A Study on Latency Prediction in 5G network

Seunghan Choi
Packet Network Research Section
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea
shchoi@etri.re.kr

Changki Kim
Packet Network Research Section
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea
ckkim1@etri.re.kr

***Abstract*—** These days, due to the increase in the use of mobile terminals such as smartphones, tablets, and XRM(Extended Reality and Media) service terminals, heterogeneous networks for various services are often connected to the 5G network. Low latency should be supported on the network for these services. At the time of measuring the latency at the current time point, recalculating the end-to-end QoS path, or informing the XRM service application, it can be a past value, which can lead to an inaccurate situation. To overcome this situation, 5G network needs to predict latency in advance, recalculate end-to-end QoS paths based on this information, or informs XRM applications to meet more effective QoS requirements. In this paper, we have evaluated the performance of several machine learning models for predicting latency, and introduce the results of experimenting with performance.

Keywords—5G; latency; prediction; machine learning

I. INTRODUCTION

These days, due to the increase in the use of mobile terminals such as smartphones, tablets, and XRM(Extended Reality and Media) service terminals, heterogeneous networks for various services are often connected to the 5G network.

Interactive high-latency applications such as virtual reality(VR), augmented reality(AR), extended reality(XR), and online games are becoming increasingly popular. To support these XRM services on the Internet, L4S technology is being standardized in the IETF[1]. In addition, standardization is underway to provide XRM services in 3GPP[2].

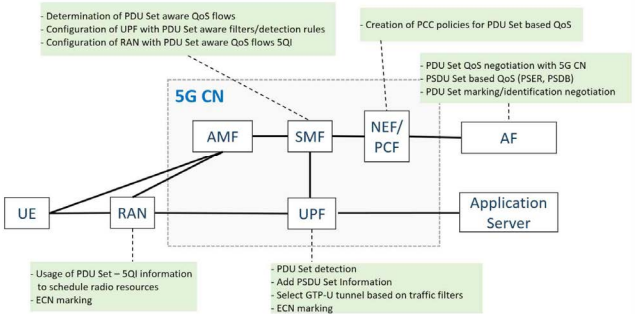

Fig. 1. 5GS QoS framework for XRM service

High reliability and ultra-low latency communication are required to meet various vertical industry domain requirements. Among the communication technologies that meet these requirements is IEEE 802.1 Time-Sensitive Networking (TSN) [3] technology, and 3GPP is standardizing 5G mobile communication to integrate 5G systems with IEEE TSN after 2018 Rel-16 [4].

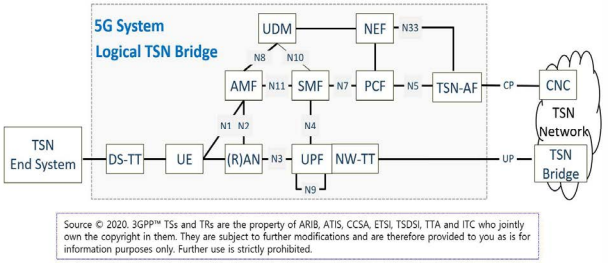

Fig. 2. System architecture view with 5GS appearing as TSN bridge[4]

The above two services require end-to-end qos that satisfies low latency and low loss. In the face of increasing latency and loss, it can be effective to replace existing NG-RAN(Next-Generation Radio Access Network) path with other NG-RAN path with low loss and latency, or to replace other UPFs (User Plane Functions) with low loss and latency in 5G network. Also, In the event of congestion or loss in the 5G network, the intermediate node of the 5G network must quickly inform the terminal of network status information so that congestion or loss can be avoided. At the time of measuring the latency at the current time point, recalculating the end-to-end QoS path, or informing the XRM service application, it can be a past value, which can lead to an inaccurate situation. To overcome this situation, 5G network needs to predict latency in advance, recalculate end-to-end QoS paths based on this information, or informs XRM applications to meet more effective QoS requirements.

In this paper, we have evaluated the performance of several machine learning models for predicting latency, and introduce the results of experimenting with performance.

II. RELATED WORKS

Latency is a kind of time series data, and time series data is usually predicted by a regression model. regression is a technique that collectively refers to a model for the correlation between several independent variables and one dependent variable. From a machine learning point of view, the independent variable corresponds to features, and the dependent variable corresponds to the result value. The key to machine learning regression prediction is to find the optimal regression

coefficient through learning from the given features and result value data.

A. Linear regression

Simple linear regression is a regression with one independent number of sessions and one dependent variable. It is a model that optimizes regression coefficient so that RSS(Residual Sum of Squares) of predicted and actual values can be minimized, and does not apply regularization.

B. Ridge regression

The cost function of the linear model only considered minimizing the difference between the actual value and the predicted value. In this case, the volatility becomes rather severe, and the prediction performance is likely to deteriorate. Reflecting this, the cost function should balance the RSS minimization method of minimizing the residual error value of the learning data and the method of preventing the regression coefficient value from increasing to prevent overfitting. The method of improving overfitting by reducing the size of the regression coefficient value by penalizing the cost function is called regularization. Regularization is divided into L1 and L2 methods. Ridge regression is a model applied with L2 regularization.

C. Polynomial features regression

It is called polynomial regression that regression is expressed in polynomials such as quadratic and tertiary equations, not monomial equations of independent variables. Depending on the dataset, the prediction performance may be better to analyze with polynomial regression rather than monomial regression in the form of a linear equation. That is, polynomial regression can model complex polynomial relationships rather than linear relationships of features. As the order of the polynomial increases, it is possible to model the relationship between very complex features.

D. Random forest regression

Random Forest is supervised learning algorithm having classification and regression capabilities. It is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest[5]. It has relatively fast performance speed among ensemble algorithms and high predictive performance in various areas. However, it also has the disadvantage of having too many hyperparameters, which consumes a lot of time for tuning.

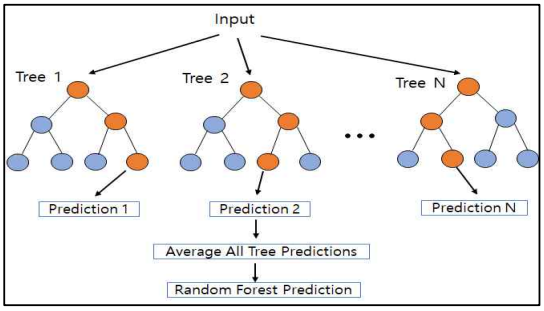

Fig. 3. Random Forest Tree

E. GBM(Gradient Boost Machine) regression

Boosting algorithm is a method that creates a strong learner by weighting the learner that predicts the error-prone data while sequentially learning and predicting multiple weak learners. GBM[6] uses gradient descent to update weights. Gradient descent is a method of differentiating the loss function into a parameter to obtain the slope, and to move the parameter in the direction of decreasing this loss value. GBMs often have slightly better predictive performance than Random Forest. However, it takes a long time to perform and requires parameter tuning to improve prediction performance.

F. XGBoost(eXtra Gradient Boost) regression

Tree boosting is a highly effective and widely used machine learning method. In particular, Gradient Boost Machine (GBM) is often used as a technique to solve the update values of weights so that errors can be minimized through repetitive operations. A XGBoost[7] is a scalable tree boosting system based on GBM. GBM has a problem of overfitting and slow performance time. But, it guarantees faster performance than GBM by performing in parallel. It has its own internal overfitting regularization function to prevent overfitting. And, Its core library is written in C/C++. XGBoost uses level-wise tree growth. Level-wise tree growth is a method of horizontal growth by first traversing nodes close to root nodes to reduce and balance the tree depth. For level-wise tree growth, additional operations are needed to balance, but the advantage is that regularized training is possible.

G. LGBM(Light GBM) regression

The biggest advantage of LightGBM[8] is that it has less learning time than XGBoost. It uses the leafwise tree segmentation scheme, unlike the tree segmentation scheme in the normal GBM family. LGBM uses leaf-wise tree growth. It continuously splits leaf nodes with max delta values without balancing the tree. Split-generated rule trees have the advantage of minimizing prediction error loss in some cases over balanced tree methods as learning repeats. If you have a lot of data, you can guarantee robustness and efficiency, but if you don't have a lot of datasets, LGBM is relatively easy to overfit.

H. Stacking regression

Stacking has something in common with bagging and boosting in that it combines multiple individual algorithms to derive prediction results. However, the biggest difference is that we perform predictions again based on the data predicted by individual algorithms. In other words, the prediction result data set of individual algorithms is made into a final metadata set, final learning is performed with a separate machine learning algorithm, and final prediction is performed again based on test data.

I. Weighted blending

A weighted blending is an extension to the voting ensemble that assumes that all models are equally skilled and make equal proportional contributions to the predictions made by the ensemble. Each model is assigned a fixed weight that is used to

multiply the predictions made by the model and calculate the total or average predictions. The challenge for this type of ensemble is how to compute, assign, or retrieve model weights that provide better performance than an ensemble that uses the same model weights as the contributing models.

III. Experiment and Result

A. Dataset

We used the 5G datatset released in other study[9] for latency prediction. This dataset contains packet captures (PCAPs) of a 5G campus network. A 5G campus network is a 5G network for the users affiliated with the campus organization, e.g., an industrial campus, covering a prescribed geographical area. A 5G campus network can operate as a so-called 5G non-standalone (NSA) network access or as a 5G standalone (SA) network.We used the dataset of 5G SA one way delay for data learning. We divided the dataset into learning data and test data at a ratio of 7:3. Also, standard scaling was performed to change the dataset to a standard normal distribution form. The 24 latency values are stored as x values and the 25th value as y value.

B. Prediction Models

Linear, Ridge, Polynomial, Voting, Random forest, GBM, XGBoost, LGBM, Stacking, and Weighted blending regression model were used to compare the performance of the prediction models. The ridge model was used with the polynomial feature processing. Linear, ridge, lasso, and poly ridge models were used for the voting model. For the primary model of the stacking model, the random forest and polynomial model were used, and the XGBoost model was used as the final model. The weighted blending models used random forest, XGBoost, and LGBM models. The prediction values were multiplied by the weights of 0.5, 0.3, and 02 respectively, and the final prediction values were derived by adding them all.

C. Performance

RMSE(Root Mean Square Error) and R2 were selected as error metrics for performance comparison. The higher the RMSE value, the better a model fits a dataset. The R2 is a statistical measure of the proportion of variance of the dependent variable described as an independent variable or variable in the regression model.. The higher the R2 value, the better a model fits a dataset.

TABLE I. PEFORMACE EVALUATION

Model	RMSE	R2
Linear	0.08835	0.33745
Ridge	0.08835	0.33740
Polynomial	0.08602	0.37196
Voting	0.09058	0.30368
Random forest	0.08313	0.41346
GBM	0.08369	0.40549
XGB	0.08609	0.37087
LGBM	0.08339	0.40977
Stacking	0.08929	0.32334
Weighted blending	0.08219	0.42668

Table 1 shows the performance evaluatiom about ten models. According to the performance evaluation results, The weighted blending model shows the best performance of other models on error metric RMSE and R2.

IV. Conclusion

In this paper, Linear, Ridge, Polynomial, Voting, Random forest, GBM, XGBoost, LGBM, Stacking, and Weighted blending regression model were used to compare the performance of the prediction models. RMSE(Root Mean Square Error) and R2 were selected as error metrics for performance comparison.

The weighted blending model shows the best performance of other models on error metric RMSE and R2. We plan to apply this research in the end-to-end QoS path calculation between 5G network and XRM service to satisfy more effective QoS needs.

Acknowledgment

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00974, Development of ultra-reliable and low-latency 5G+ core network and TSN switch technologies).

References

[1] Bob Briscoe, Koen De Schepper, Marcelo Bagnulo, Greg White, “Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture”, IETF, Tech. Rep. RFC 9330. [Online]. Available: https://tools.ietf.org/html/rfc9330

[2] 3GPP, TR 23.700-60, ” Study on XR (Extended Reality) and media services (Release 18) “, 2022. 12.

[3] Janos Farkas et al., “Time-Sensitive Networks Standards,” IEEE Comm. Magazine, June 2018, pp. 20-68.

[4] 3GPP TS 23.501, “System Architecture for the 5G System; Stage 2,” 2020.

[5] L. Breiman, "Random forests," Machine learning, vol. 45, pp. 5-32, 2001.

[6] J. Friedman. Greedy function approximation: a gradient boosting machine. Annals of Statistics, 29(5):1189–1232, 2001.

[7] Chen, Tianqi; Guestrin, Carlos. "XGBoost: A Scalable Tree Boosting System". Proceedings of the 22nd ACM SIGKDD InternationalConference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. ACM. pp. 785–794

[8] Ke, Guolin; Meng, Qi; Finley, Thomas; Wang, Taifeng; Chen, Wei; Ma, Weidong; Ye, Qiwei; Liu, Tie-Yan. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". 31st Conference on Neural Information Processing Systems (NIPS 2017)

[9] Justus Rischke, Peter Sossalla, Sebastian Itting, Frank H. P. Fitzek, Martin Reisslein, “5G Campus Networks: A First Measurement Study” IEEE Access, Vol.9 September 2021, pp. 121788-121803.

[10] “scikit-learn” Machine Learning in Python https://scikit-learn.org/stable/