IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 15, 2021, accepted December 23, 2021, date of publication December 24, 2021,
date of current version December 31, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3138695

Predicting High Delays in Mobile Broadband Networks

AZZA H. AHMED[1,2], (Member, IEEE), STEVEN HICKS[1,2], (Member, IEEE), MICHAEL ALEXANDER RIEGLER[1], AND AHMED ELMOKASHFI[1], (Member, IEEE)

[1]SimulaMet—Simula Metropolitan Center for Digital Engineering, 1325 Oslo, Norway
[2]Department of Computer Science, OsloMet—Oslo Metropolitan University, 0167 Oslo, Norway

Corresponding author: Azza H. Ahmed (azza@simula.no)

ABSTRACT The number of applications that run over mobile networks, expecting bounded end-to-end delay, is increasing steadily. However, the stochastic and shared nature of the wireless medium makes providing such guarantees challenging. Using several network interfaces simultaneously can help address fluctuating delays, provided that transport protocols can switch between them in a timely manner. Today's protocols are mostly closed-loop and thus require at least one round trip before reacting to increased delay. This paper examines whether jumps in round trip times (RTTs) have a pattern that can be predicted beforehand. Using per second RTT measurements from hundreds of probes in two Long Term Evolution (LTE) cellular networks, we train an ensemble of classifiers to detect increases in delay. We construct a parsimonious explainable model that provides an accuracy of 80% and does not appear to be specific to a particular mobile operator. Further, we examine whether our model can be extended to 5G using a small dataset with extra 5G metadata, resulting in an accuracy of 88%. Our model indicates that RTTs are long-range correlated and shows that radio measurements of channel occupancy are accurate predictors of the onset of high delays. These results suggest that it is feasible to build an open-loop control system for multiplexing among several interfaces to proactively bound delays.

INDEX TERMS Delay, prediction, machine learning, LTE, 5G.

I. INTRODUCTION

Guaranteeing low and stable end-to-end delay over mobile networks is one of the key motivations for 5G. Ultra-reliable low-latency communication is one of three use cases 5G is envisioned to cater for [1]. Reliable latency is important for supporting interactive applications such as hepatic control, virtual and augmented reality, and critical applications such as smart grid metering and public safety communication.

Delays can increase for a number of reasons, including interference, handover, and congestion both in the radio and beyond [2], [3]. New error correction mechanisms and novel radio access strategies, such as the flexible numerology introduced by 5G new radio [4], may help drive delay down [5]. However, addressing congestion and handover remains challenging because of the stochastic, shared and time-slotted nature of the wireless medium.

Leveraging the availability of several radios per end device has also emerged as a potential approach to bound performance unpredictability. Previous studies have shown that network availability can be boosted to five nines by connecting to two mobile operators simultaneously and the throughput can be enhanced markedly [6], [7]. Several multipath transport protocols, such as Multipath Transmission Control Protocol (MP-TCP) and QUIC multipath, are standardized to support the simultaneous use of multiple links [8], [9]. These protocols use a scheduler that monitors the state of each link in use before deciding which link to use next. Similar to TCP, performance monitoring is essentially a closed-loop that requires at least a single round trip, but often several, before taking a qualified decision. Unfortunately, this waiting time can be too long to meet the expectations of delay-sensitive applications.

To address these limitations, we ask the simple question of RTTs over mobile networks can be predicted by end devices. We are not interested in the exact value of the RTT, but rather whether it falls below or above a certain threshold. Furthermore, the prediction should be based only on measurements and metadata that are available to end devices, like for example, signal strength. Accurate predictions can help

The associate editor coordinating the review of this manuscript and approving it for publication was Ehab Elsayed Elattar.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

TABLE 1. List of abbreviations.

Abbreviation	Description
RTT	Round Trip Time
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indicator
RAT	Radio Access Technology
MCS	Modulation Coding Scheme
NSA	Non-Stand Alone
UE	User Equipment
NNE	NorNet Edge
ARIMA	AutoRegressive Integrated Moving Average
LR	Logistic Regression
RF	Random Forest
SVM	Support Vector Machine
DT	Decision Tree
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
MCC	Matthews Correlation Coefficient
SMOTE	Synthetic Minority Oversampling Technique

transport protocols to short-circuit the closed control loop by making local decisions instead of waiting for at least one RTT.

We leveraged end-to-end measurements and metadata from a large number of stationary probes connected to two mobile operators over LTE. Then, we trained a number of machine learning classifiers to verify whether delays could be reliably predicted. We focused on stationary measurements because it was the simplest scenario and thus succeeding in predicting stationary delays is the first step towards scenarios with complex mobility. Furthermore, many use cases with stringent delay requirements are associated with low to no mobility (e.g., smart meters). Interestingly, we found that a binary ensemble classifier could accurately predict low and high delay in 80% of the cases. In fact, the classifier also predicted correctly 75% of the worst 10% of the RTTs. More importantly, the model is interpretable and transferable to other network operators and requires minimal retraining to remain effective over an extended period. Moreover, we tested our classification model on a small 5G dataset of RTT measurements and extra metadata. The model achieved an accuracy of 88% for classifying the delays. Our findings can be readily used to improve the performance of multipath protocols when using several wireless links for bounding delays.

The rest of the paper is organized as follows: We present our measurement data in Sec. II. We then discuss our approach for predicting delays in Sec. III and present the prediction results in Sec. IV. Sections V and VI dig deeper into failed predictions and examine the prediction accuracy over time. In Sec. VII, we investigate the performance of our model on 5G data. We review related work in Sec. VIII. The main findings are discussed in Sec. IX before concluding in Sec. X

II. MEASUREMENT DATA

In this section, we describe our measurement setup, dataset, and pre-processing steps.

We studied RTT measurements from a set of geographically spread stationary probes. These probes are part of

FIGURE 1. Measurement node. The red box encloses the single board computer. The box also includes a smart power socket that can be rebooted via SMS.

the NorNet Edge (NNE) platform, which is a country-wide setup for measuring commercial mobile broadband networks in Norway. The probe is a single-board computer that runs Linux and connects to at least two mobile operators using commercial off-the-shelf user equipment (UE) and subscriptions. More specifically, we use the APU2 platform from PC Engines (see Figure 1).[1] Our board is equipped with a quad core CPU, 4GB RAM and two miniPCI slots. To connect to commercial mobile networks, we use the Sierra Wireless AirPrime MC7455 miniPCI modem, which supports LTE CAT 6 (LTE-advanced).[2] The modem uses external antenna, which are visible in Figure 1. To enhance the availability of the nodes, we attach them to a smart power socket that can be power-cycled remotely via SMS. Our probes conduct end-to-end measurements to a set of well provisioned servers that we control, these include delay, packet loss, and speed. Figure 2 illustrates the measurement setup. An NNE node connects to the Internet via commercial mobile subscriptions and performs end-to-end measurements to the NNE backend.

In this study, we consider measurements from the two largest mobile operators in Norway, which we refer to as Op_1 and Op_2 in the sequel. The probes measure RTTs by sending a 20-bytes User Datagram protocol (UDP) packet every second, over all available connections, to a well-provisioned server that echoes it back. We focused on the RTT measurements collected over LTE during September and October 2018. The Op_1 dataset includes more than 44.96 million RTT data points from 79 probes, while the Op_2 dataset includes approximately 14.47 million data points from 77 probes. The

[1] https://www.pcengines.ch/apu2.htm
[2] https://www.sierrawireless.com/iot-solutions/products/mc7455/

difference between the two datasets stems from the fact that Op_2 connections were on 3G for a non-trivial duration, and this data had to be filtered out. At the time of the study, Op_2 did not implement handover between radio access technologies while a UE was actively sending data, i.e. data sent by a UE over 3G would not be handed over to 4G. Therefore, many of the connections to Op_2 were on 3G for an extended period of time. Besides filtering out these periods, we removed all instances where a probe underwent maintenance or the NNE backend had issues. The NNE backend is connected to the Internet via a well provisioned link through a research and educational network. However, to avoid including times where the measured RTTs were influenced by congestion in the research and educational network, we filtered all measurements where a large fraction of probes, across operators, registered larger than usual RTTs.

In addition to the active measurements, the probes collect connection metadata. These include radio and connectivity parameters, which are listed below.

- **Received signal strength indicator (RSSI)** is a measure of the power received by the UE, including both the signal and noise.
- **Reference signal received power (RSRP)** is a measure of the power in the LTE reference signal and is averaged over the entire bandwidth. RSRP is a more accurate estimate of the received useful power.
- **Reference signal received quality (RSRQ)** is a measure of the quality of the received signal. A low RSRQ often coincides with a loaded cell.
- **Radio access technology (RAT)** indicates mobile generation in use, that is, 2G, 3G, and 4G.

The metadata is collected every minute, as well as whenever there is a change. We associated every RTT measurement with the closest past metadata value. Next, we removed all RTT measurements without the corresponding metadata. The removed fractions are 0.2% and 0.3% for Op_1 and Op_2, respectively. Finally, we checked the sanity of the metadata values and removed all RTT measurements that were associated with metadata values outside the correct value ranges, that is, RSRP (-44dBm to -140dBm), RSRQ (-3dB to -20dB), and RSSI (-6dBm to -100dBm).

III. PREDICTING ROUND TRIP DELAY

This section takes a closer look at our dataset and approaches to predict delays.

A. RTT MEASUREMENTS

Figure 3 shows the distribution of RTTs for Op_1 and Op_2. There were no clear differences between the two operators for the bulk of the initial part of the distributions. Approximately 60% of the RTTs were within 50ms for both operators. The picture, however, started to change as we look at the worst 20% RTTs with Op_1 performing worse than Op_2.

We are interested in investigating whether high delays, the top 10%, can be predicted based on historical RTT values and available high-level metadata about connection quality. We intentionally avoid using cross-layer information such as MAC layer scheduling decisions and physical layer reports. This is because leveraging these in practice would require complicated APIs that can communicate with the underlying chipset, for example, the approach that tools such as MobileInsight use [11]. Hence, our problem is a classical forecasting problem, which may suggest that available time series analysis techniques such as ARIMA can be a good fit [12]. However, these methods base their prediction chiefly on past values and patterns in the time series and do not lend themselves easily to regularization, that is, adjusting forecasting by incorporating side information about relevant factors such as signal quality. Thus, machine learning (ML) appears to be a viable alternative.

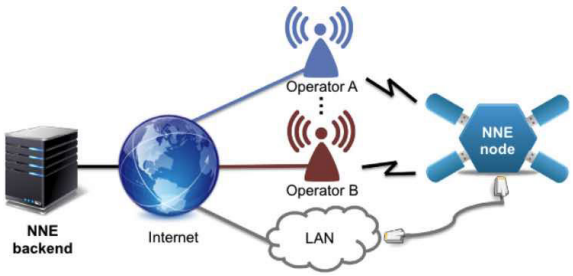


FIGURE 2. Measurement setup [10].

Figure 3 shows that attempting to predict high delays means that we need to handle a heavily imbalanced dataset. Specifically, we categorized the delays into low and high using one threshold per operator, which is 80ms and 60ms for Op_1 and Op_2, respectively. These thresholds are meant to designate the top 10% delays as high, that is, our classes have a relative ratio 9:1 by design. To prepare a balanced dataset, we investigated both oversampling and undersampling. We use the synthetic minority oversampling technique (SMOTE), which applies a nearest neighbor algorithm to generate synthetic data for the minority class [13]. For the undersampling, we used the NearMiss algorithm, which removes samples from the majority class. It removes values that are close to the minority class to increase the spacing between the two classes and avoid information loss [14]. Fitting a random forest classifier, a supervised ensemble learning method [15], and using both SMOTE and NearMiss to balance our data, yields a comparable accuracy of $\approx 79\%$. We decided to proceed with undersampling because it does not require the use of synthetic data.

B. CLASSIFICATION ALGORITHMS

As explained above, our problem is essentially a classification and prediction problem. To this end, we compare the performance of four supervised classification algorithms, which are listed as follows:

1) LOGISTIC REGRESSION (LR)

An interpretable binary classifier that uses a logistic function to model the binary variable. However, it usually does not perform well when the feature space is large [15].

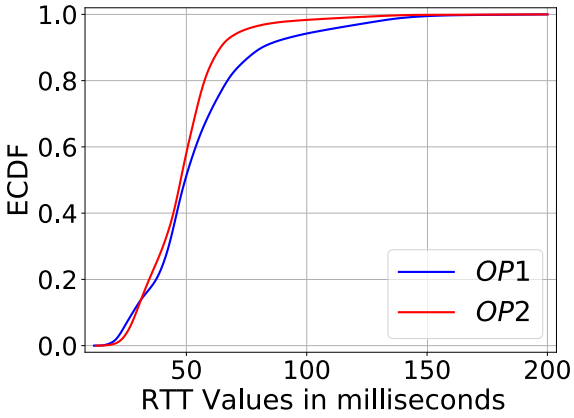


FIGURE 3. Distribution for RTT values for Op_1 and Op_2.

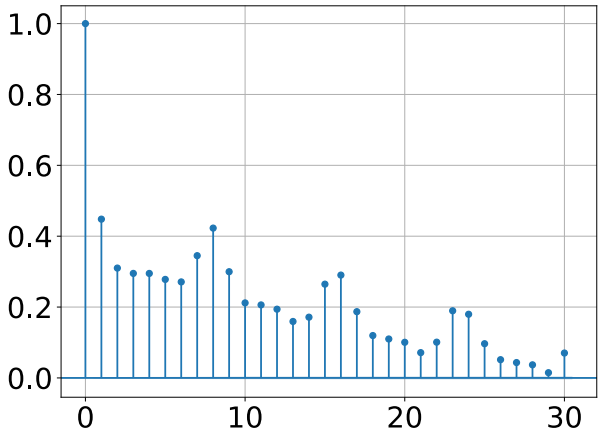


FIGURE 4. Temporal auto-correlation for RTT at time lag = 0,5,10,..,30 seconds.

2) RANDOM FOREST (RF)

An ensemble-based learning algorithm that uses many decision trees to perform either classification or regression [15]. For classification, each decision tree makes an independent prediction, which is then counted to produce the final output. RF is quite robust, but offers less interpretability than LR.

3) LightGBM

A gradient-boosting framework is based on decision tree algorithms [16]. Gradient boosting algorithms combine iteratively a number of weak learners into a single strong learner. Similar to RF, LightGBM is less interpretable than LR.

4) ENSEMBLE

This approach combines the logistic regression, LightGBM, and random forest classifiers into a single model [17]. Each algorithm is trained separately. Then, a gradient-boosted decision tree is trained, based on the predictions from each algorithm along with the input data. This allows for weighting the contribution of each classifier, resulting in a combination that is an improvement over the individual classifiers.

C. RELEVANT FEATURES

We used four groups of features to train the classifiers. The guiding principle in picking these features is to limit ourselves to features that can be readily available to applications and minimize dependencies on cross-layer features. The four groups comprise radio reception quality, diurnal, spatial, and time-series effects.

1) RADIO RECEPTION FEATURES

These involve RSSI, RSRP and RSRQ.

2) DIURNAL EFFECTS FEATURES

The RTT exhibits a certain periodicity in both daily and weekly patterns. To model these effects, we assign each RTT measurement to the respective hour of the day and day of the week.

3) SPATIAL FEATURES

To account for the location of the probe (e.g., urban vs. rural), we identified the coordinates of each probe and mapped it to a (1km×1km) geographical unit that is provided by the state [18]. We then found the population that resides in each identified geographical unit. Based on the distribution of the population per geographic unit, we defined three categories for this feature: (i) low ($<$ 10,000), (ii) medium (10,000-15,000) and (iii) high ($>$ 15,000). These thresholds were determined based on the distribution of the country's population.

4) TIME SERIES FEATURES

We examined whether the RTT time series exhibited autocorrelation and long-range dependence [12]. Figure 4 shows the autocorrelation function (ACF) for the RTT time series from a sample connection at different lags in seconds. We recorded a non-negligible autocorrelation that spreads over several lags. The ACF became weaker for higher lags. While this may be expected because the dataset is dominated by low RTTs, it also indicates that high RTTs may have a serial pattern to them. Therefore, we investigated whether previous RTTs can help predict upcoming delays. To determine how long we need to look back at time, we evaluated the correlation between the current RTT and RTTs from the past 3, 5, 10, 15, and 30s. Limiting ourselves to the past five seconds yielded a reasonable accuracy.

D. APPROACH

We trained four classifiers, one per each of the above algorithms, using one week worth of data from Op_1. We applied these algorithms using the implementations provided by the Python library scikit-learn [19] and LightGBM [16]. We applied RF using a maximum of 600 estimators and adjusted the weights proportionally to the class size. We used K-fold (k = 5) cross-validation for the hyperparameters selection. Further, as for LightGBM, we used the gradient boosted

TABLE 2. Values of accuracy and MCC for different classifiers.

Classifier	Accuracy	MCC
Logistic Regression	64.2%	0.30
Random Forest	78.8%	0.58
LightGBM	76.1%	0.53
Ensemble	80.0%	0.60

decision tree algorithm with a learning rate of 0.01. Owing to the large dataset size, we chose a larger learning rate to reduce the required number of iterations.

We used the first week of September for training and the second week for validation and evaluation. Furthermore, we only focused on Op_1 when fitting the model and used the Op_2 dataset to check whether the model generalizes to other operators.

IV. PREDICTION PERFORMANCE

We now proceed to evaluate the performance of the aforementioned classification algorithms. To this end, we investigate their general accuracy, as well as their efficacy in predicting high delays and model transferability to other network operators.

A. PREDICTION ACCURACY

We applied a number of metrics to compare the four classifiers in use, which we summarize next.

- **Accuracy.** Ratio of correctly classified samples. We also present the accuracy in the form of a confusion matrix.
- **Receiver operating characteristic (ROC) curve.** A graphical measure of the separability of a binary classifier as we vary the discrimination threshold.
- **Precision-Recall curve.** The plot describes the trade-off between precision and recall for different thresholds. A high area under the curve represents both a high recall and high precision. This is more appropriate for imbalanced datasets.
- **Matthews correlation coefficient (MCC).** A measure of the correlation between the actual and predicted samples [20]. Unlike other metrics, MCC is symmetric because it assigns all classes equal importance.

In our evaluation, the true positives (TPs) are the correctly classified high-delay samples. True negatives (TNs) are the correctly classified low-delay samples. The false positives (FPs) are the incorrectly classified low-delay samples. Finally, false negatives (FNs) are the samples incorrectly classified as high-delay.

Table 2 presents the prediction accuracy and MCC of the four classification algorithms. The ensemble, random forest, and lightGBM outperformed logistic regression by a clear margin. The MCC confirms that the results of the ensemble and random forest correlate well with the actual classes across the board. The ensemble model achieved a very good accuracy compared to the closest related work by khatouni et. al [21], which achieved an accuracy of 67% when applying DT using the same features defined by the authors. We believe that the lower accuracy of [21] is due to having neglected the effect of historical data in their model.

The ROC curves and the respective area under the curve (AUC) values in Figure 6 further confirm the above observations for a range of thresholds. The ensemble classifier outperformed all the other three classifiers, with an AUC value of 0.88. The random forest classifier had an AUC of 0.87. The corresponding values for the lightGBM and logistic regression are 0.84 and 0.69, respectively. Moreover, as expected the precision-recall curves in Figure 7 show similar results.

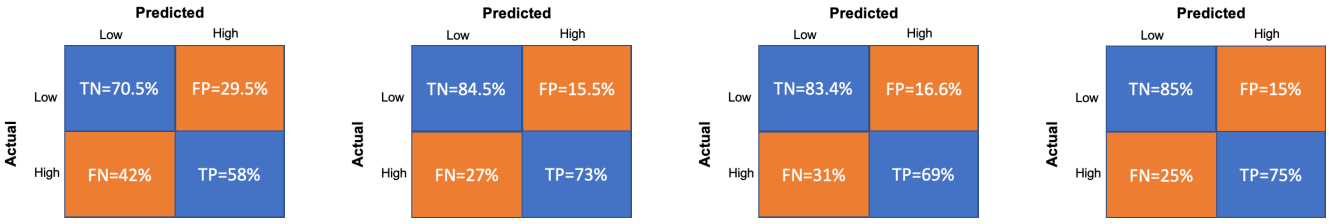
The confusion matrices for the four classifiers (see Figure 5) further confirm that the ensemble and random forest predict both the TPs and TNs with reasonable accuracy, although the performance is marginally worse when predicting high delays. The ensemble classifier correctly predicted 75% of the high-delay samples. Although this is a relatively good accuracy, we need to investigate whether the model can accurately predict high jumps in delay, as these have the worst impact on end-to-end performance.

B. FEATURES IMPORTANCE

To gain more insight into our model, we identify the features that contribute the most to the decision-making of the model. Figure 8 presents the top 10 features, along with their importance. Historical RTTs, taking the first five spots, play the most important role. Additionally, the network features RSSI, RSRP, and RSRQ contribute to discriminating features in the model. We evaluated our model when relying on historical RTTs only; the ARIMA model used a lag order of 5, resulting in 63% accuracy. Therefore, a model that uses only historical RTT data (e.g., moving average, exponential smoothing, and ARIMA [12]) does not work well. Our model also exhibits some spatial dependencies through the population feature based on the probe location.

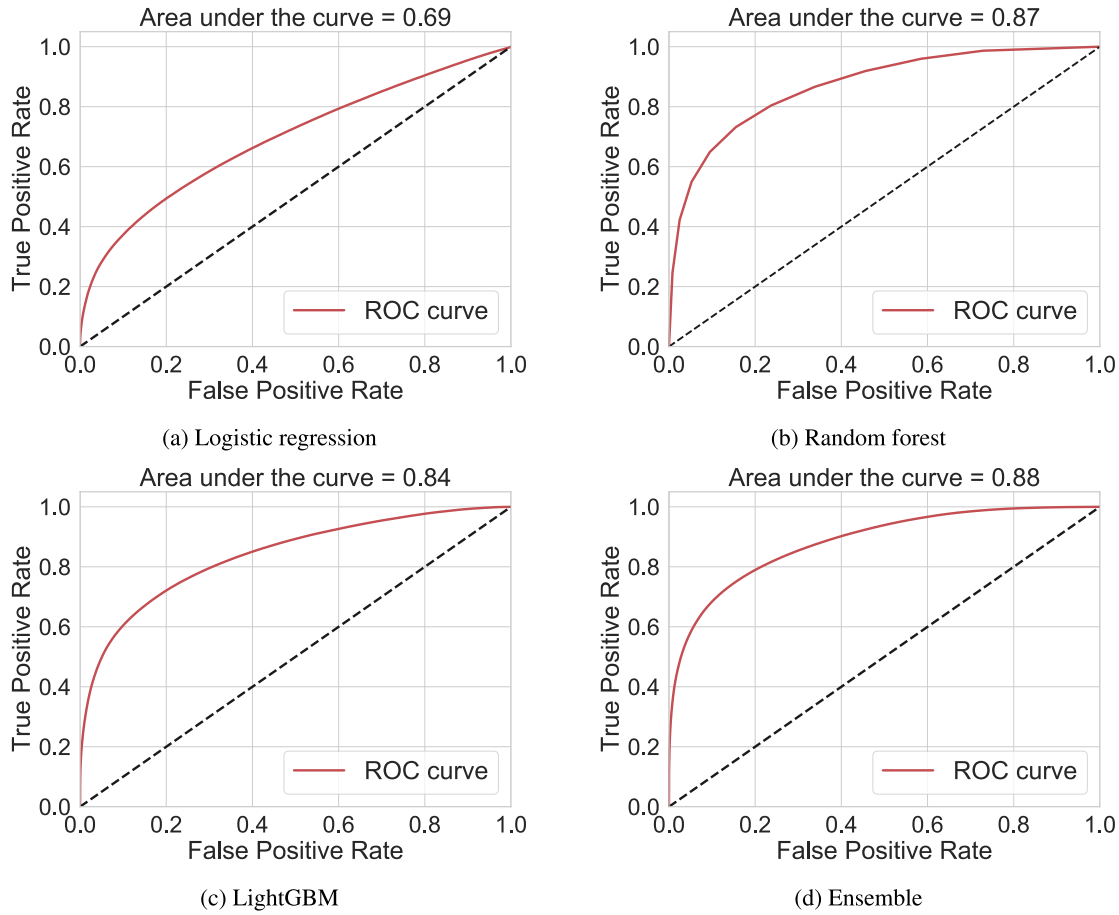
C. MODEL ACCURACY PER PROBE

We now break down our analysis of accuracy per probe, which should provide a more fine-grained idea about the failure of the model. Figure 9(a) depicts the fraction of FNs per each probe, which is the fraction of high delays that are incorrectly predicted. We observed marked differences between the probes. While a sizable majority had an FN rate below 0.1, eight probes had a rate over 0.3. However, there are fewer variations in FPs across probes (see Figure 9(b)), with accuracy below 0.2 for almost all probes. To gain insights into the high variability in the FN rate, we compared the distribution of the number of consecutive seconds with high delays for the two probes with the highest (probe 8) and lowest (probe 12) FN rates. This comparison is motivated by the fact that past RTT values are the most central features. The results show that the node with the lowest FN rate suffers longer periods with high delays as opposed to the node with the highest FN rate. Here, 90% of high-delay episodes last two seconds or shorter. Furthermore, probes with higher FN rates are generally characterized by



(a) Logistic regression (b) Random forest (c) LightGBM (d) Ensemble

FIGURE 5. Confusion matrix for four classifiers.



(a) Logistic regression (b) Random forest (c) LightGBM (d) Ensemble

FIGURE 6. Comparative evaluation of four classifiers based on ROC curve and AUC.

lower delays and less variations in delay, whereas those with medium and lower FN rates suffer higher delays. Accordingly, the classifier fails to predict on-off hikes in delay but performs well for connections with a challenging delay profile.

D. MODEL TRANSFERABILITY

Many machine learning models are limited to a specific context, which necessitates building new models as the context changes. Hence, an important question is whether our classifier is transferable to other network operators. To verify this, the model was used to predict delays for probes from the second operator Op_2 in our dataset, while training it on data from Op_1. Note that all the results above are for Op_1. A blind application results in a poor accuracy of 63%. The main reason for the performance degradation is that the two operators have different delay profiles (see Figure 3). Recall that Op_2 exhibits lower delays with 90% of RTTs lower than 60ms, while the corresponding number for Op_1 is 80ms. Accordingly, when we changed the threshold that separates low and high delay for Op_2 to 60ms, the accuracy of the model increased to 81%, which is similar to the Op_1's case. Figure 10 shows the prediction recall and ROC curves for Op_2, which closely match the corresponding plots for Op_1.

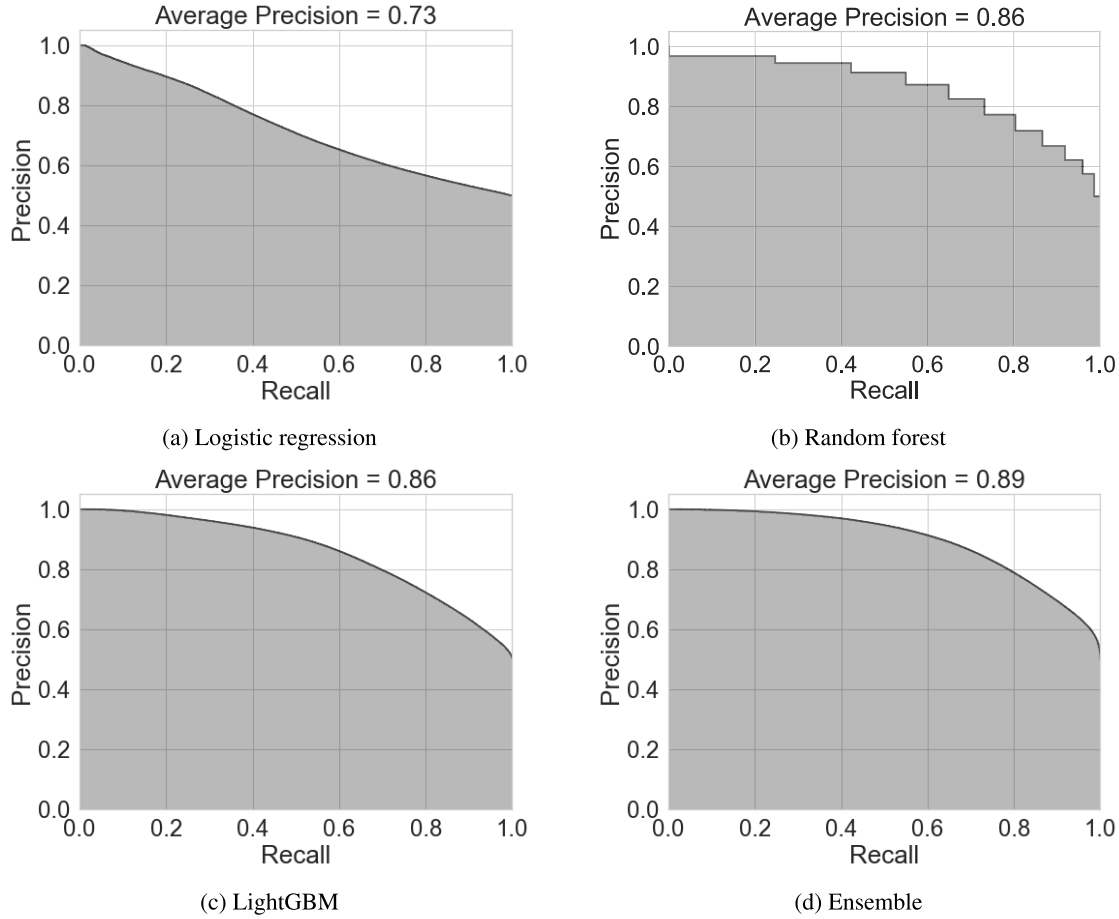


FIGURE 7. Comparative evaluation of four classifiers based on Precision-Recall curve and the average precision.

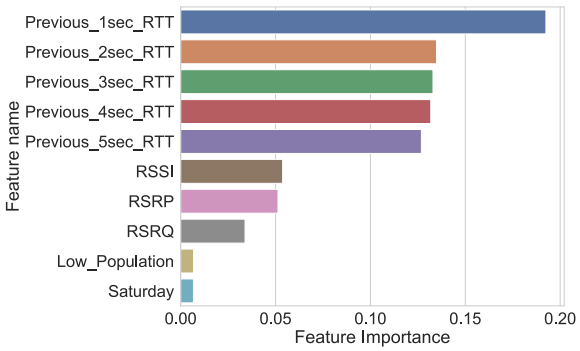


FIGURE 8. Top 10 important features using random forest.

This shows that the model is transferable once it is adjusted to the profile of the new operator.

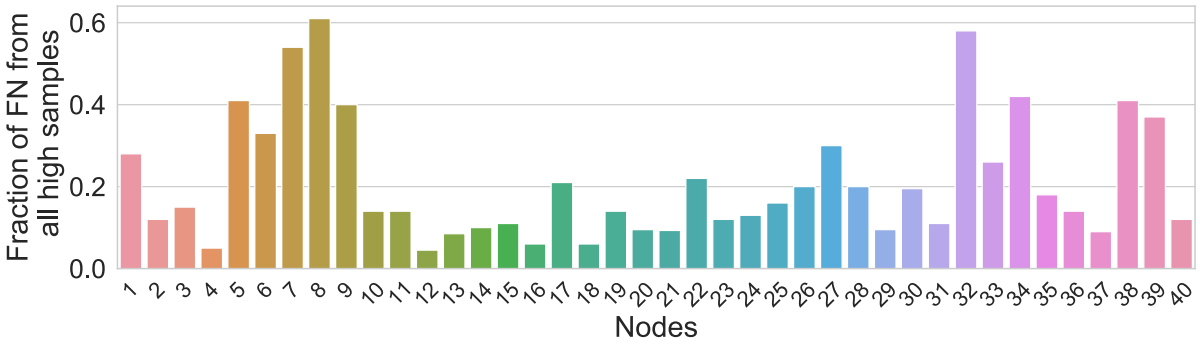
Takeaways. A simple machine learning classifier can predict fairly well whether future delays will be over or below a specific threshold. Our ensemble learning classifier is accurate in 80% of the cases and is able to predict 75% of high-delay instances. Recent RTTs, signal quality, and number of users are the most important discriminating features. Furthermore, the accuracy of the model varies across probes and is a function of the delay profile of the probe. The model is more accurate for probes with high-delay episodes that last longer. Finally, the model is transferable to other contexts that require only minor adjustments.

V. DISSECTING AND INTERPRETING THE MODEL PERFORMANCE

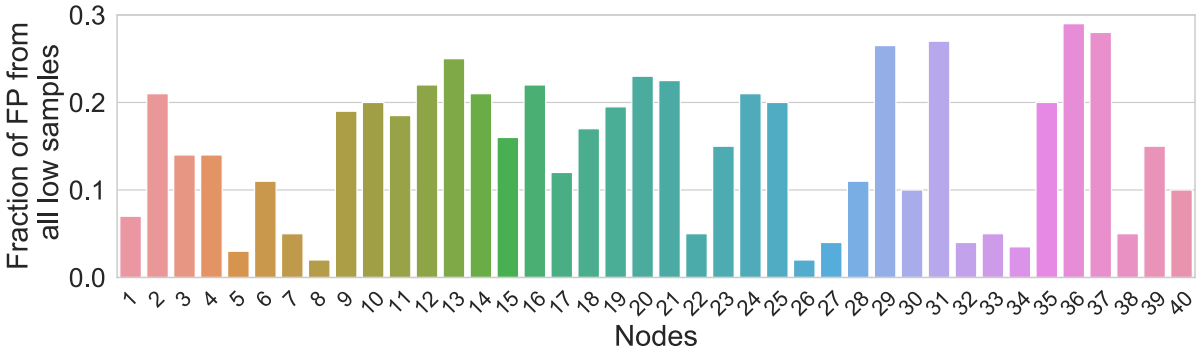
Having seen that high delays can be predicted with a reasonable accuracy, we dig deeper into the misclassified instances. Overall, our model misclassified 20% of the tested samples. These include both the FPs and FNs, which we investigate next. More specifically, we examine the high-importance features of the misclassified samples in comparison with the correctly predicted ones.

A. FALSE NEGATIVES

Recall that by FNs we refer to high delays that are incorrectly predicted as low delays. This is approximately 25% of the total high delays. Considering that historical RTT values are the most important features in our model, we compared the distribution of the previous 1-second and 2-second RTTs for the FNs with those for the TPs. The left panel in Figure 11(a) illustrates that the previous second RTT is evidently higher

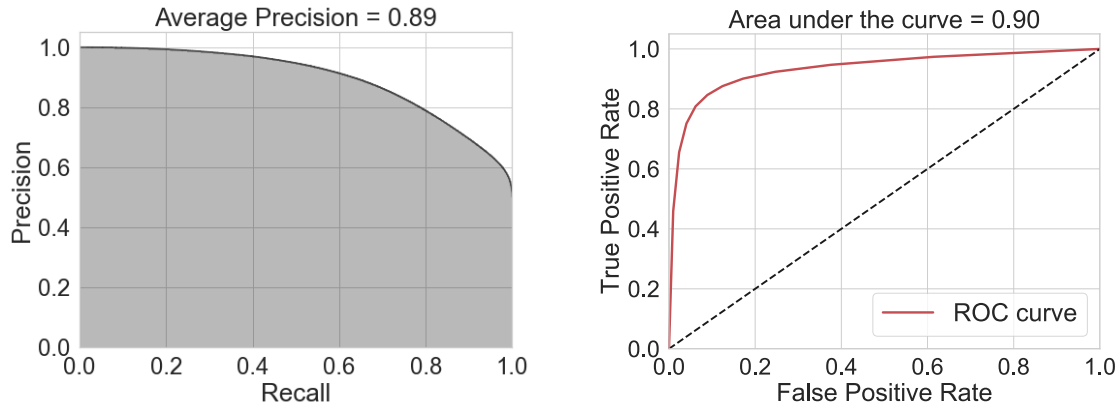


(a) The fraction of the false negatives per probe



(b) The fraction of the false positives per probe

FIGURE 9. Distribution of false negatives and false positives per probe.



(a) (b)

FIGURE 10. Prediction-recall curve and ROC curve for Op_2.

for the TP case. The previous second RTT was in the low category for 90% of FNs, as opposed to 70% of TPs. The right panel shows that, unlike TPs, FNs are often followed by high RTTs. We also compared the distribution of radio metadata (i.e., RSSI, RSRP, and RSRQ) for FNs and TPs, which are almost identical, indicating that differences in these features do not offer further details to explain FNs.[3]

[3]We do not include the respective figures due to space limitations.

Recall from the previous section that FNs appear to increase as the duration of high-delay episodes decreases. To confirm this, we plotted the accuracy as a function of the delay episode duration, as shown in Figure 12. The results indeed confirm the earlier observations, our model predicted only 57% of the high episodes of length one, i.e., spikes and 63% of length two episodes. The accuracy continues to improve as the duration of the high-delay episodes increases. Accordingly, FNs have two characteristics: 1) they often

appear after periods with low delay. That is, an FN sample may be the first sample with a high delay, and 2) they belong to delay episodes that are short.

Interestingly, our model still succeeds in predicting a considerable fraction of delay episodes of lengths one and two, which begs the question of what features help discriminate these short delays. Looking at all features, we found that the RSRQ is the most important discriminator.

Figure 13(a) compares the RSRQ distribution for the cases of length one. The TPs were evidently associated with worse RSRQ. We refer to RSRQ values as good or bad according to LTE RSRQ mapping table defined by 3GPP in [22]. We also analyzed high-delay episodes that lasted for two seconds. Here, we have four cases: 1) both delay samples are predicted correctly as high and they contribute to the true positives, 2) both delay samples are predicted incorrectly as low, 3) the first sample is predicted correctly and the other one is not and 4) lastly the second sample is predicted correctly, and the first one is not. For the first case, both samples have relatively low RSRQ values (see Figure 13(b)), which drives the model to predict them correctly as high. In the second case, both samples have a good or fair RSRQ value (see Figure 13(c)). For case three, both RSRQ values are relatively low, which explains why the first sample is predicted correctly (see Figure 13(d)). Finally, for the fourth case, the RSRQ values are higher for the first sample than for the second one (see Figure 13(e)). These results indicate that RSRQ reliably contributes to flag short delay episodes. A worse RSRQ is indicative of a congested cell. To confirm this, we break the correctly predicted high delays that belong to short episodes of lengths one and two, down to per hour of day. Figure 14 shows this breakdown, where we can clearly see that such delays tend to occur more at peak hours.

B. FALSE POSITIVES

Similarly, we investigated the FPs by examining the distributions of historical RTTs. The plots in Figure 11(b) compare the past second and two seconds RTT for FPs and TNs. We recorded a qualitative difference between FPs and TNs, where a nontrivial fraction of FPs appears to follow high-delay instances, that is, the previous second had a high delay. Looking at the RTTs in the seconds that immediately follow an FP, reveal that these seconds are often associated with low delays.

Takeaways. Short-delay episodes are difficult to predict. RSRQ helps identify short episodes that are likely to be caused by congested cells. These amounts to 57% of the episodes of length one. More frequent measurements, that is, at a frequency less than 1*s*, can help in predicting false negatives. In addition, the model struggles to demarcate the ends of some delay episodes, resulting in false positives.

VI. MODEL STABILITY AND NEED FOR RETRAINING

As machine learning-based models are trained on data collected from the past, they often degrade over time owing to external changes in the environment. Model degradation is often rectified through a system for retraining, that is, keeping the model up to date through training on the data collected during production. In this section, we investigate whether our model remains stable over time and how to rectify performance degradation, if any. Figure 15 shows the performance of an ensemble model trained on the first week of September 2018, which was then tested on data for the following seven weeks, which is the second half of September and the whole of October. The graph shows a clear trend of performance degradation, where the accuracy drops from 73% to 68%. Hence, it is necessary to retrain the model.

To gain insight into how a model may degrade over a small period of time, we monitored the performance of a model trained on the first week of September 2018 and deployed over the following seven weeks. Figure 15 shows the results for a model that was not retrained, a model trained every week, and a model trained every day. Each retraining session used data collected from the previous training session. We observe that a model that does not perform any retraining exhibits a downward trend in performance. Retraining every day shows an improvement over not retraining but still has a downward slope. We believe this is because the model does not have sufficient training data to accurately represent day-to-day changes in mobile delay. Retraining every week shows an improvement over retraining every day, and now shows a slight upward trend in performance. The experiments show that retraining a model helps stabilize the performance but may still change from day to day due to unforeseen circumstances. An example of such a change can be seen on October 15th, where the network had a surge in dropped packets and high delays due to the failure of a central component.

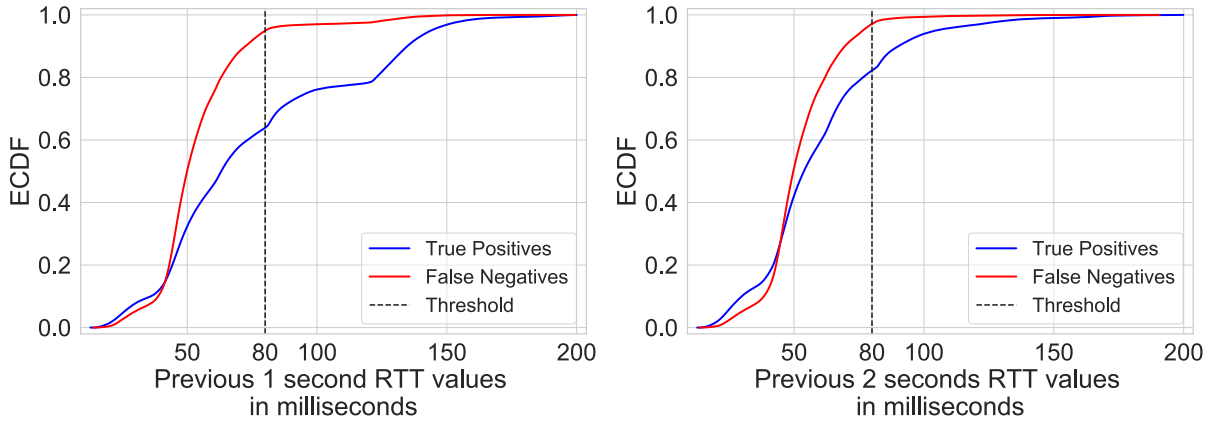
Takeaways. The model performance degrades as time progresses. As expected, retraining can help address this. However, the retraining cycle must be adjusted to include all important patterns in the underlying dataset. We found that a modest weekly cycle performed fairly well.

VII. ARE HIGH DELAYS ALSO PREDICTABLE IN 5G?

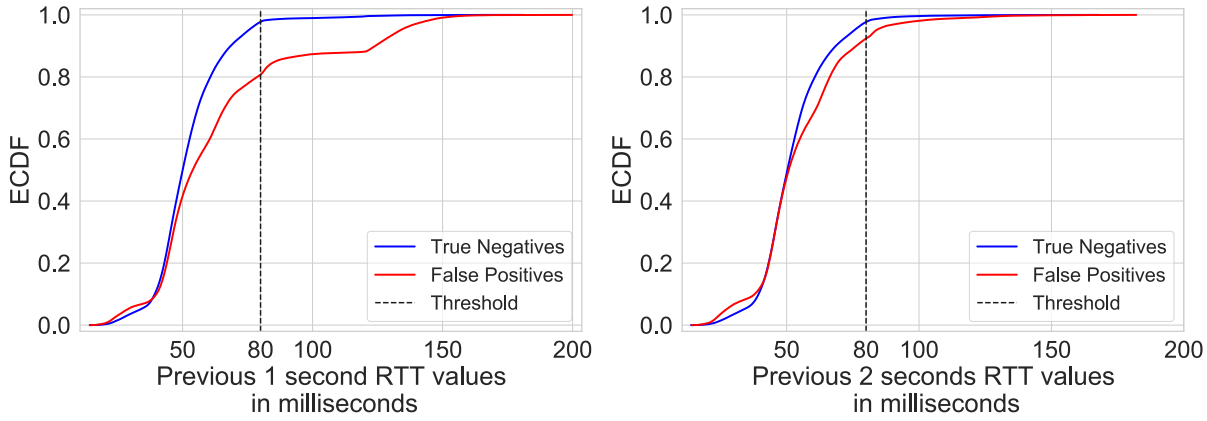
In this section, we examine whether our ensemble classifier can be extended to 5G.

A. DATASET

We collected RTT measurements, following a procedure similar to that for 4G, using three measurement nodes that connect to the newly launched sub-6GHz Non-Stand Alone (NSA) 5G [23] service by *Op*1. These nodes connect to the 5G network using Huawei CPE Pro 2 [24] and commercial subscriptions. Similar to the 4G nodes, the three nodes were placed indoors in an urban environment. In addition to active measurements, the probes collect the connection metadata. The collected metadata involves the same 4G metadata described in Section II and a set of extra metadata. The additional metadata include the modulation coding schemes (MCS) in use (i.e., the number of bits that can be sent in a resource block for both uplink and downlink [25]),



(a) False negatives and true positives



(b) False positives and true negatives

FIGURE 11. The distributions of historical delay features (previous 1 second RTT and previous 2 seconds RTT values) for the false negatives, true positives, false positives and true negatives.

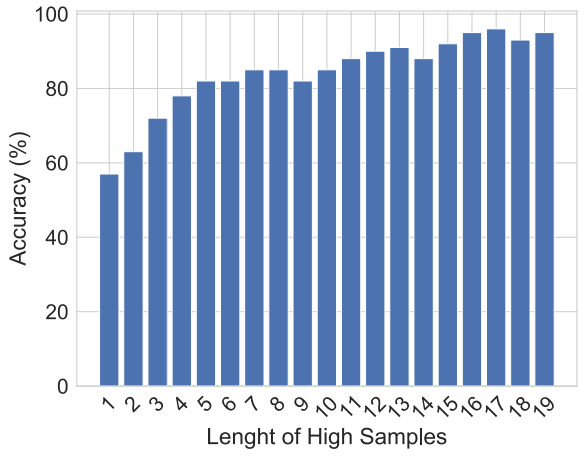


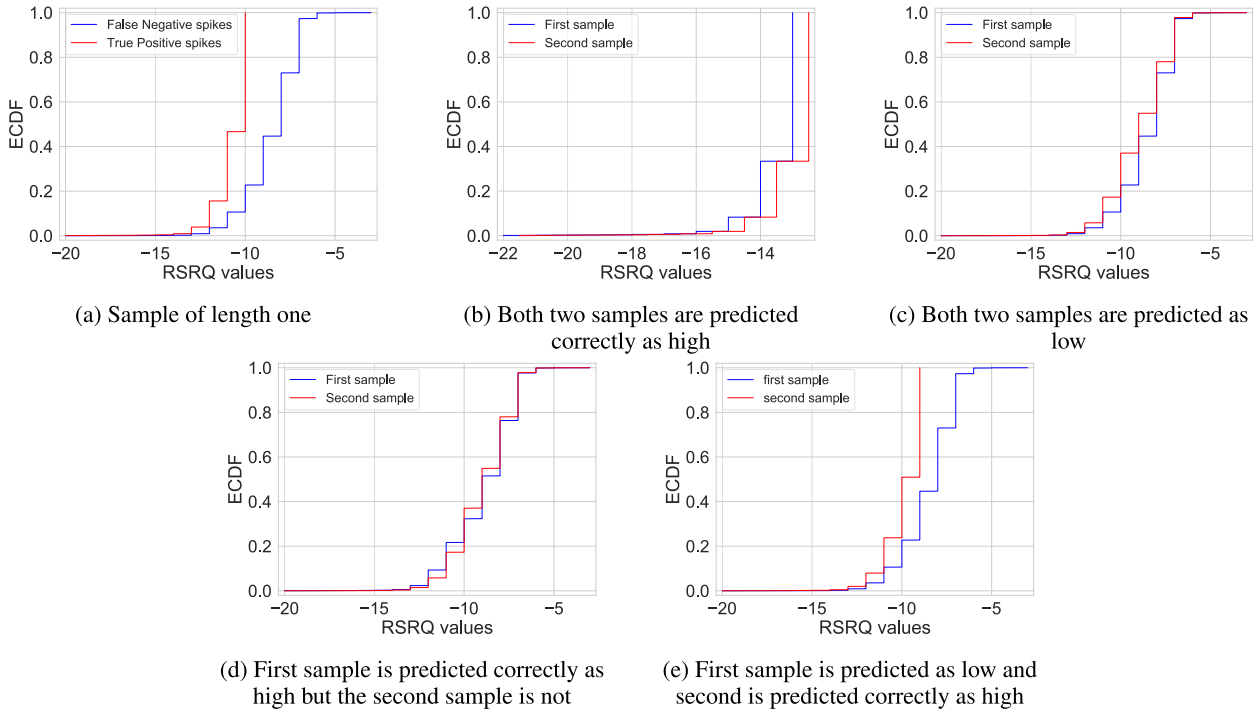
FIGURE 12. Accuracy of the high delay episodes grouped by how long the episode lasts.

the measured power on the physical uplink shared channel, the physical uplink control channel, and the power of the sounding reference signal. The MCS captures the quality of the downlink, whereas the three power measures capture the quality of the uplink. Measurements were conducted in February 2020. After merging the RTT measurements with metadata, we obtained a dataset containing 838,796 samples.

Figure 16 shows the RTT distribution for the 5G data *Op*1. The top 10% RTTs, categorized as high delays, correspond to RTTs exceeding 28*ms*. This is a major improvement over 4G. Note that the 5G NSA still uses the 4G core, albeit with a flattened architecture. However, it deploys a different air interface, that is, 5G New Radio. The measured network delivers 5G NSA over a number of frequencies, but only focuses on the commonly used 3.5 GHz frequency. The flattened architecture and differences in the air interface explain most of the savings in RTT [5].

B. PREDICTION ACCURACY

To verify whether high delays can be predicted equally well on 5G as in 4G, we trained an ensemble classifier with three weeks 5G data and tested it using a one-week dataset. Further, we conducted two experiments: one using the same features as for the LTE in Sec III and another using the extra metadata as features. The first experiment resulted in an accuracy of 83%, while the second experiment achieved an accuracy



(a) Sample of length one

(b) Both two samples are predicted correctly as high

(c) Both two samples are predicted as low

(d) First sample is predicted correctly as high but the second sample is not

(e) First sample is predicted as low and second is predicted correctly as high

FIGURE 13. The distributions of RSRQ of high delay episodes of lengths one and two.

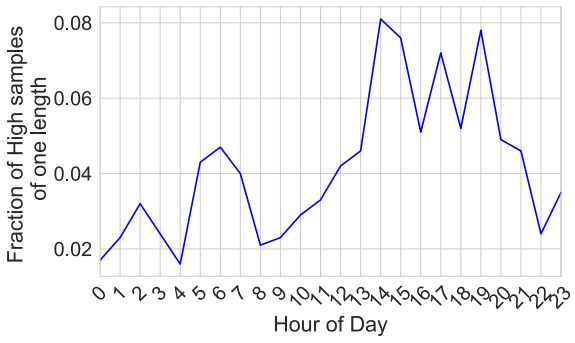


FIGURE 14. The diurnal pattern of short high delay episodes.

of 88%. Figure 17 shows the performance of the second experiment. The model had high precision and an AUC close to 1. From the confusion matrix in Figure 17(c), the model can successfully predict approximately 85% of the high-delay instances.

C. FEATURES IMPORTANCE

Figure 18 presents the top 15 features along with their importance for the 5G model. As for 4G, historical RTTs ranked as the top five important features. Next comes the downlink MCS and RSRP. Note that all the new features bring a non-negligible contribution to the model.

D. MODEL TRANSFERABILITY TO 5G

In Sec. IV, we verified that the model is transferable between operators in the case of 4G data. However, when we evaluated the 4G model on 5G data, this resulted in a significant drop in accuracy of 51%. Figure 19 shows the density plots for 4G data from *Op*1 and *Op*2, which are quite similar and explain why the model is transferable between *Op*1 and *Op*2. On the other hand, the density plot for 5G data for *Op*1 is completely different from the 4G data. Thus, the model from 4G data cannot be used for 5G data. New models should be trained on 5G data to achieve high prediction accuracy.

Takeaways. The ensemble classifier works very well on 5G data, although the distribution of RTT values is very different from that of 4G. The model accurately predicted 85% of the high-delay cases. Enhancing the set of features increases the model accuracy from 83% to 88%, which is a reasonable improvement.

VIII. RELATED WORK

Existing studies have explored several approaches for predicting delay. The authors in [26] discussed different RTT prediction systems and classified them into three classes: a) localisation measurement systems; which use direct RTT measurements to form a structured overlay network to predict RTT, b) network coordinate systems; which use the geometric space to position the actual RTT measurement in order to predict RTT without direct measurement, and finally, c) matrix factorisation systems; that solve a large distance matrix in order to predict RTT. Further, they reviewed the performance, robustness, and security of these system. In the context of Internet, [27] surveyed some techniques for end-to-end Internet delay prediction, including *the time series*

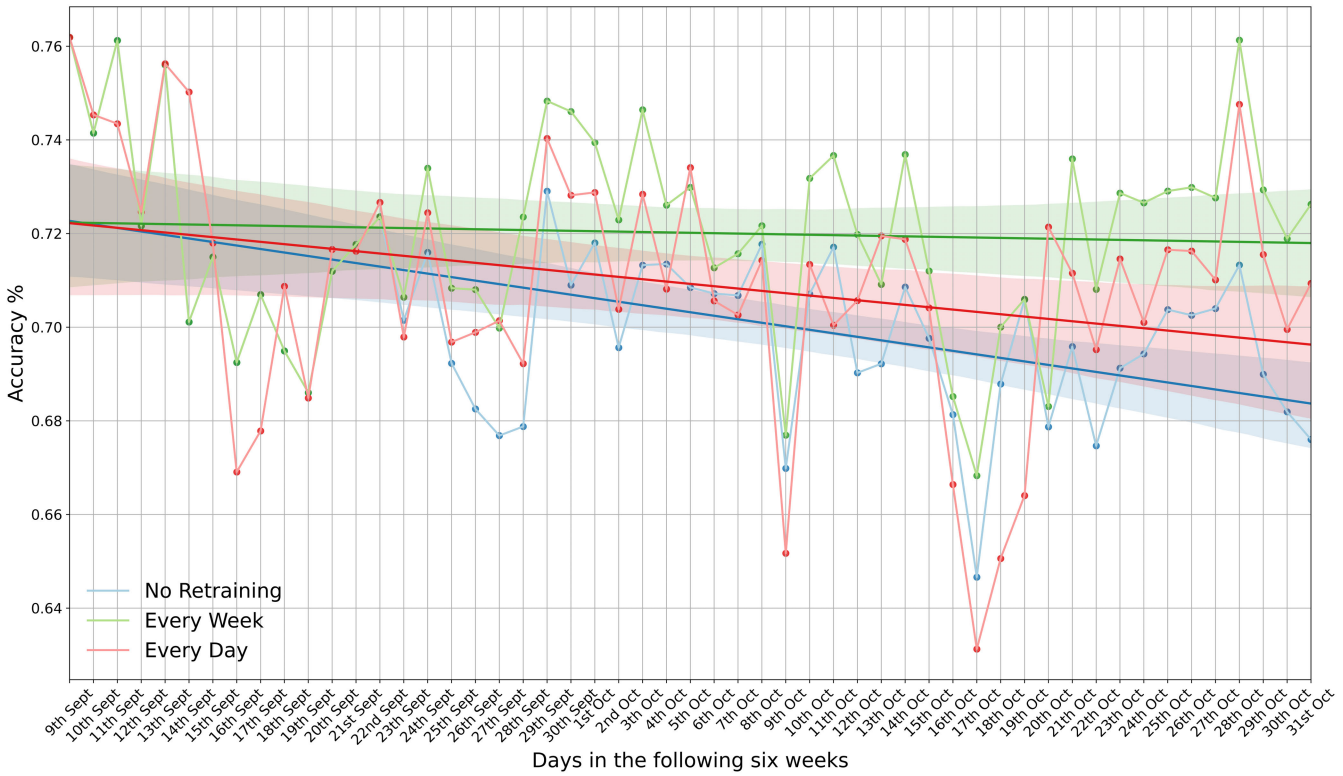


FIGURE 15. The performance of three ensemble models trained on the first week of September 2018 and evaluated over the following seven weeks. One model was not retrained over the evaluation period, one model was retrained every day, and one model was retrained every week.

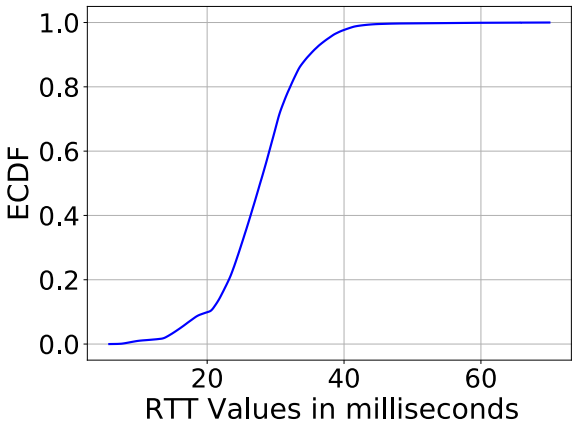


FIGURE 16. Distribution for RTT values for 5G data.

approach, queuing theory, machine learning, and neural networks. Time-series approaches using the autoregressive moving average (ARMA) models have been widely used to predict RTT. [28] presented an autoregressive eXogenous (ARX) model to study the variations in end-to-end packet delay on the Internet. Although, [28] succeeded in using the ARX model to formulate the Internet delay dynamics as a control engineering problem, [29] showed that the ARMA model as a linear time-invariant model is not suitable for predicting the Internet delay owing to the high variations in delay. Further, [30] used a machine-learning technique known as "Experts" framework to estimate the RTT, each of several "experts" provides an estimated value. The weighted average of these estimated values is used to estimate the final RTT, with the weights updated after every RTT measurement.

Recently, deep learning methods that use historical data have been increasingly used for network performance prediction. In the context of delay prediction, [31], [32] used recurrent neural networks (RNNs) to model and predict Internet delays. Although RNNs have proved to be very helpful in understanding delay dynamics, the long training time makes them unpopular for online prediction. To overcome the long training time, [33] proposed a new RNN approach with a minimal gated unit (MGU) to capture temporal features of RTT and reduce the computing cost. The proposed RNNs achieved a root mean square error (RMSE) of 1.543. In addition, [34] presented a hybrid neuro-fuzzy approach for client-cloud server communication round-trip time (RTT) prediction, achieving an accuracy of 79.36%. [35] presented a Markov model with two states to predict the probability density function of RTT instead of the actual value in LTE and WiFi networks. Also, [36] proposed a machine learning regression model to predict RTT for TCP in LTE networks. Then, they discussed how such a model can be used to enable more reliable scheduling between multiple communication paths in the field of automated vehicles. The above studies used different datasets and different ways of formulating the

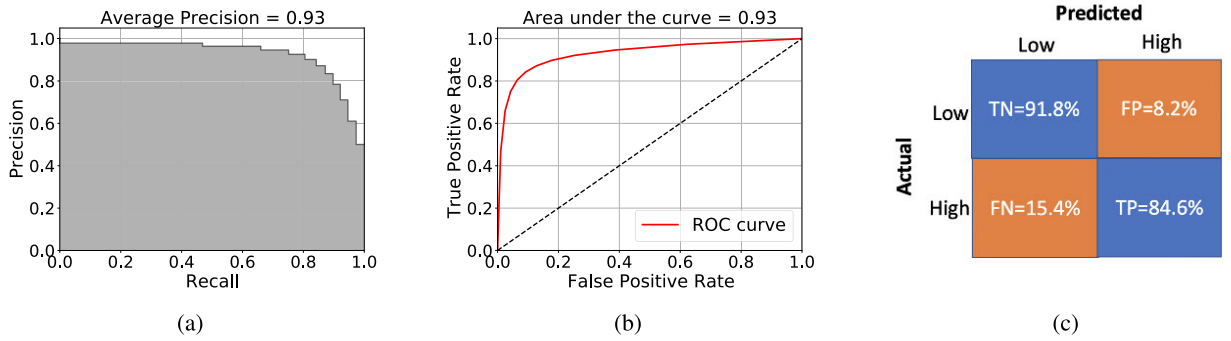


FIGURE 17. Prediction-recall curve, ROC curve, and confusion matrix for 5G data.

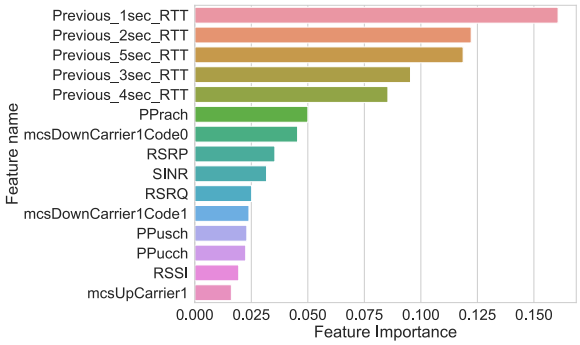


FIGURE 18. Top 15 high important features using random forest for 5G data.

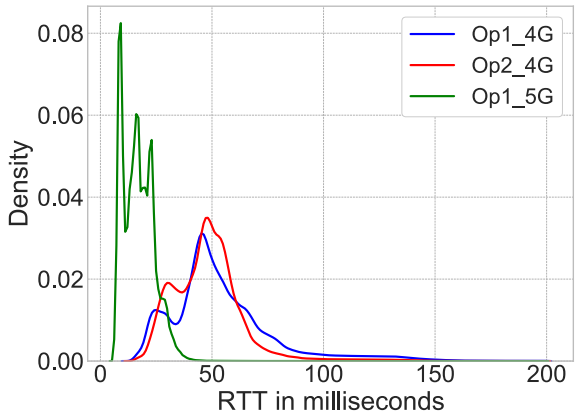


FIGURE 19. Density plots for RTT values for 4G and 5G data.

problem of RTT prediction, which do not allow for direct comparison of results. However, the approach proposed in our paper is close to the work in [21], which used traditional machine learning to predict the latency in mobile broadband networks. [21] revealed that the use of SVM, DT, and LR models to predict RTT in mobile broadband networks does not show high efficiency. The results show that performance of 71% (F1-score) when using DT. In our work, we improve the machine learning model by using an ensemble of different classifiers and incorporating historical RTTs as features. Further, we investigate the model transferability to 5G and identify the cases in which the model fails.

IX. DISCUSSION

We built an ensemble model that can predict the occurrence of high delays with 75% accuracy in 4G data. With high delays, we refer to the worst 10% delay. In the presence of multiple network interfaces that connect to different independent operators, the proposed model can be used to decide which interface to use for sending the next packet (i.e., by a protocol such as MPTCP or multipath QUIC). As a result, we can ensure an RTT within 60*ms* or 80*ms*, depending on the operator, for up to 97.5% of the time. This is a marked improvement over the default of 90%. We extend these results to 5G; by using extra metadata, we can predict 85% of the high delays. Our model captures the relationship between the misclassified samples and the duration of the high-delay episodes. High-delay instances that last for one second are contributing to almost 50% of the high-delay samples that were incorrectly predicted. To improve the accuracy, we need more fine-grained measurements (e.g., every 100*ms*) for RTT to detect such short episodes. However, this implies a trade-off between accuracy and measurement overhead, which we would like to explore in future work.

We identified two key properties of such delays as side products for predicting high delays. First, they tend to cluster time, and second, a non-trivial fraction is related to congested radio links. The clustered nature of high delays suggests that applications with strict delay requirements may need to consider multi-connectivity. A limitation of our model is that it cannot be directly applied to mobility cases. Nevertheless, we believe that many use cases with stringent delay requirements are stationary (e.g., smart meters and Industry 4.0) [1]. Delays for moving users are strongly influenced by handovers and channel fading [37]. The handover decision is highly controlled by RSRP and RSRQ levels [38]. It is not clear whether only these features can accurately help in predicting high delays under mobility. In the future, we plan to investigate this issue.

For our model to be useful, it must be deployable on end devices with relatively limited resources. This implies that we cannot opt for deep learning approaches that perform well in time-series prediction tasks such as recursive neural networks (RNNs). Our model has the ability to recognize temporal patterns without the need to manually craft complex

high-level features. For example, when using random forest in our first dataset training, the size of the single tree saved to the hard drive is approximately 0.6 MB. The memory required for neural network solutions depends on the total number of parameters, gradient, and activation. Recently, [39] presented a comprehensive assessment of the trade-offs between the performance of various machine learning models for binary classification datasets. Their results confirm that our selection of traditional machine learning methods can be more effective in terms of memory and CPU than deep learning-based neural networks.

X. CONCLUSION

We empirically investigated whether RTTs in mobile broadband networks could be accurately predicted. Using measurement data from a large number of probes, we found that a binary ensemble learning-based model can accurately predict delay classes 80% and 88% of the time for 4G and 5G, respectively.

The model is both interpretable and transferable. Furthermore, the model does not require extensive retraining but rather a modest retraining with a weekly cycle. However, it struggles when predicting short delay episodes and, to a lesser extent, by demarcating the end of a delay episode. Despite this, the model performs fairly well. For example, an application using it to anticipate high delays (i.e., the worst 10%) should be able to react positively to 75% of them when using a 4G connection.

Our findings are encouraging and can help inform the scheduling of multipath transport protocols that aim to bound delays. Next, we plan to implement our model in a multipath scheduler and investigate the means to improve the detection of short-lasting delay episodes. In addition, we plan to explore modelling scenarios with high mobility.

REFERENCES

[1] M. Iwamura, "NGMN view on 5G architecture," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, May 2015, pp. 1–5.
[2] Y. Li, Z. Yuan, and C. Peng, "A control-plane perspective on reducing data access latency in LTE networks," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 56–69.
[3] N. Larson, D. Baltrunas, A. Kvalbein, A. Dhamdhere, K. Claffy, and A. Elmokashfi, "Investigating excessive delays in mobile broadband networks," in *Proc. 5th Workshop All Things Cellular, Oper., Appl. Challenges*, Aug. 2015, pp. 51–56.
[4] S.-Y. Lien, S.-L. Shieh, Y. Huang, B. Su, Y.-L. Hsu, and H.-Y. Wei, "5G new radio: Waveform, frame structure, multiple access, and initial access," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 64–71, Jun. 2017.
[5] D. Xu, A. Zhou, X. Zhang, G. Wang, X. Liu, C. An, Y. Shi, L. Liu, and H. Ma, "Understanding operational 5G: A first measurement study on its coverage, performance and energy consumption," in *Proc. Annu. Conf. ACM Special Interest Group Data Commun. Appl., Technol., Archit., Protocols Comput. Commun.*, Jul. 2020, pp. 479–494.
[6] A. Elmokashfi, D. Zhou, and D. Baltrünas, "Adding the next nine: An investigation of mobile broadband networks availability," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 88–100.
[7] B. Han, F. Qian, S. Hao, and L. Ji, "An anatomy of mobile web performance over multipath TCP," in *Proc. 11th ACM Conf. Emerg. Netw. Exp. Technol.*, Dec. 2015, pp. 1–7.
[8] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, *TCP Extensions for Multipath Operation With Multiple Addresses*, document RFC 6824, 2013.
[9] Q. De Coninck and O. Bonaventure, "Multipath QUIC: Design and evaluation," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2017, pp. 160–166.
[10] A. Kvalbein, D. Baltrūnas, K. Evensen, J. Xiang, A. Elmokashfi, and S. Ferlin-Oliveira, "The nornet edge platform for mobile broadband measurements," *Comput. Netw.*, vol. 61, pp. 88–101, Mar. 2014.
[11] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, "Mobileinsight: Extracting and analyzing cellular network information on smartphones," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2016, pp. 202–215.
[12] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.
[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, Jan. 2002.
[14] S.-J. Yen and Y.-S. Lee, "Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset," in *Intelligent Control and Automation*. Berlin, Germany: Springer, 2006, pp. 731–740.
[15] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 112. New York, NY, USA: Springer, 2013.
[16] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 3146–3154. [Online]. Available: http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf
[17] R. Polikar, "Ensemble learning," in *Ensemble Machine Learning*. Boston, MA, USA: Springer, 2012, pp. 1–34.
[18] *Maps and Geodata From Statistics Norway*. Accessed: Nov. 1, 2019. [Online]. Available: https://www.ssb.no/natur-og-miljo/geodata
[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
[20] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, 2020.
[21] A. S. Khatouni, F. Soro, and D. Giordano, "A machine learning application for latency prediction in operational 4G networks," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 71–74.
[22] *3GPP Specification*, document TS 36.133 Release 8, 2013. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2420
[23] *3GPP Specification*, document TR 21.915 Release 15, 2019. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3389
[24] (Feb. 2020). *HUAWEI 5G CPE Pro 2*. [Online]. Available: https://consumer.huawei.com/en/routers/5g-cpe-pro-2/
[25] Y. Wang, W. Liu, and L. Fang, "Adaptive modulation and coding technology in 5G system," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2020, pp. 159–164.
[26] D. Mirkovic, G. Armitage, and P. Branch, "A survey of round trip time prediction systems," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1758–1776, 3rd Quart., 2018.
[27] M. Yang, X. R. Li, and H. Chen, "Predicting internet end-to-end delay: An overview," in *Proc. 36th Southeastern Symp. Syst. Theory*, 2004, pp. 210–214.
[28] E. Kamrani, H. R. Momeni, and A. R. Sharafat, "Modeling internet delay dynamics for teleoperation," in *Proc. IEEE Conf. Control Appl. (CCA)*, 2005, pp. 1528–1533.
[29] P. X. Liu, M. Meng, X. Ye, and J. Gu, "End-to-end delay boundary prediction using maximum entropy principle (MEP) for internet-based teleoperation," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, May 2002, pp. 2701–2706.
[30] B. A. A. Nunes, K. Veenstra, W. Ballenthin, S. Lukin, and K. Obraczka, "A machine learning approach to end-to-end RTT estimation and its application to TCP," in *Proc. 20th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2011, pp. 1–6.
[31] S. Belhaj and M. Tagina, "Modeling and prediction of the internet end-to-end delay using recurrent neural networks," *Proc. J. Netw.*, vol. 4, no. 3, pp. 528–535, 2009.

[32] A. G. Parlos, "Identification of the internet end-to-end delay dynamics using multi-step neuro-predictors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 3, 2002, pp. 2460–2465.
[33] A. Dong, Z. Du, and Z. Yan, "Round trip time prediction using recurrent neural networks with minimal gated unit," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 584–587, Apr. 2019.
[34] R. Damaševičius and T. E. Sidekerskien, "Short time prediction of cloud server round-trip time using a hybrid neuro-fuzzy network," *J. Artif. Intell. Syst.*, vol. 2, no. 1, pp. 133–148, 2020.
[35] S. Yasuda and H. Yoshida, "Prediction of round trip delay for wireless networks by a two-state model," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.
[36] J. Schmid, P. Purucker, M. Schneider, R. V. Zwet, M. Larsen, and A. Höß, "Integration of a RTT prediction into a multi-path communication gateway," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.* Cham, Switzerland: Springer, 2021, pp. 201–212.
[37] H. Deng, C. Peng, A. Fida, J. Meng, and Y. C. Hu, "Mobility support in cellular networks: A measurement study on its configurations and implications," in *Proc. Internet Meas. Conf.*, Oct. 2018, pp. 147–160.
[38] M. Tayyab, X. Gelabert, and R. Jäntti, "A survey on handover management: From LTE to NR," *IEEE Access*, vol. 7, pp. 118907–118930, 2019.
[39] D. Preuveneers, I. Tsingenopoulos, and W. Joosen, "Resource usage and performance trade-offs for machine learning models in smart environments," *Sensors*, vol. 20, no. 4, p. 1176, Feb. 2020.

AZZA H. AHMED (Member, IEEE) received the master's degree from the University of Nottingham, in 2012. She is currently pursuing the Ph.D. degree with the Simula Metropolitan Center for Digital Engineering, Oslo, Norway. Her research interests include communication networks management and control, network performance optimization, network automation, and machine learning to solve networks problems.

STEVEN HICKS (Member, IEEE) received the master's degree from the University of Oslo, Oslo, Norway, in 2018, where he studied explainable machine learning for medical use-cases. He is currently pursuing the Ph.D. degree with SimulaMet, Oslo. His research interests include machine learning, explainable artificial intelligence, computer vision, and medical multimedia.

MICHAEL ALEXANDER RIEGLER received the Ph.D. degree from the Department of Informatics, University of Oslo, Oslo, Norway, in 2015. He is currently working as a Chief Research Scientist at SimulaMet, Oslo. His research interests include machine learning, video analysis and understanding, image processing, image retrieval, crowdsourcing, social computing, and user intentions.

AHMED ELMOKASHFI (Member, IEEE) received the Ph.D. degree from the University of Oslo, Oslo, Norway, in 2011. He is currently a Research Professor at the Simula Metropolitan Center for Digital Engineering, Oslo. He is also working as the Head of the Center for Resilient Networks and Applications (CRNA), which is part of the Simula Metropolitan Centre, which is funded by the Norwegian Ministry of Transport and Communication. His research interests include network measurements and performance. In particular, he focused on studying resilience, scalability, and evolution of the internet infrastructure; the measurement and quantification of robustness in mobile broadband networks; and the understanding of dynamical complex systems. Over the past few years, he has been leading and contributing to the development, operation, and management of the NorNet testbed infrastructure, which is a countrywide measurement setup for monitoring the performance of mobile broadband networks in Norway.