Deep Neural Network-based Approach for IoT Service QoS Predicti

Chrisson Awanyo, Nawal Guermouche

Deep Neural Network-based Approach for IoT Service QoS Prediction

Christson Awanyo[1] and Nawal Guermouche[1]

LAAS-CNRS, University of Toulouse, INSA, Toulouse, France
{kjbc.awanyo,nguermou}@laas.fr

Abstract. Building innovative and complex applications on top of the Internet of Things (IoT) services provided by huge connected devices and software while satisfying quality of service (QoS) parameters has become a challenging topic. Identifying suitable services according to their QoS parameters is one of the main underlying features to enable optimal selection, composition, and self-management of IoT systems. Checking each service to get its accurate QoS is not feasible. QoS prediction has been proposed these last years to try to cope with this issue. Mainly, the existing approaches rely on collaborative filtering methods, which suffer from scalability issues, that can considerably hamper the performance of QoS prediction. To overcome this limit, in this paper, we propose a deep-learning-based QoS prediction approach for IoT services. The approach we propose relies on Long Short-Term Memory (LSTM) to capture the service representation through a service latent vector and on Residual Network (ResNet) for QoS prediction. Unlike existing deep-learning-based approaches that assume a pre-defined static set of services, our approach addresses the QoS prediction problem for dynamic environments where the services are not necessarily fixed in advance.

Keywords: IoT · QoS prediction · Deep Learning · LSTM· ResNet· Smart City

1 Introduction

The Internet of Things (IoT) has emerged as a promising technology to build smart systems on top of distributed and connected physical devices. IoT is increasingly adopted across a broad range of domains, including healthcare, agriculture, transportation, factories, and in general smart cities. With IoT, real-time data can be collected, processed, and analyzed to enable smart decision-making. The Web of Things (WoT) has further propelled theses advances by enabling physical objects to be connected to the Web through their virtual representations [7], enabling them to be discovered and used as IoT services. Although WoT has brought great opportunities to provide new value added services that can leverage a wide range of potentially heterogeneous IoT devices to fulfill complex IoT needs, it also brings several challenges that remain open, such as IoT service discovery, selection and composition. Indeed, IoT environments are highly dynamic

where IoT devices can be mobile and can have limited resources. Moreover, the number of IoT devices is increasing rapidly and can exceed 70 billion by 2025 according to Gartner.

In this context, to select the suitable set of IoT services that meet awaited quality of service (QoS) requirements, it is crucial to get for each potential candidate IoT service its accurate QoS values. With the proliferation of IoT devices and their services, this remains a challenging concern. QoS prediction presents an interesting alternative to cope with this issue [21, 14, 9, 18]. Mainly, the existing approaches use Collaborative Filtering (CF) method. CF-based approaches enable to predict QoS parameters of a given service for a given user based on previous historical QoS values of other services. Although CF-based approaches have shown interesting results to QoS prediction, they suffer from scalability issues [2], which limits their application in large scale IoT systems.

In recent years, Deep Neural Network (DNN) based models have been explored [5, 17, 18]. Mainly, they combine DNN with CF-based methods. Despite such hybrid methods enhance the efficiency of the classical CF-based approaches, they remain limited to making real-time QoS prediction for large-scale systems. The fully DNN-based models address these drawbacks [18]. Despite their advancements and performance, they require to inventory beforehand all the services for which it could be necessary to predict their QoS values. This assumption is very restrictive in IoT environments known for their inherently dynamic nature.

To tackle these limitations, in this paper, we propose a DNN-based approach that combines and leverages the power of Long Short-Term Memory (LSTM) [12] and Residual Network (ResNet) models [3]. LSTM network is a special kind of Recurrent Neural Network (RNN), able to learn long-term dependencies, which makes them well-suited to handle sequential data-based problems. In our work, LSTM is used to efficiently extract the underlying latent vector representation of services based on the history of their QoS values for a set of tasks they have performed. This provides valuable insights into the historical behavior of services, which overcomes the limits of existing DNN-based approaches which assume that the set of services must be fixed. Then, a ResNet model is used to predict QoS parameters according to the built latent vector. The main contributions of the paper can be summarized as follows:

1. We propose a new and efficient DNN-based approach that combines LSTM and ResNet models that outperforms the existing CF and DNN-based approaches.
2. Our approach enables to capture the services' representation automatically, which avoids the assumption of DNN-based existing approaches that rely on a predefined exhaustive list of services and their representation
3. The proposed approach has been implemented and evaluated against existing CF and DNN approaches. The results show significant improvements and demonstrate the effectiveness of our approach in terms of performance and accuracy.

The rest of the paper is organized as follows. Section 2 discusses the related works. The proposed approach is presented in Section 3. Experimental results are discussed in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

The problem of QoS prediction has been widely studied in the literature [11, 21, 14, 9, 18]. We distinguish particularly two categories: 1- *CF-based* and 2- *DNN-based* approaches. CF methods are the most commonly used for QoS prediction [10]. A QoS value of a service for a given user is predicted based on the QoS values of similar users and services. These approaches can be classified into *neighborhood-based* and *model-based* methods. Neighborhood-based methods, also known as memory-based methods, assume a stable similarity relationship between users or services. Model-based approaches aim to build a predefined model to predict the required QoS values. Matrix factorization (MF) is the most popular model-based CF method that decomposes the user-item scoring matrix into a combination of several parts. The authors in [21] used MF to find the latent vector of every user and every service. Then Pearson Correlation Coefficient (PCC) has been applied to find the Top-K similar users and services. The desired QoS value is just the weighted average of the QoS values of similar users and services. The authors in [11] proved that adding a non-negative constraint may be suitable for the QoS prediction because almost all the QoS parameters are positive values. The authors in [14] used the non-parametric correlation coefficient Kendall's Tau to compute the similarity between users and services. It is worth noting that Kendall's Tau is less sensitive to outliers and it produces better accuracy. To improve the accuracy of QoS prediction, other works have integrated contextual information such as location and reputation into the proposed MF-based model [9, 15]. In [6], the authors used a sequence of matrices for QoS experiences representation. Then, the prediction of the QoS values for each user on each service for a current time slice is computed based on the average of the precedent values. Although model-based approaches enhance QoS prediction compared to neighborhood-based approaches, they also suffer from scalability issues. This is due to the fact that the whole process operates at runtime, and the substantial volume of data that the model must handle directly impacts the performance of the approach.

More recently, new DNN-based models have been investigated. In [1], a time-aware QoS prediction based on a DNN with gated recurrent units (GRU) model has been proposed. This approach considers temporal slices of service invocations. In this context, QoS experiences are captured as a three-dimensional matrix. The binarization and neighborhood features are integrated to respectively represent the user's and service's features. GRU is used to mine temporal features among users and services so that QoS parameters can be predicted. In [13], the authors proposed a Recurrent Neural Network based collaborative filtering approach for QoS-prediction for the Internet of Vehicles (IoV). A matrix factorization model with Convolutional Neural Network has been proposed in [17].

In [16], the authors provided an improved ARIMA model for QoS prediction in mobile edge computing. The QoS values are captured as a temporal sequence of QoS matrices, which are compressed using Singular Value Decomposition (SVD). Then, the values of the next temporal QoS matrice are predicted. In [18], the authors proposed a model based on ResNet. A user and a service are represented by a multidimensional vector through an embedding layer. Probabilistic distributions of user and service QoS values are added to form the input to the ResNet model.

This work operates on a fixed directory of services and tasks, which presents a serious limitation in open systems such as IoT-based systems.

Despite the advancements of DNN-based approaches in terms of accuracy and performance according to CF-based works, their effectiveness remains limited, especially in the context of open large-scale IoT systems. Indeed, on one side, the works that combine CF with DNN inherit partially the limits of CF-based models, and on the other side, fully DNN-based approaches require fixing the set of services for which QoS prediction could be necessary. To address these limitations, we propose an efficient novel LSTM and ResNet-based approach, in which the set of services is not fixed.

3 Neural networks based QoS prediction approach

As depicted in Fig. 1, the approach we propose involves three parts:

- *Input description*: corresponds to the representation of the required task to fulfill and the history of QoS values of a targeted service (i.e., service experience).
- *Latent Block*: generates the service latent vector representation from the given input data.
- *ResNet Block*: uses the generated latent vector to predict QoS value of the given service for the given task. Hereafter, we detail each step.

3.1 Input Description

The model's inputs are a task description and a service experience (i.e., QoS history). Tasks are identified through their unique identifier, which is a positive integer between 0 and the maximum number of given tasks. The representation of a service history s_j is in the form of a dictionary where each task performed by s_j is associated with the QoS value that has been recorded. It is in the form : $s_j = \{task\_id_1 : qos\_value_1, task\_id_2 : qos\_value_2, ..., task\_id_n : qos\_value_n\}$

Example 1. We consider for instance response time as a QoS parameter. If we suppose that, in past execution, a service s has taken 3.5 unit of time to execute a task t_1 and 1.2 unit of time to execute a task t_4 of a given abstract process, the experienced representation is $s = \{t_1 : 3.5, t_4 : 1.2\}$

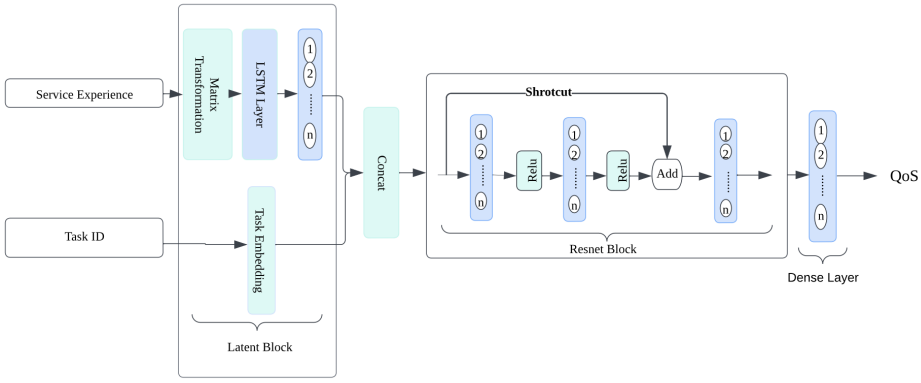

Fig. 1. DNN-based architecture of the proposed approach

3.2 Latent Block

As illustrated in Fig. 1, the latent block relies on task embedding, matrix transformation, and LSTM presented below.

Task Embedding A task is represented by its *One-Hot-Encoding vector*. One-hot-encoding is used in machine learning as a method to quantify categorical data. In short, it produces a vector with a length equal to the number of abstract tasks to satisfy. All the components of the vector are assigned the value 0 except for the one which makes it possible to distinguish each task. The one-hot-encoding is then propagated through a dense layer to yield a more enriched and diversified representation of each task.

Example 2. If the number of task listed in our system is 5, the dimensionality of one hot encoding vector is set to five. For instance, the tasks t_1, and t_4 can be represented respectively as follows:

$$v_1 = \left[1, 0, 0, 0, 0\right], v_4 = \left[0, 0, 0, 1, 0\right]$$

Matrix Transformation The QoS record transformation aims to transform each service QoS history into a matrix. To do so, each pair of the QoS service history $(task\_id\textbf{:}\ qos\_value)$ is replaced by the vector: $qos\_value \times OneHotEncoding(task\_id)$.

Example 3. According to the history representation of the service $s = \{t_1 : 3.5, t_4 : 1.2\}$ given in Example 1, the vectors $3.5\times$ OneHotEncoding(t_1) and $1.2\times$ OneHotEncoding(t_4) are generated respectively as follows:

$$v_{t_1} = \left[3.5, 0, 0, 0, 0\right], v_{t_4} = \left[0, 0, 0, 1.2, 0\right]$$

The final service representation is the matrix formed by the vectors of all the historical experiences of each service for each task. The generation of the representation of the services is depicted in Algorithm 1. Its input is a QoS history record Q. The algorithm starts with the transformation of each service experience for each task into a vector. The resulting vector is then added to the final matrix. This has a linear complexity calculus.

Algorithm 1 Services experience transformation

1: **Input:** QoS history Q
2: **Output:** Service representation matrix
3: $service\_matrix \leftarrow []$
4: **for** each $id\_task : qos\_value \in Q$ **do**
5: $\quad vect \leftarrow qos\_value \times OneHotEncode(id\_task)$
6: $\quad service\_matrix.add(vect)$
7: **end for**

Example 4. If we consider the generated vectors given in Example 3, the service representation of s_1 is as follows:

$$\begin{bmatrix} 3.5 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1.2 \\ 0 & 0 \end{bmatrix}$$

LSTM Layer In our model, we handle the service matrix as a sequential data structure, where each column represents a time step. This sequential representation is then fed into an LSTM layer to generate a condensed and informative representation of the service QoS experience. The LSTM model will enable to capture long-term dependencies to extract and gather relevant information from the service's experiences into a vector. This vector is then passed to a Dense Layer to produce the final representation of the service as a vector based on its QoS history (i.e., service latent vector).

3.3 ResNet Block

Based on a given service and a given task representations, the aim is to identify their relationships with the expected QoS value to predict. The generated service latent and task representation vectors are concatenated and inputted into the Resnet block. This is performed through a Residual Network [3] which is a continuation of dense layers with a skip connection. Finally, the output of the Resnet block is then fed into one last Dense Layer, whose aim is to generate the final output which corresponds to the predicted QoS value of the execution of the given task by the given service.

Given that we are dealing with a regression problem, we have therefore opted for the use of Mean Absolute Error (MAE) as a loss function, given in Equation 1. MAE is more stable for managing outliers which suits the prediction of QoS values we tackle.

$$MAE = \frac{\sum_{s,t}(|q_{s,t} - \hat{q}_{s,t}|)}{N} \tag{1}$$

where $\hat{q}_{s,t}$ and $q_{s,t}$ are the predicted and ground truth value of the target service when invoked for a specific task respectively, N is the number of the predicted QoS values.

It is worth noting that during the training phase, the *Adam (Adaptive Moment Estimation)* [8] optimization algorithm is used to update the parameters of the Dense Layers. This algorithm offers several advantages, such as the ability to achieve faster convergence with minimal hyperparameter tuning, which leads to enhancing the efficiency and effectiveness of the training.

4 Experimental evaluation

The proposed approach has been implemented using Python 3.11. The experiments have been conducted on Windows 10 64-bit with 2.4 Ghz Intel(R) I7 processor and 16GB RAM.

4.1 Used dataset

We have conducted experiments using the WSDream dataset [19]. This dataset provides a matrix of the response time and throughput values collected from 5 825 services and 339 users. It gathers 1974675 historical records of service invocations. The response time values are concentrated between 0 and 2 ms and the throughput is between 0 and 200 kbps.

Data transformation The original dataset is in the form of a matrix $n \times m$ where missing values are represented by -1. We recorded each task by its identifier. For the services, the histories that correspond to the rows in the matrix are transformed into a dictionary $service\_id : qos\_value$. Thus we obtained an adapted version of the dataset.

4.2 Comparative study

In order to evaluate and quantify the prediction quality of our approach, we use the two metrics: *Mean Absolute Error (MAE)* given above in Equation 1 and *Root Mean Squared Error (RMSE)*. MAE is a linear score which means that all the individual differences are weighted equally in the average. MAE reflects the overall accuracy of QoS prediction, which averages absolute deviations to the ground truth QoS values.

RMSE, given in Equation 2, measures the deviations between the predicted QoS and their corresponding observed QoS, which is then squared and averaged for calculating the square root. RMSE gives a relatively high weight to large errors due to the fact that errors are squared before they are averaged.

$$RMSE = \sqrt{\frac{\sum_{s,t}((q_{s,t} - \hat{q}_{s,t})^2)}{N}} \tag{2}$$

A comparative study has been conducted to compare our approach with the following main existing approaches: *U-I-PCC (User-Item Pearson Correlation Coefficient)* [20], *NMF (Non-Negative Matrix Factorisation approach)* [11], *NeuMF (Neural Matrix Factorization)* [4], *PLRes (A Probability Distribution and Location-aware ResNet Approach for QoS Prediction)* [18].

Fig. 2 and Fig. 3 show a comparison of our approach with the different approaches listed above based on MAE and RMSE respectively on four different matrix densities (5%, 10%, 15%, and 20%).

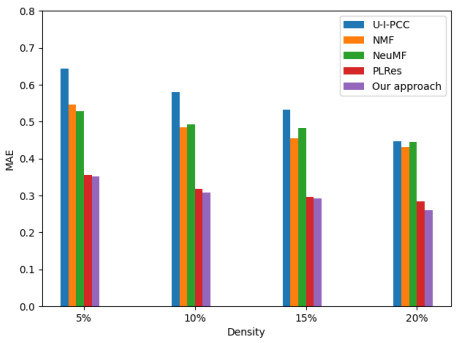

Fig. 2. MAE loss comparaison

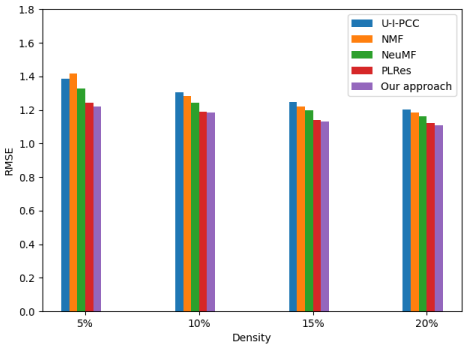

Fig. 3. RMSE loss comparaison

The accuracy of all approaches increases as density increases. This is due to the fact that a higher density of data provides considerable information for models to learn, which leads to more accurate predictions. We note that our approach demonstrates enhanced results.

Fig 4 illustrates a comparison of the prediction time of our approach with the four approaches. As can be seen, our approach outperforms significantly CF-based approaches. However, PLRes shows a slight gain in terms of prediction time compared to our approach. This is directly related to the adopted service representation. Specifically, the PLRes approach relies on a fixed static set of services. As mentioned earlier, this limitation restricts the applicability of the proposed approach in open IoT systems, as it cannot predict QoS values for services outside the fixed set. In contrast to this method, our approach auto-

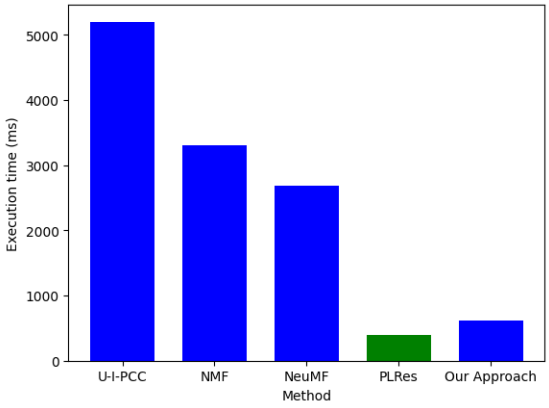

Fig. 4. Prediction time evaluation

matically generates service representations from their histories using an LSTM model. This flexibility renders our approach highly suitable for open dynamic systems.

5 Conclusion

In this paper, we have proposed a QoS prediction approach that combines the strengths of Long Short-Term Memory (LSTM) and Residual Network (ResNet) models. The LSTM model enables the generation of services latent vector by capturing relevant information from the service history. This latent vector is then fed into a ResNet which handles the underlying patterns and relationships to enable efficient QoS prediction. To evaluate and validate the effectiveness of the proposed approach, we conducted a series of comparative experiments with existing approaches. The results clearly demonstrate that our approach outperforms the existing works and offers better overall performance. In particular, it provides significant improvements in prediction time while guaranteeing a high level of accuracy. In addition, this approach avoids the limitations of DNN-based approaches that rely on a static set of services.

We plan to extend the proposed approach to tackle the challenging problem of predictive QoS-aware dynamic IoT service composition while considering other properties such as the mobility dimension and the spatio-temporal properties.

Acknowledgements This work was supported by the ANR LabEx CIMI (grant ANR-11-LABX-0040) within the French State Programme "Investissements d'Avenir".

References

1. Deeptsqp: Temporal-aware service qos prediction via deep neural network and feature integration. Knowledge-Based Systems (2022)

2. Al-Ghuribi, S., Noah, S.A.M.: Multi-criteria review-based recommender system – the state of the art. IEEE Access (2019)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
4. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web. p. 173–182 (2017)
5. Huang, W., Zhang, P., Chen, Y., Zhou, M., Al-Turki, Y., Abusorrah, A.: Qos prediction model of cloud services based on deep learning. IEEE/CAA Journal of Automatica Sinica (2022)
6. Jin, Y., Guo, W., Zhang, Y.: A time-aware dynamic service quality prediction approach for services. Tsinghua Science and Technology (2020)
7. Khadir, K., Guermouche, N., Guittoum, A., Monteil, T.: A genetic algorithm-based approach for fluctuating qos aware selection of iot services. IEEE Access (2022)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Li, S., Wen, J., Luo, F., Cheng, T., Xiong, Q.: A location and reputation aware matrix factorization approach for personalized quality of service prediction. In: IEEE International Conference on Web Services (ICWS) (2017)
10. Lo, W., Yin, J., Deng, S., Li, Y., Wu, Z.: Collaborative web service qos prediction with location-based regularization (2012)
11. Luo, X., Zhou, M., Xia, Y., Zhu, Q.: Predicting web service qos via matrix-factorization-based collaborative filtering under non-negativity constraint. In: 2014 23rd Wireless and Optical Communication Conference (WOCC) (2014)
12. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on pp. 2673 – 2681 (1997)
13. Tingting Liang, Manman Chen, Y.Y.L.Z., Ying, H.: Recurrent neural network based collaborative filtering for qos prediction in iov. IEEE Transactions on Intelligent Transportation Systems pp. 2400–2410 (2022)
14. White, G., Palade, A., Cabrera, C., Clarke, S.: Iotpredict: Collaborative qos prediction in iot. In: 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom) (2018)
15. Xu, J., Zheng, Z., Lyu, M.R.: Web service personalized quality of service prediction via reputation-based matrix factorization. IEEE Transactions on Reliability (2016)
16. Yan, C., Zhang, Y., Zhong, W., Zhang, C., Xin, B.: A truncated svd-based arima model for multiple qos prediction in mobile edge computing. Tsinghua Science and Technology (2022)
17. Yin, Y., Chen, L., Xu, Y., Wan, J., Zhang, H., Mai, Z.: Qos prediction for service recommendation with deep feature learning in edge computing environment. Mob. Networks Appl. (2020)
18. Zhang, W., Xu, L., Yan, M., Wang, Z., Fu, C.: A probability distribution and location-aware resnet approach for qos prediction. CoRR (2020)
19. Zheng, Z., Zhang, Y., Lyu, M.R.: Investigating qos of real-world web services. IEEE Transactions on Services Computing pp. 32–39 (2014)
20. Zheng, Z., Ma, H., Lyu, M.R., King, I.: Wsrec: A collaborative filtering based web service recommender system. In: IEEE International Conference on Web Services (2009)
21. Zheng, Z., Ma, H., Lyu, M.R., King, I.: Collaborative web service qos prediction via neighborhood integrated matrix factorization. IEEE Transactions on Services Computing (2013)