ - the actual target value;
$y_{tpredicted.i}$ - the value, that is obtained using trained model.

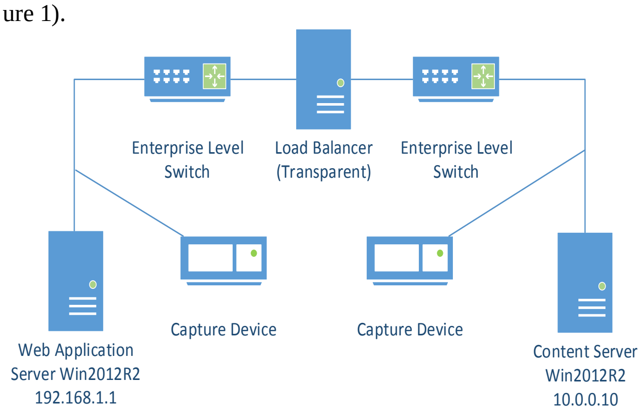
3.3. Problem Statement

The dataset for analysis had been taken from the open source [7] and represents information about the actual network tests in the distributed environment.
Two datasets of CSV data of more than 300 MB each were collected out of the achieved raw data at the Google cloud (UserPcap - 355.5 MB and ServerPcap - 357.7 MB). The raw data from the tests possess the metrics of network traffic, that were recompiled for the analysis purposes.
The problem resides in the distributed application environment resources usage optimization by means of the trained artificial intelligence model. The model could predict the service proper functioning or degradation, including anomalies in application components.

3.4. Experimental Testbed Preparation

During the experimental part of the work, the software automated service from the Amazon Web Services cloud services provider was used in the format machine learning as a service (ML-as-a-Service). This service is useful for applying data analysis on cloud performance, as well as forecasting the load on the network when the distributed application is running.

The experimental testbed with the description is provided in (Figure 1).


Enterprise Level Switch
Load Balancer (Transparent)
Enterprise Level Switch
Capture Device
Capture Device
Web Application Server Win2012R2 192.168.1.1
Content Server Win2012R2 10.0.0.10

Source: *Jasper Bongertz*, The Megalodon Challenge // https://blog.packet-foo.com/2015/07/the-megalodon-challenge/, 26 July 2015

Fig. 1: The experimental testbed. The web application server accesses the content server through the enterprise network, which consists of two switches and transparent load balancer. There are two capture devices installed on per each server.

To be able to use the machine learning tools for predictive analytics in order to generate forecasted values, it is necessary to prepare data. In accordance with the order of steps in Section III, the datasets were uploaded, the original ServerPcap set was divided into a training set and a test set, the data was stirred. At the normalization stage, the parameter formats were checked and the target parameter of size was defined, as shown in the table (Table 1).

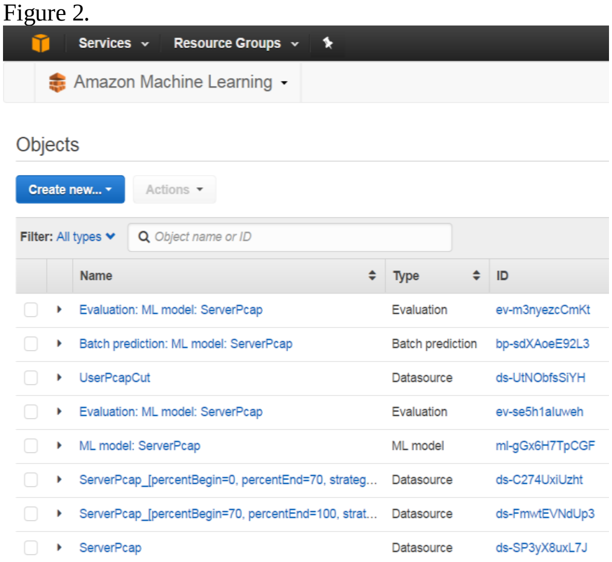
Table 1: Dataset parameters

#	Dataset parameters	
	Name	Parameter definition
1	No.	Sequence number of the protocol packet
2	Time	Timestamp
3	Server	The sender of packets
4	Protocol	Type of network protocol
5	Length (target)	Total packet size

Based on the learning outcomes, the model was tested for efficiency on the training set. Figure 2 shows the results of machine learning:

1. three sets of ServerPcap: source data, training dataset (70%), test dataset (30%);
2. one set for prediction UserPcapCut, a trimmed set of 20,000 packages (value vectors) of the UserPcap set;
3. one model on the original set ML Model: ServerPcap;
4. two test results Evaluation: ServerPcap, the first - on the set for testing, the second - on the set for forecasting;
5. one result of Batch Prediction forecasting on ML Model ServerPcap model, predicted on the UserPcap set.

The Amazon Machine Learning workspace with source datasets and resulting batch predictions and evaluations are provided in Figure 2.



		Name	Type	ID
□	▸	Evaluation: ML model: ServerPcap	Evaluation	ev-m3nyezcCmKt
□	▸	Batch prediction: ML model: ServerPcap	Batch prediction	bp-sdXAoeE92L3
□	▸	UserPcapCut	Datasource	ds-UtNObfsSiYH
□	▸	Evaluation: ML model: ServerPcap	Evaluation	ev-se5h1aluweh
□	▸	ML model: ServerPcap	ML model	ml-gGx6H7TpCGF
□	▸	ServerPcap_[percentBegin=0, percentEnd=70, strateg...	Datasource	ds-C274UxiUzht
□	▸	ServerPcap_[percentBegin=70, percentEnd=100, strat...	Datasource	ds-FmwtEVNdUp3
□	▸	ServerPcap	Datasource	ds-SP3yX8uxL7J

Fig. 2: The Amazon Machine Learning workspace. There source dataset and machine learning resulting outputs in the objects list.

4. Experimental Results

Upon receiving the results of machine learning on the training set, the model was tested using the method of calculating the root-mean-square error using the training set. Performance indicators were as follows: RMSE 19.5, mean error value 701.5, average deviation 682 (Figure 3).

The model was tested for efficiency on the dataset for the usage in the prediction. Performance indicators were: RMSE 23.5, average error value 703.6, mean deviation 680 (Figure 4).

Due to the fact that the purpose of the prediction was to detect anomalies in the sizes of traffic packets, the actual and predicted values were estimated by the deviation size for each packet (Figure 5).

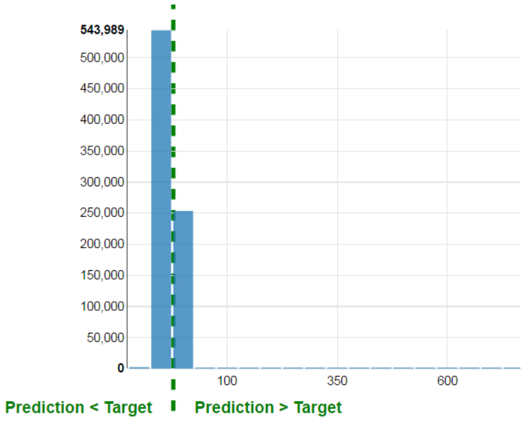


Fig. 3: The results of model testing on prediction. RMSE 19.5, mean error value 701.5, average deviation 682

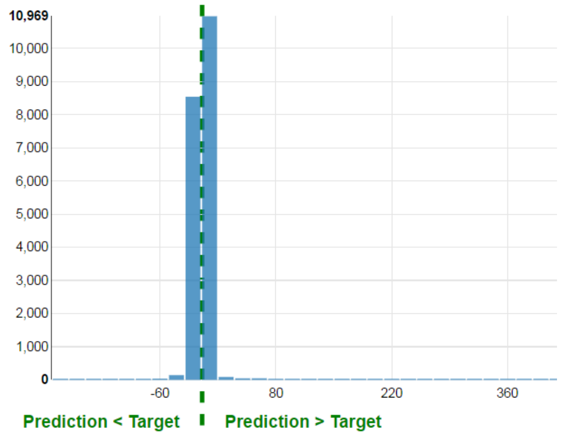


Fig. 4: The results of model testing on prediction. RMSE 23.5, average error value 703.6, mean deviation 680.

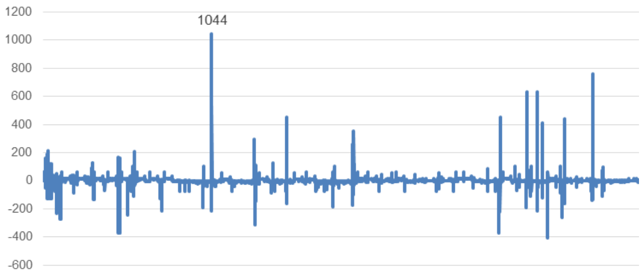


Fig. 5: Prediction error shows the anomalies in the network traffic. Model predicts the packet length that is compared with the actual value afterwards. In the case of largest prediction error, this anomaly is thoroughly reviewed.

In the case of a maximum deviation of 1044 (1514 - valid, 470 - predictive), packages were determined and analyzed.

Thus, with the help of automatic forecasting using the trained model, anomalous moments in the client-server traffic that are subject to optimization were revealed.

5. Conclusion

In this paper, we consider an example of the use of machine learning to solve a practical problem of optimizing the network distributed application components interaction.

The future development of this study is to use a trained model for predicting and detecting anomalies in real time.
Further research could be in the application of machine learning techniques for more complicated distributed environments and

enlarging the number of protocols to prepare communication patterns.

References

[1] A. F. Alam, A. Soltanian, S. Yangui, M. A. Salahuddin, R. Glitho, and H. Elbiaze. “A cloud platform-as-a-service for multimedia conferencing service provisioning”. In 21st IEEE Symposium on Computers and Communications (ISCC), pages 289--294. IEEE, 2016.

[2] E. Amazon. Amazon elastic compute cloud. Retrieved Feb, 10, 2009.

[3] W.-H. Bai, J.-Q. Xi, J.-X. Zhu, and S.-W. Huang. “Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model”. Mathematical Problems in Engineering, 2015, 2015.

[4] W. Li, L. Wu, Y. Xia, Y. Wang, K. Guo, X. Luo, M. Lin, and W. Zheng. On stochastic performance and cost-aware optimal capacity planning of unreliable infrastructure-as-a-service cloud. In Algorithms and Architectures for Parallel Processing, pages 644--657. Springer, 2016.

[5] “Amazon Machine Learning”, https://aws.amazon.com/machine-learning/

[6] Tim Roughgarden, Gregory Valiant, “CS168: The Modern Algorithmic Toolbox Lecture #6: Stochastic Gradient Descent and Regularization”, http://theory.stanford.edu/~tim/s16/l/l6.pdf, Апрель 13, 2016

[7] “The M-Lab”, https://www.measurementlab.net/data/