IEC
Institute of Electronics and Computer

Journal of Artificial Intelligence and Systems, 2019, 1, 133-148
https://iecscience.org/journals/AIS
ISSN Online: 2642-2859

Short time prediction of cloud server round-trip time using a hybrid neuro-fuzzy network

Robertas Damaševičius[1, *], Tatjana Sidekerskienė[2]

[1] Department of Software Engineering, Kaunas University of Technology, Kaunas, Lithuania. Email: robertas.damasevicius@ktu.lt
[2] Department of Applied Mathematics, Kaunas University of Technology, Kaunas, Lithuania. Email: tatjana.sidekerskiene@ktu.lt
*Corresponding Author: Robertas Damaševičius, Email: robertas.damasevicius@ktu.lt

How to cite this paper: Robertas Damaševičius, Tatjana Sidekerskienė (2020). Short time prediction of cloud server round-trip time using a hybrid neuro-fuzzy network. Journal of Artificial Intelligence and Systems, 2, 133–148. https://doi.org/10.33969/AIS.2020.21009.

Received: February29, 2019
Accepted: March 19, 2020
Published: March 23, 2020

Abstract

The paper presents a cloud server roundtrip time prediction approach for cloud datacenters using neuro-fuzzy network with eight probability distribution functions (Normal, Rayleigh, Weibull, Gamma, Birnbaum-Saunders, Extreme Value, and Generalized Pareto) used for fuzzification and defuzzification. We predict the Round-Trip Time (RTT), i.e., the time for a network packet to travel from a client to a server and back. The proposed approach can achieve significant reduction in the short-time RTT prediction error, achieving an accuracy of 79.36%. The approach could be useful for increasing the efficiency of client-cloud systems, for example, when taking effective decisions for computational offloading, and contribute to the development of smart cloud computing.

Keywords

Neuro-fuzzy network, probability distributions; round trip time; Quality of Service (QoS); smart cloud computing.

1. Introduction

Cloud computing provides a versatile and efficient computing environment to support on-demand online services to host client applications based on a principle of pay-on-demand [1]. Cloud is a collection of distributed computing systems consisting of a stack of virtualized and interconnected systems that are dynamically provisioned as computing resource based on a service level agreement (SLA) among the cloud service provider (CSP) and the cloud users [2]. Cloud computing represents an important advancement towards computer-supported cooperative work (CSCW) [3], providing an effective and trustworthy computing framework for

collaborative computation task executions [4] and infrastructure for collaborative enterprises [5] and establishing collaborative networked organizations [6]. The benefits of cloud computing stem from the fundamental idea that CSCW can be enabled by arranging and using distributed resources which may be controlled by different hosts.

Cloud computing has fundamentally shifted the landscape of computing. Despite many advantages, it suffers from the problems of dynamic resource scaling [7]. These factors can make a client-cloud system to become inefficient. Prediction of network traffic parameters such as Round-Trip Time (RTT) can be used to improve the efficiency of client-cloud systems by enabling crucial decisions such as computational offloading or message forwarding [8]. The problem is still relevant as current approaches such as virtualization for sharing resources among users via virtual machines (VM), fall behind providing efficient results. Aiming to maximize the efficiency of sharing and maintaining the high level of the quality of services (QoS), one needs effective resource scaling policies based on analyzing historical data and predicting network congestion and workload of available cloud servers in the short-term future.

Current trends show that the demands for cloud-based applications are expected to increase. Therefore, currently available cloud resources may become insufficient. This motivates the need for outsourcing virtual machines and storage capabilities to other cloud service providers. A Cloud Federation has emerged as platform for upgrading resource scaling strategies on clouds [9]. Interclouds address the limitations imposed by the single-provider policy such as the lack of interoperability between platforms, limited availability resources on peak time, and QoS degradation [10]. Such federated clouds, interclouds, or cloud-of-clouds, is a kind of System of Systems designed to fill the highly increasingly complex business needs for agility, dynamic adaptivity and resource scalability [11] to have higher, reliability, better QoS and flexibility. A smart cloud computing environment for just in time opportunities and scalable provisions of application services, and achieving QoS targets under variable workloads, resource and networks conditions is discussed in [12]. Resource scaling may depend on multiple factors such as the number of clients, network congestions, server usage patterns, and many others. Reactive scaling is not useful because of changing characteristics of network and cloud server workload. However, one can take the proactive approach where the performance limiting factors such as network traffic can be forecasted from historical data and the current state of the network connection.

The inclusion of predictive analytics to interconnected cloud systems contributes to the achievement of the smart cloud computing vision [13], in which providing

autonomic adaptation services based on artificial intelligence (AI) methods is key for developing large-scale complex Cyber Physical Systems (CPS). The existing global cloud computing infrastructure must be managed intelligently in order to ensure its efficiency and sustainability [14], and robustness in case of technological failures or terrorist attacks. In Smart Cloud computing, services may require to be relocated to other cloud servers due to many reasons, which are difficult to foresee in order to ensure continuity of the provided services, dynamic scaling for unexpectedly high number of incoming requests to overcome restrictions on services availability imposed by unfriendly actors.

The key challenges for accurate forecasting of cloud computing related resource utilization are communications with ever-changing number of clients and non-linearity of workload. Several methods can be applied to forecast resource utilization such as maximum or average values. However, when using the maximum workload forecasting model, the resources will stay underutilized for most of the time. If the average workload model is used, then the system will face the shortage of resources leading to performance degradation [15]. Machine learning methods which use historical data are commonly used for more accurate forecasting [16]. For example, Huang et al. [17] use Recurrent neural network (RNN) with long short-term memory (LSTM) units to forecast server performance and load. Lu et al. [18] use RVLBPNN (Rand Variable Learning Rate Backpropagation Neural Network), which allowed to achieve an accuracy of 61.71% in estimating memory-intensive workloads. Kumar et al. [15] use self-adaptive differential evolution for optimization of a neural network, which can forecast workloads. Ci et al. [19] used the Long Short-Term Memory Network (LSTM) to predict network traffic flow. Shen et al. [20] used temporal clustering analysis combined with deformable convolution neural network for network traffic speed prediction. Xu et al. [21] proposed wavelet neural network based on mind evolutionary algorithm (MEA-WNN) for a short-term traffic flow prediction. Tuli and Kumar [22] used artificial neural network (ANN) to forecast packet delay in a mobile ad hoc network (MANET). Aibin [23] used ANN to predict network traffic characteristics in elastic optical networks. Yasuda and Yoshida [24] proposed a two-state Markov process based method to predict the probability density function (PDF) of the round-trip time (RTT) in a mobile/wireless network.

Recently, the hybrid methods were successfully adopted for handling time-series data and time-periodic events [25] as well as for supervised learning [26]. For example, a neuro-fuzzy machine learning was used for classifying medical data in the context of mobile cloud computing environment [27]. Xu et al. [28] applied a hybrid nonlinear auto-regressive with exogenous - random forest (NARX-RF)

model to predict traffic on a cellular network during holidays.

In this paper we propose a neuro-fuzzy prediction framework for the near-time prediction of the cloud server response time. Our contribution is (1) the adoption of statistical probability distribution functions used for fuzzification of network response data, which has not been done before; (2) and the proposed of a hybrid neuro-fuzzy method, which can be used to effectively model and predict the round-trip time of network packets.

2. Method

2.1. Preliminaries

Computational off-loading is a kind of opportunistic computing [29] that promises to provide performance-efficiency of client-cloud systems, especially for mobile applications. We adopt the opportunistic strategy for computation offloading as proposed in [30]. To efficiently manage the cloud off-loading, the system architecture has services implemented to perform the monitoring of off-loading and appropriately schedule the computation tasks. The Intelligent Offloading Management Module (see Figure 1) deployed on a resource stricken mobile device manages the CPU-hungry computations and performs the offloading of costly functions onto the cloud server. It consists of ANN that analyzes network speed and forecast the short time ahead network speed. Based on these results, the smart offloading decision is taken.

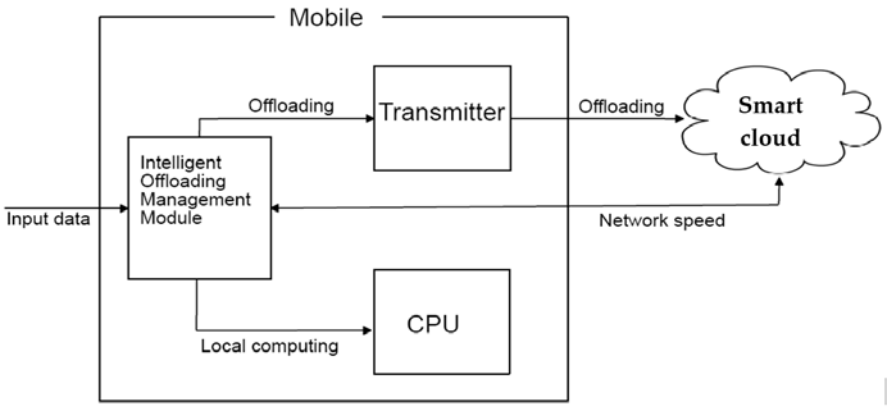

Figure 1. Intelligent computation offloading management

2.2. Neuro-fuzzy prediction

We apply the neuro-fuzzy approach [31] for prediction. The integration of neural networks and fuzzy reasoning allows to utilize the advantages of both approaches simultaneously. The approach is summarized in Figure 2. The method uses

historical network traffic delay data to train an artificial neural network to predict the network delay (RTT) in the future. Finally, the results are evaluated.

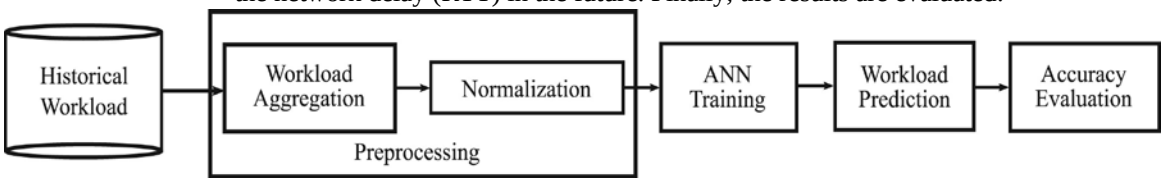

Figure 2. Summary of the neuro-fuzzy workload prediction

2.2. Fuzzification of data

Fuzzy Logic uses a "soft" linguistic system of variables and a continuous range of truth values introduced by Zadeh [32] to deal with uncertainty problems. Here we adopt a variant of fuzzy logic, called soft logic [33]. Let U be an initial universe set and let E be a set of parameters. Pair (F, E) is named a soft set over U if and only if F is a mapping of E into the set of all subsets of the set U, i.e. $F: E \to P(U)$, where $P(U)$ is the power set of U. Every set $F(e)$, for $e \in E$, from this family may be considered as the set of e-elements of the soft set (F, E).

For the fuzzification stage, we use the probability distribution functions (PDF) of the following statistical distributions: Normal, Rayleigh, Weibull, Gamma, Birnbaum-Saunders, Extreme Value, and Generalized Pareto, which are depicted visually in Figure 3. The normal (Gaussian) distribution is commonly used in natural sciences, and is described by the following PDF:

$$p(x) = \frac{1}{\rho\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad (1)$$

here μ is the mean and ρ is standard deviation of values.

The Rayleigh distribution is observed in the directional vector data, and its PDF is:

$$p(x) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}. \quad (2)$$

The Weibull distribution is used for survival and failure analysis with this PDF:

$$p(x) = \frac{k}{\eta}\left(\frac{x}{\eta}\right)^{k-1} e^{-\left(\frac{x}{\eta}\right)^k}, \quad (3)$$

here $k > 0$ is the shape parameter and η >0 is the scale parameter.

The Gamma distribution is characterized by the following PDF:

$$p_{GG}(x, a, b, p) = \frac{|a|}{\beta\Gamma(p)}\left(\frac{x}{b}\right)^{ap-1} e^{-\left(\frac{x}{b}\right)^a}, \quad (4)$$

here a is the shape parameter and b is the rate parameter.

The Birnbaum-Saunders distribution is used to simulate the failure times in reliability applications, and is described by this PDF:

$$p(x) = \frac{1}{2\sqrt{2\pi}\alpha\beta}\left(\left(\frac{x}{\beta}\right)^{\frac{1}{2}} + \left(\frac{\beta}{x}\right)^{\frac{3}{2}}\right) exp\left(-\frac{1}{2\alpha^2}\left(\frac{x}{\beta} + \frac{\beta}{x} - 2\right)\right), \quad (5)$$

here α is the shape parameter and β is the scale parameter.

The extreme value distribution is often used for assessing various financial risks or various extreme climate events and has such PDF:

$$p(x) = e^{-(1+\xi(x-\mu)/\sigma)^{-1/\xi}}. \quad (6)$$

here ξ is the shape parameter.

The Pareto distribution is characterized by the following PDF:

$$p(x) = \frac{\alpha(x_m)^{\alpha}}{x^{\alpha+1}}, \quad (7)$$

here x_m is the minimum possible value of x, and α is a positive parameter.

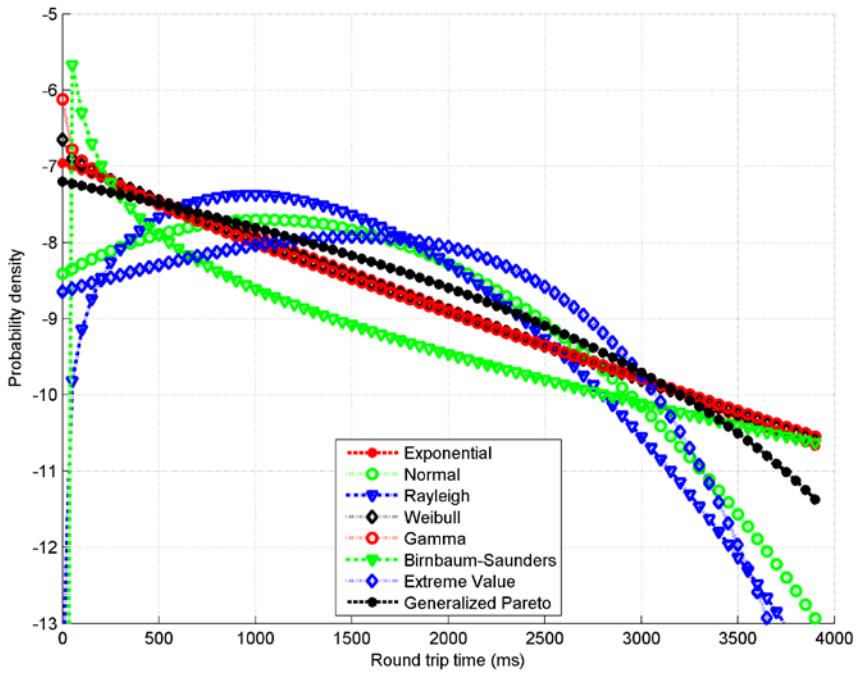

Figure 3. Probability distribution functions of analyzed statistical distributions

2.3. Neural network

The neural network architecture processes the probability values from the network traffic for short term prediction. In the proposed architecture (Figure 4) we have used an idea of Radial Basis Functions (RBF) on the input composed with classic Sigmoid Bipolar Functions on other layers.

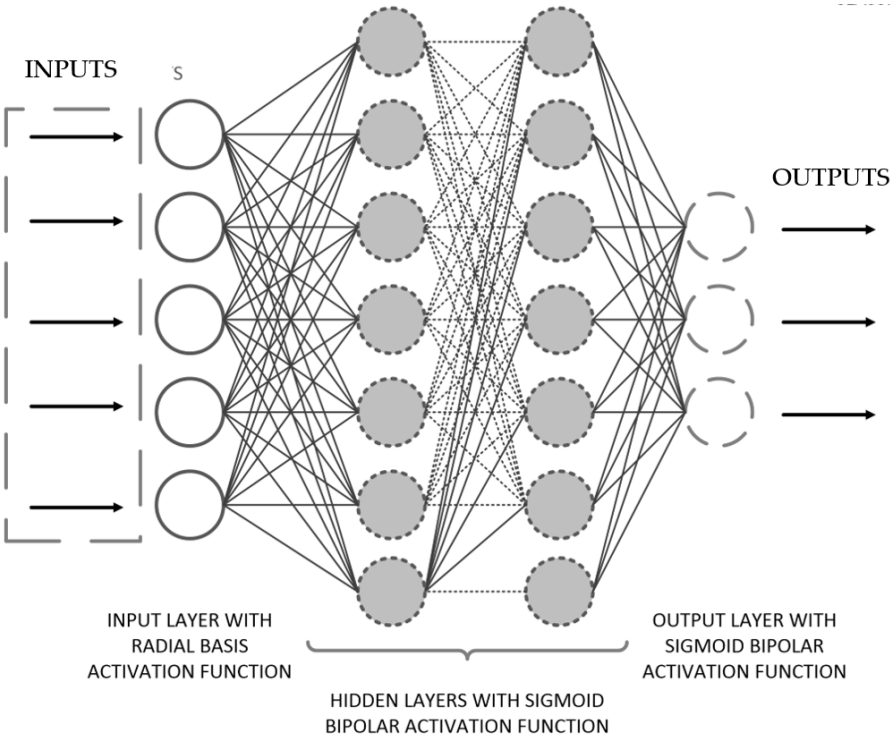

Figure 4. Architecture of the neural network

The novelty of our solution is in the form of the input to the neural network. We have introduced fuzzy weights to all the input descriptors to diversify the importance of the input information on each input unit. On the input, the neurons apply the Radial Basis Function (RBF) as follows:

$$F(\cdot) = exp\left(\frac{||input||}{\lambda^2}\right) \tag{8}$$

where λ is the distribution spread and $||input||$ are the inputs.

The signal received by neurons in the subsequent layers is computed as

$$s_i^k = \sum_{j=1}^{N^{k-1}} y_j^{k-1} w_{ij}^k, \tag{9}$$

where N^k is the count of neurons in layer k, w_{ij}^k is weight of impulse between k and $k+1$ layers from unit j to unit i and y_j^{k-1} is the impulse. Received signal is scaled on the neurons by applying the activation function

$$\begin{aligned} f\left(s_i^k\right) = & \quad f\left(\sum_{j=1}^{N^{k-1}} y_j^{k-1} w_{ij}^k\right) = \\ & = \frac{1-exp\left(-\beta \sum_{j=1}^{N^{k-1}} y_j^{k-1} w_{ij}^k\right)}{1+exp\left(-\beta \sum_{j=1}^{N^{k-1}} y_j^{k-1} w_{ij}^k\right)} = \\ & = \frac{1-exp(-\beta s_i^k)}{1+exp(-\beta s_i^k)} \end{aligned} \tag{10}$$

where β is the correction coefficient for controlling the speed of convergence.

The training procedure minimizes the mean square error function

$$B(T) = \sum_T \quad \varepsilon^2(x) = \sum_{i=1}^{N_t} \quad \varepsilon_i^k = \sum_{i=1}^{N_t} \quad \left(d_i^k - y_i^k\right)^2 \tag{11}$$

for all items $n \in N_t$ in the training set T, by calculating error gradient value

$$\begin{aligned} \nabla_{ij}^k &= \frac{\partial B(T)}{\partial w_{ij}^k} = \frac{\partial B(T)}{\partial s_i^k}\frac{\partial s_i^k}{\partial w_{ij}^k} = \\ &= \left(-\frac{1}{2}\right)\frac{\partial B(T)}{\partial s_i^k}(-2)\frac{\partial s_i^k}{\partial w_{ij}^k} = -2\delta_i^k\frac{\partial s_i^k}{\partial w_{ij}^k} = -2\delta_i^k x_j^k \end{aligned} \tag{12}$$

and used to scale weights w_{ij}^k in each iteration according to

$$w_{ij}^k \leftarrow w_{ij}^k |d_i^k - y_i^k| + 2\eta\delta_i^k x_j^k, \tag{13}$$

where η is the training coefficient, $\delta_i^k = -\frac{1}{2}\frac{\partial B(T)}{\partial s_i^k}$ is error function change, and x_j^k is the input signal.

2.4. Data preparation algorithm for network

To prepare the data for the neural network, the following algorithm is applied:

1. Perform time windowing of data by selecting an appropriate window size w.
2. Perform the fitting of distribution of data in each window to eight probability distribution functions discussed in section 2.2.
3. Perform logarithmical (basis 2) binning of data and calculate the number of data points falling to each bin from each of fitted probability distributions.
4. Used the obtained values as a feature vector with the length of $8b$ (where b is the number of bins) for network training.

3. Experiments and results

3.1. Data acquisition

The performance of the proposed method was evaluated using the real-world network traffic data collected during the cloud-based game sessions. The game is based on the use of virtual physics, therefore expensive computations are needed, which are offloaded to a game server (more details about the dataset can be found in [34]). The example values of data with its statistical distribution are shown in Figure 5 and analyzed in Figure 6. The distribution of data has a heavy tail as evidence by its cumulative distribution function (CDF), while the logarithmically transformed data shows that it has a bimodal nature. The 70% of data are used for training, 15% - for validation, and the remaining 15% - for testing. Processing of data, the neural network and visualization of results was implemented in MATLAB (R2019a) on ASUS computer with Intel i5-8265U 1.60GHz CPU and 8GB RAM running Windows 10 64-bit operating system.

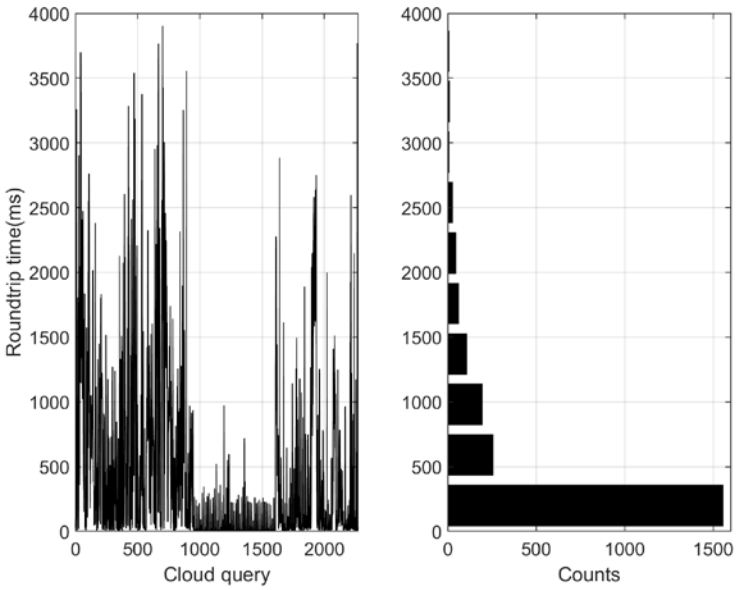

Figure 5. Example of roundtrip delay data (right) with its histogram (left)

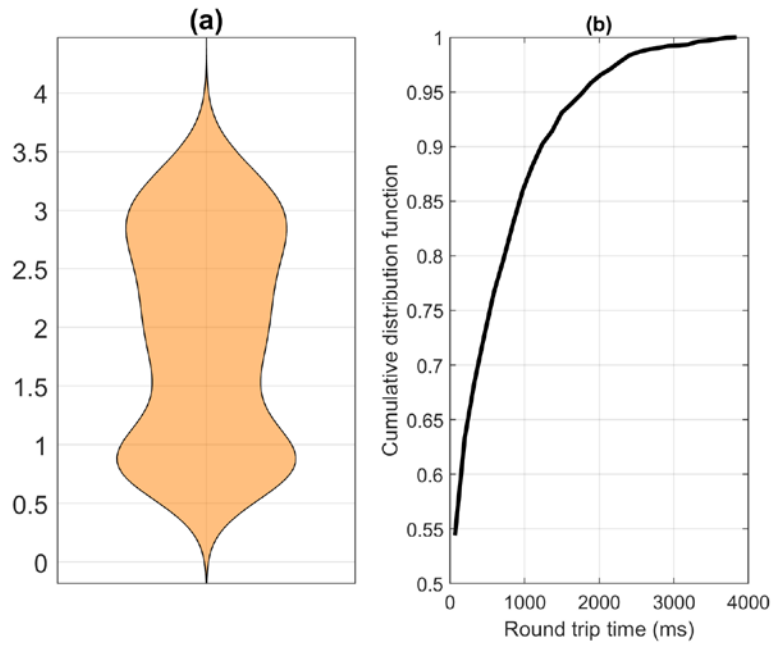

Figure 6. Violin plot showing the statistical distribution of RTT data after logarithmic transform (a), and the cumulative distribution function of RTT data (b)

3.2. Results

The logarithmic transform was applied to the network delay data before

commencing the training of the proposed neuro-fuzzy network. The training of the network model is fast (only 1.94 s). The network prediction power is evaluated in Figure 7, which presents a histogram of both training, validation and testing errors. The results show that most of errors have a value which is close to 0 indicating that the network has learned the data well.

Figure 7. Error histogram of neural network predictions

The accuracy of the network was computed as the number of accurately predicted bins to which the RTT value was assigned. As a result of testing, we achieved an accuracy of 80.2% for the one-step look-ahead case. The confusion matrix of the prediction results is presented in Figure 8.

In Figure 9 we present an example of 1-step look-ahead predicted bin values with respect to true bin values of RTT in the analyzed network dataset. The background of the figure shows the expected probability of the round-trip time, while the markers show the true and predicted bin of round-trip time, respectively.

To test the performance of the hybrid network model of its ability to predict the values of RTT longer in the future, we trained a network to perform prediction of the L={2..20} steps in the future and repeated each experiment 10 times. The results are presented in Figure 10. Note that there is no noticeable downwards trend meaning that the network model has a good ability to generalize on data, while the grand mean of all experiments is 79.36%.

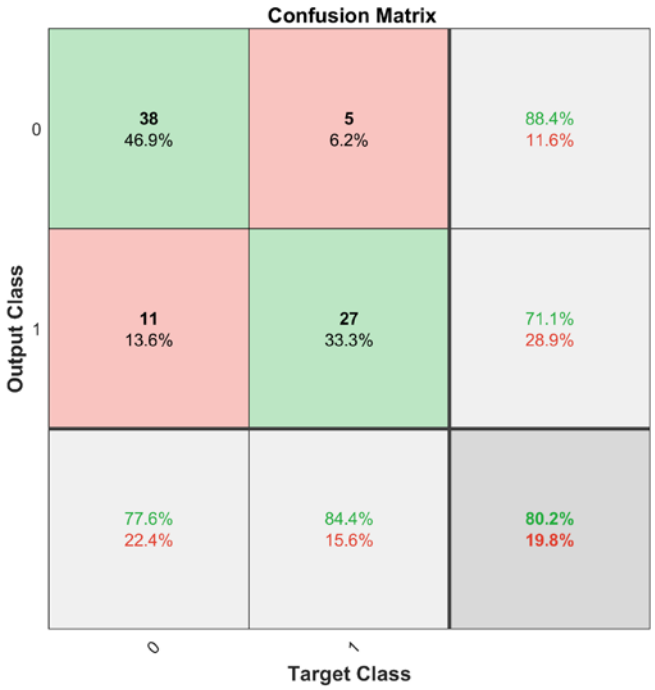

Figure 8. Confusion matrix of one-step look-ahead prediction results

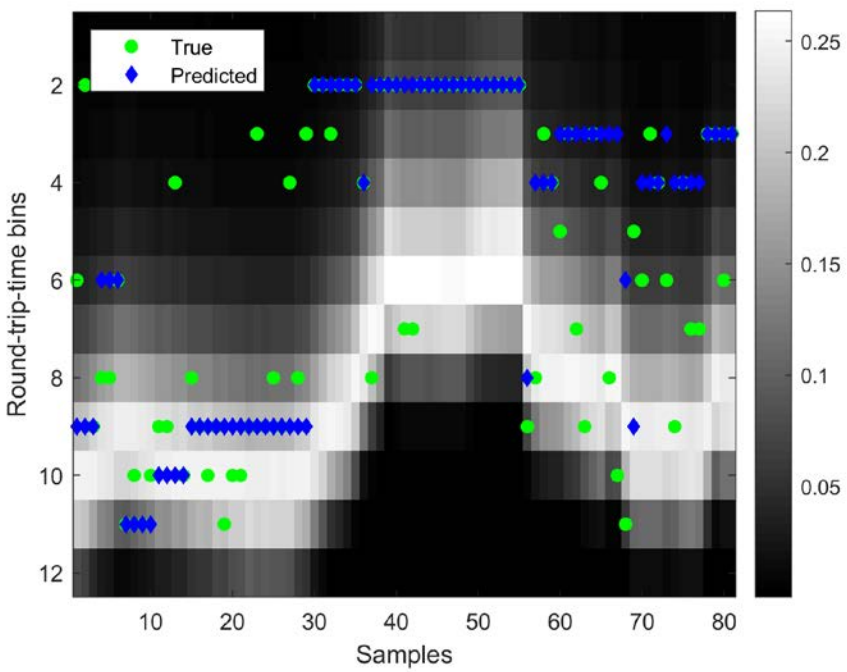

Figure 9. Example of 1-step look-ahead prediction values of rount-trip time bins

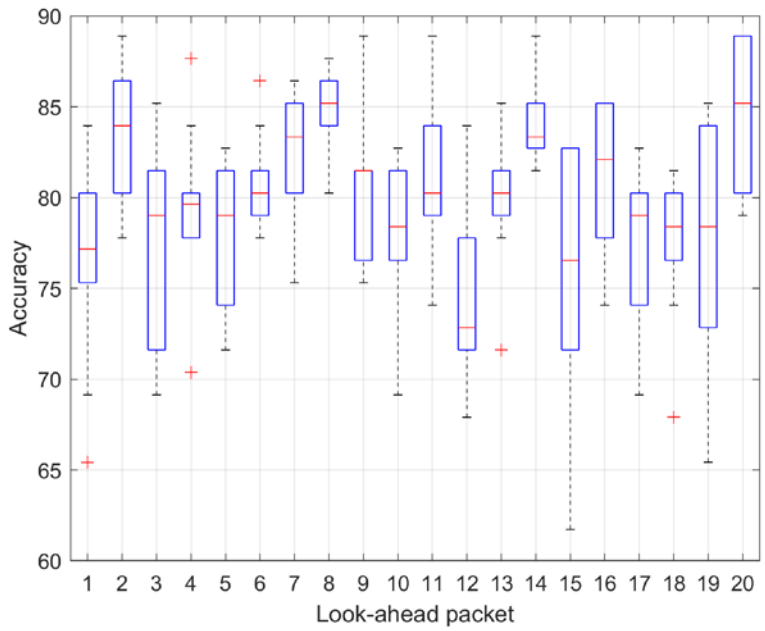

Figure 10. Results of multi-step look-ahead prediction

3.3. Evaluation and comparison of results

We compare our results with other authors, who also work on prediction the round-trip time in computer networks in Table 1. Note that Yasuda and Yoshida [24] predicted only the distribution characteristics of the RTT values rather than actual values. However, the different datasets used and the different methodologies for evaluation of results do not allow for direct comparison of results. Perhaps the most similar approach to the proposed in this paper is presented by Khatouni et al. in [35]. They also divided the values of the 4G network latency into the bins thus turning the latency forecasting problem to a multilabel classification problem.

Table 1. Comparison of results with other methods

Method	Performance*	Reference
Random Forest	0.71 (F-score)	Khatouni et al [35]
Recurrent neural networks	1.534 (RMSE)	Dong et al [36]
Convolutional neural network	0.05 (RMSE) 0.591(NPRE)	Mohammed et al. [37]
Two-state Markov process	na	Yasuda and Yoshida [24]
Adaptive matrix completion algorithm	na	Tripathi and Rajawat [38]
Three-layer ANN	0.2 (NMSE)	Iqbal et al [39]
Neural network with fuzzified inputs	79.36% (accuracy)	This paper

* NPRE - ninetieth percentile relative error. RMSE – Root Mean Square Error.

NMSE - normalized mean square error.

4. Conclusions

In this paper we have presented a hybrid neuro-fuzzy approach for client-cloud server communication round-trip time (RTT) prediction. We used eight probability distribution functions (Normal, Rayleigh, Weibull, Gamma, Birnbaum-Saunders, Extreme Value, and Generalized Pareto) used for fuzzification of data. Thus we converted the regression problem into the classification problem, for which we used a Radial Basis Function Neural Network. We have tested the approach on our own dataset collected during the cloud-based game session using a cloud-based distributed infrastructure. The proposed approach achieved a grand mean accuracy of 79.36% in predicting the short-time RTT. The approach can be used to increase the efficiency of client-cloud communications, when taking effective decisions for increasing Quality of Service (QoS) and performing computational offloading, for example, in case of power-demanding mobile applications, and contribute to the development of smart cloud computing infrastructure.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] S. S. Gill, R. Buyya (2017). A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View. CoRR abs/1712.02899

[2] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic (2009). Cloud computing and emerging IT platforms: Vision, hype and reality for delivering computing as the 5th Utility. Future Generation Computer Systems, 25, 599-616.

[3] Jiang J., Yang G. (2010) Examining Cloud Computing from the Perspective of Grid and Computer-Supported Cooperative Work. In: Antonopoulos N., Gillam L. (eds) Cloud Computing. Computer Communications and Networks. Springer, London.

[4] Chen, X., Zhou, Z., Wu, W., Wu, D., Zhang, J. (2018). Socially-motivated cooperative mobile edge computing. IEEE Network, doi:10.1109/MNET.2018.1700354

[5] Fugini, M. G., & Cellary, W. (2017). Enabling technologies: Infrastructure for collabo-rative enterprises: 2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2017, xi-xiv. doi:10.1109/WETICE.2017.39

[6] Mollahoseini Ardakani, M. R., Hashemi, S. M., & Razzazi, M. (2018). A cloud-based solution/reference architecture for establishing collaborative networked organizations. Journal of Intelligent Manufacturing, 1-17. doi:10.1007/s10845-017-1387-2

[7] Jonathan, O., Misra, S., Ibanga, E., Maskeliunas, R., Damasevicius, R., & Ahuja, R. (2019). Design and implementation of a mobile webcast application with google analytics and cloud messaging functionality. Journal of Physics: Conference Series, , 1235(1) doi:10.1088/1742-6596/1235/1/012023

[8] Odun-Ayo, I., Geteloma, V., Misra, S., Ahuja, R., & Damasevicius, R. (2020). Systematic mapping study of utility-driven platforms for clouds. Proceedings of ICETIT 2019 pp 762-774. doi:10.1007/978-3-030-30577-2_68

[9] Toosi, A. N., Calheiros, R. N., & Buyya, R. (2014). Interconnected cloud computing environments: Challenges, taxonomy, and survey. ACM Computing Surveys, 47(1) doi:10.1145/2593512

[10] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, M. Morrow (2009). Blueprint for the intercloud - protocols and formats for cloud computing interoperability. In 4th International Conference on Internet and Web Applications and Services, 328–336.

[11] Mathlouthi, W., & Saoud, N. B. B. (2017). Flexible composition of system of systems on cloud federation. IEEE 5th International Conference on Future Internet of Things and Cloud, FiCloud 2017, 358-365. doi:10.1109/FiCloud.2017.18

[12] Buyya, R., Ranjan, R., & Calheiros, R. N. (2010). InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services. In Algorithms and Architectures for Parallel Processing (pp. 13–31). Springer Berlin Heidelberg. Doi:10.1007/978-3-642-13119-6_2

[13] Getov V., Srinivasan S. (2011). From Invisible Grids to Smart Cloud Computing. In: Euro-Par 2010 Parallel Processing Workshops. Lecture Notes in Computer Science, vol 6586, pp. 263-270. Springer, Berlin, Heidelberg.

[14] Müller, G., Sonehara, N., Echizen, I., & Wohlgemuth, S. (2011). Sustainable cloud computing. Business and Information Systems Engineering, 3(3), 129-131. doi:10.1007/s12599-011-0159-3

[15] J. Kumar, and A. K. Singh, (2018). Workload prediction in cloud using artificial neural network and adaptive differential evolution. Future Generation Computer Systems, vol. 81, pp. 41–52.

[16] Cetinski, K., & Juric, M. B. (2015). AME-WPC: Advanced model for efficient workload prediction in the cloud. Journal of Network and Computer Applications, 55, 191–201. doi:10.1016/j.jnca.2015.06.001

[17] Z. Huang, J. Peng, H. Lian, J. Guo, and W. Qiu (2017) Deep Recurrent Model for Server Load and Performance Prediction in Data Center. Complexity, vol. 2017, Article ID 8584252, 10 p. doi:10.1155/2017/8584252.

[18] Y. Lu, J. Panneerselvam, L. Liu, and Y. Wu (2016). RVLBPNN: A Workload Forecasting Model for Smart Cloud Computing. Scientific Programming, vol. 2016, Article ID 5635673, 9 p. doi:10.1155/2016/5635673.

[19] Ci, Y., Xiu, G., & Wu, L. (2019). A short-term traffic flow prediction method based on long short-term memory network. In Green Intelligent Transportation Systems, Lecture Notes in Electrical Engineering 503, doi:10.1007/978-981-13-0302-9_59

[20] Shen, G., Chen, C., Pan, Q., Shen, S., & Liu, Z. (2018). Research on traffic speed prediction by temporal clustering analysis and convolutional neural

network with deformable kernels. IEEE Access, 6, 51756-51765. doi:10.1109/ACCESS.2018.2868735

[21] Xu, L., Du, X., & Wang, B. (2018). Short-term traffic flow prediction model of wavelet neural network based on mind evolutionary algorithm. International Journal of Pattern Recognition and Artificial Intelligence, 32(12) doi:10.1142/S0218001418500416

[22] Tuli, H., & Kumar, S. (2018). Packet delay prediction in MANET using artificial neural network. In: Lobiyal D., Mansotra V., Singh U. (eds) Next-Generation Networks. Advances in Intelligent Systems and Computing, vol 638, pp 369-375. Springer, Singapore. doi:10.1007/978-981-10-6005-2_39

[23] Aibin, M. (2018). Traffic prediction based on machine learning for elastic optical networks. Optical Switching and Networking, 30, 33-39.

[24] Yasuda, S., & Yoshida, H. (2018). Prediction of round trip delay for wireless networks by a two-state model. IEEE Wireless Communications and Networking Conference, WCNC, 1-6. doi:10.1109/WCNC.2018.8377039

[25] Selvachandran, G., Quek, S. G., Lan, L. T. H., Son, L. H., Long Giang, N., Ding, W., Albuquerque, V. H. C. (2019). A New Design of Mamdani Complex Fuzzy Inference System for Multi-attribute Decision Making Problems. IEEE Transactions on Fuzzy Systems, 1–1. doi:10.1109/tfuzz.2019.2961350

[26] De Souza, R. W. R., De Oliveira, J. V. C., Passos, L. A., Ding, W., Papa, J. P., & Albuquerque, V. (2019). A Novel Approach for Optimum-Path Forest Classification Using Fuzzy Logic. IEEE Transactions on Fuzzy Systems, 1–1. doi:10.1109/tfuzz.2019.2949771

[27] Moreira, M. W. L., Rodrigues, J. J. P. C., Al-Muhtadi, J., Korotaev, V. V., & de Albuquerque, V. H. C. (2018). Neuro-fuzzy model for HELLP syndrome prediction in mobile cloud computing environments. Concurrency and Computation: Practice and Experience, e4651. doi:10.1002/cpe.4651

[28] Xu, M., Wang, Q., & Lin, Q. (2018). Hybrid holiday traffic predictions in cellular networks. IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018, 1-6. doi:10.1109/NOMS.2018.8406291

[29] Witkowski, M., Brenner, P., Jansen, R., Go, D. B., & Ward, E. (2010). Enabling sustainable clouds via environmentally opportunistic computing. 2nd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2010, 587-592. doi:10.1109/CloudCom.2010.111

[30] Li, W., You, X., Jiang, Y., Yang, J., & Hu, L. (2019). Opportunistic computing offloading in edge clouds. Journal of Parallel and Distributed Computing, 123, 69-76. doi:10.1016/j.jpdc.2018.09.006

[31] Georgy, M. E., Chang, L.-M., & Zhang, L. (2005). Prediction of Engineering Performance: A Neurofuzzy Approach. Journal of Construction Engineering and Management, 131(5), 548–557.

[32] Zadeh, L.A.: Fuzzy sets. Inform. Control. 8 (1965) 338-353.

[33] D. Molodstov, Soft set theory-first results, Computers Math. Applic. 37 (4/5), (1999), 19-31.

[34] Danevičius, E.; Maskeliūnas, R.; Damaševičius, R.; Połap, D.; Woźniak, M.

(2018) A Soft Body Physics Simulator with Computational Offloading to the Cloud. Information, 9, 318. Doi: 10.3390/info9120318

[35] A. S. Khatouni, F. Soro and D. Giordano, "A Machine Learning Application for Latency Prediction in Operational 4G Networks," 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, 2019, pp. 71-74.

[36] Dong, A., Du, Z., & Yan, Z. (2019). Round trip time prediction using recurrent neural networks with minimal gated unit. IEEE Communications Letters, 23(4), 584-587. doi:10.1109/LCOMM.2019.2899603

[37] S. A. Mohammed, S. Shirmohammadi and S. Altamimi, "Artificial Intelligence-Based Distributed Network Latency Measurement," 2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Auckland, New Zealand, 2019, pp. 1-6.

[38] R. Tripathi and K. Rajawat, "Dynamic Network Latency Prediction with Adaptive Matrix Completion," 2018 International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 2018, pp. 407-411. doi: 10.1109/SPCOM.2018.8724422

[39] Iqbal, M. F., Zahid, M., Habib, D., & John, L. K. (2019). Efficient Prediction of Network Traffic for Real-Time Applications. Journal of Computer Networks and Communications, 2019, 1–11. Doi: 10.1155/2019/4067135