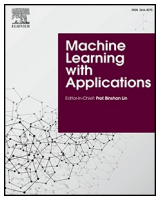
Machine Learning with Applications 17 (2024) 100564

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa



# Ensemble prediction of RRC session duration in real-world NR/LTE networks

Roopesh Kumar Polaganga [a,b,*], Qilian Liang [c]

[a] T-Mobile US Inc, Bellevue, WA, USA 98012
[b] The University of Texas at Arlington, Arlington, TX, USA 76010
[c] Fellow, IEEE, The University of Texas at Arlington, Arlington, TX, USA 76010

ARTICLE INFO

*Keywords:*
Machine learning (ML), Weighted ensemble learning
5G new radio (NR)
Long term evolution (LTE)
6G
Fixed wireless access (FWA)
Live telecommunication networks, Auto-gluon

ABSTRACT

In the rapidly evolving realm of telecommunications, Machine Learning (ML) stands as a key driver for intelligent 6 G networks, leveraging diverse datasets to optimize real-time network parameters. This transition seamlessly extends from 4 G LTE and 5 G NR to 6 G, with ML insights from existing networks, specifically in predicting RRC session durations. This work introduces a novel use of weighted ensemble approach using AutoGluon library, employing multiple base models for accurate prediction of user session durations in real-world LTE and NR networks. Comparative analysis reveals superior accuracy in LTE, with 'Data Volume' as a crucial feature due to its direct impact on network load and user experience. Notably, NR sessions, marked by extended durations, reflect unique patterns attributed to Fixed Wireless Access (FWA) devices. An ablation study underscores the weighted ensemble's superior performance. This study highlights the need for techniques like data categorization to enhance prediction accuracies for evolving technologies, providing insights for enhanced adaptability in ML-based prediction models for the next network generation.

## 1. Introduction

Amidst the revolutionary advancements in wireless communication systems, Machine Learning (ML) has emerged as a driving force reshaping the landscape of innovation. As we navigate the transition from 4 G LTE, known for its efficiency and reliability in providing high-speed mobile internet, to 5 G NR, which offers significantly higher data rates and lower latency, we lay the groundwork for the evolution of 6 G networks. ML takes center stage in this transition, introducing unparalleled capabilities and intelligent solutions (Rekkas et, al., 2021). Its adeptness in analyzing extensive datasets, discerning intricate patterns, and executing data-driven decisions establishes ML as a cornerstone for gaining network insights. Beyond this, it plays a pivotal role in predicting user behaviors, thereby optimizing network performance, and elevating the overall efficiency of telecommunication (telco) systems.

ML techniques can be broadly classified into regression, classification, and clustering tasks, each serving distinct purposes in handling various types of data and challenges. Regression involves predicting continuous numerical values, making it ideal for scenarios where the output is a quantitative measure. Classification, conversely, assigns data points to predefined categories, making it suitable for tasks with categorical outcomes. Clustering seeks to identify inherent patterns or groupings within data without predefined labels. The choice between these tasks depends on the nature of the problem at hand. However, the inherent diversity of techniques and algorithms poses a challenge in determining the optimal model for specific use cases. Navigating through the intricacies of selecting suitable models for diverse applications within the telco industry can be a complex undertaking. Nevertheless, solutions such as Automated Machine Learning (AutoML) play a pivotal role in mitigating this complexity.

Anticipating changes in user behavior patterns, informed by early predictions of session duration, is invaluable for operators in tailoring effective management strategies and mitigating operational risks. To attain this objective, operators can leverage insights gleaned from historical mobile broadband (MBB) records within the telco domain. While existing research, such as (Luo et al., 2016; Wilhelmi et al., 2021; Brezov et al., 2023) has primarily focused on classification problems or simulator data, our work addresses a critical gap by emphasizing the significance of predicting RRC session duration as a regression problem. Leveraging AutoGluon's AutoML approach on real-world network data, our study uniquely contributes to optimizing network performance and ensuring adaptability in evolving wireless technologies. This approach

* Corresponding author.
*E-mail addresses:* RoopeshKumar.Polaganga@Mavs.Uta.edu (R.K. Polaganga), Liang@Uta.edu (Q. Liang).

https://doi.org/10.1016/j.mlwa.2024.100564
Received 2 May 2024; Received in revised form 31 May 2024; Accepted 3 June 2024
Available online 6 June 2024

distinguishes our work in the telecommunication domain, providing a novel perspective on regression-based predictions for user's RRC session duration.

The following sections of the paper are structured as follows: The remainder of Section I introduces the weighted ensemble approach and the AutoGluon library. Section II provides detailed insights into the real-world 5 G NR and LTE network data used in this study, along with the measured performance metrics. In Section III, the prediction results for both technologies are presented, including a comparative analysis. Finally, Section IV offers concluding remarks and outlines potential avenues for future research.

### 1.1. Weighted ensemble learning

Within the dynamic landscape of telecommunications, where precision, adaptability, and performance are paramount, the significance of ensemble methods becomes particularly pronounced (Luo et al., 2016). Ensemble models harness the collective intelligence of multiple machine learning models to enhance predictive accuracy and robustness (Breiman, 1996). By combining diverse models, such as decision trees, random forests, gradient boosting machines, and neural networks, ensemble methods mitigate individual model biases and errors. For instance, a random forest aggregates predictions from multiple decision trees, while gradient boosting optimally combines weak learners to form a strong predictor (Freund et al., 1996). Fig. 1 provides an overview of weighted ensemble architecture where N number of base models provide their corresponding predictions which are aggregated within ensemble meta-model by considering multiple weight optimization schemes. These weight optimization schemes can depend on each base model's accuracy of specific target metric that is under consideration or even based on user preferred biases (How et al., 2023). Such synergy of these models in an ensemble not only improves performance but also provides a versatile framework applicable across various domains, from finance to healthcare and beyond (Dietterich, 2000).

While traditional ensemble models treat each base model equally, weighted ensembles assign varying degrees of influence to individual models, emphasizing the strengths of each while mitigating potential weaknesses. This tailored combination not only enhances predictive accuracy but also facilitates a more nuanced understanding of complex

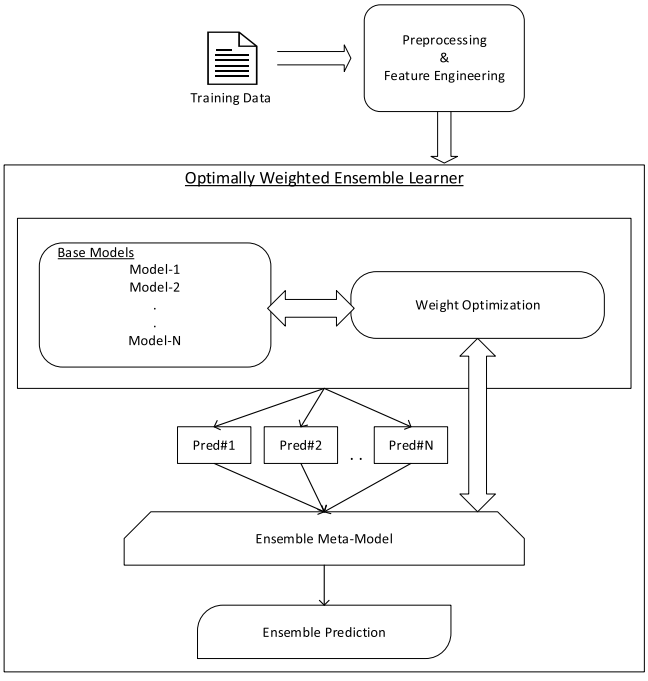



**Fig. 1.** Overview of Weighted Ensemble Architecture.

network behaviors and user interactions. In scenarios where data with distinct patterns and anomalies may exist, the versatility of a weighted ensemble approach shines (Mohr et al., 2018). This enables a finer control over the contribution of each model, leading to potentially improved predictive accuracy and adaptability to diverse datasets. Thus, in the pursuit of optimizing machine learning models for telco applications, a weighted ensemble approach emerges as a strategic and effective means of combining diverse models (Brezov et al., 2023).

### 1.2. AutoGluon-tabular

AutoGluon (developed by the Apache MXNet community) represents a cutting-edge AutoML framework designed to streamline the model development process. With a focus on enhancing accessibility and scalability, AutoGluon-Tabular (which is referred to as AutoGluon here for simplicity) is specific to tabular data and automates critical aspects of ML workflows, including model selection, hyperparameter tuning, and feature engineering (Erickson et al., 2020). Its robust capabilities are particularly beneficial in the communication networks, where the complexities of network optimization, predictive maintenance, and personalized user experiences demand efficient and powerful ML solutions (Mohr et al., 2018). AutoGluon's automated model selection and hyperparameter tuning align harmoniously with the principles of weighted ensembles, fostering a symbiotic relationship that optimally leverages diverse models for enhanced predictive performance (Erickson et al., 2020; Van der Laan et al., 2007).

In the context of machine learning ensemble techniques, multi-layer stacking, also known as stacked ensembles, is a strategy where predictions from multiple models are combined in a hierarchical or layered fashion. This approach aims to leverage the diverse strengths of individual models by having multiple layers of ensembles. As illustrated in Fig. 2, AutoGluon's multi-layer stacking works starts with a set of diverse base models that may use different algorithms or configurations. These models are trained on the training dataset. At first layer of stacking, predictions are made on the validation set using these base models. Train a meta-model (shown as concatenation in Fig. 2) on the validation set using the predictions from the base models as features. The meta-model learns to combine or weight the predictions of the base models, considering their individual strengths and weaknesses (Caruana et al., 2004). At second optional layer, stacking process is extend to additional layers if needed. Predictions from the first layer are used as features for training another meta-model. This process can be repeated for multiple layers, each learning to combine predictions from the previous layer. The final ensemble is typically a combination of predictions from the top-level meta-model and possibly some base models. The weights assigned to each model or layer in the ensemble are determined during the training process based on their performance on the validation set (Erickson et al., 2020). Such stacking approach provides benefits like diverse representations, hierarchical learning along with increased model robustness.

AutoGluon's comprehensive training strategy is outlined in Algorithm 1, after preprocessing the data, each stacking layer is allocated a time budget denoted as Ttotal/L where Ttotal represents the total time allocated to perform prediction while L represent the number of stacking layers. For this work, Ttotal is set to 48hours and doesn't come into effect. It initially estimates the required training time, and if it exceeds the remaining time for the current layer, the process advances to the next stacking layer. AutoGluon further improves its stacking performance by utilizing all the available data for both training and validation, through k-fold ensemble bagging of all models at all layers of the stack. Also called cross-validated committees (Parmanto et al., 1996), k-fold bagging is a simple ensemble method that reduces variance in the resulting predictions. This is achieved by randomly partitioning the data into k-disjoint chunks and subsequently training copies of a model with a different data k chunk held-out from each copy. AutoGluon bags all models, and each model is asked to produce out-of-fold (OOF)

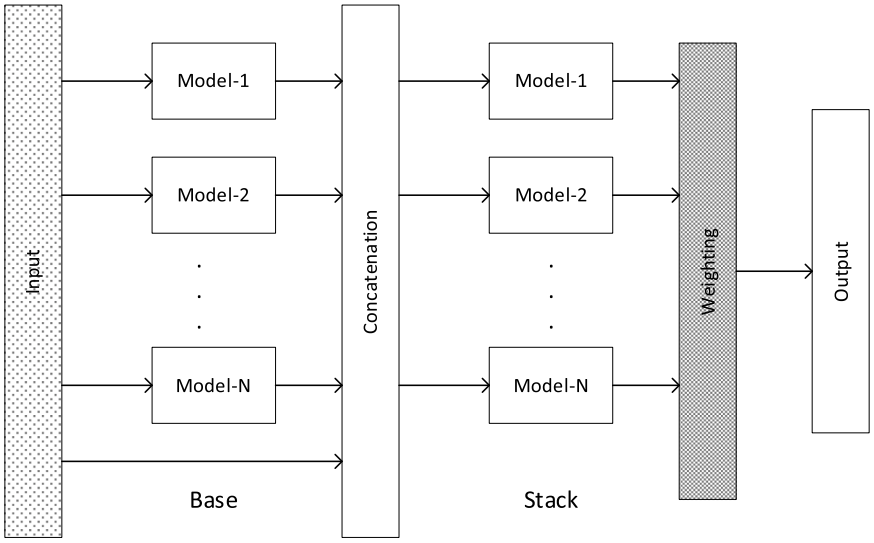



**Fig. 2.** AutoGluon's Multi-Layer Stacking Framework.

**Algorithm 1**
AutoGluon-tabular training strategy.

(multi-layer stack ensembling ± $n$- repeated $k$-fold bagging)
**Require**: data $(X, Y)$, family of models $\mathscr{M}$, # of layers $L$
1: Preprocess data to extract features
2: **for** $l = 1$ **to** $L$ **do** {stacking}
3: **for** $i = 1$ **to** $n$ **do** {$n$-repeated}
4: Randomly split data into $k$ chunks $\{X^j, Y^j\}_{j=1}^{k}$
5: **for** $j = 1$ **to** $k$ **do** {$k$-fold bagging}
6: **for each** model type $m$ in $\mathscr{M}$ **do**
7: Train a type-$m$ model on $X^{-j}, Y^{-j}$
8: Make predictions $\widehat{Y}_{m,i}^{j}$ on OOF data $X^j$
9: **end for**
10: **end for**
11: **end for**
12: Average OOF predictions $\widehat{Y}_m = \left\{\frac{1}{n}\sum_i \widehat{Y}_{m,i}^{j}\right\}_{j=1}^{k}$
13: X ← concatenate $(X, \{\widehat{Y}_m\}_{m\in\mathscr{M}})$
14: **end for**

predictions on the chunk it did not see during training. As every training example is OOF for one of the bagged model copies, this allows us to obtain OOF predictions from every model for every training example.

The prediction from the weighted ensemble model in is represented in (1) where $\widehat{y}ensemble$ is the final prediction, $N$ is the number of base models, $w_i$ is the weight assigned to the i$^{th}$ base model, and $\widehat{y}_i$ is the prediction of the i$^{th}$ base model.

$$\widehat{y}_{ensemble} = \sum_{i=1}^{N} w_i . \widehat{y}_i \tag{1}$$

Model further optimizes the weights of base models based on performance metrics. A simplified weight optimization equation is shown in (2)

$$w_i = \frac{f_i(metric)}{\sum_{j=1}^{N} f_i(metric)} \tag{2}$$

where $f_i(metric)$ is the performance of the $i^{th}$ model on the chosen metric. For this work, $f_i(metric)$ is chosen to be Mean Square Error (MSE) for optimization. By using the inverse of the MSE, better-performing models (with lower MSE) receive higher weights, ensuring that the ensemble promotes models with better performance. For example, consider two models with MSE values of 0.1 and 0.2, respectively. The weights for these models would be calculated as follows: For the model with MSE of 0.1: $w_1 = (1/0.1)/(1/0.1) + (1/0.2) = 0.67$ and for the model with MSE of 0.2: $w_2 = (1/0.2)/(1/0.1) + (1/0.2) = 0.33$. This calculation shows that the model with the lower MSE receives a higher weight, promoting better-performing models in the ensemble.

To ensure framework's predictability, models are promptly saved to disk after each training for fault tolerance. Such approach guarantees the ability to produce predictions as long as at least one model on one-fold can be trained within the allotted time. Checkpointing intermediate iterations of sequentially trained models, enables AutoGluon to generate models under stringent time limits which is crucial for mobile communication type applications (Song et, al., 2022). Additionally, anticipating potential training failures, the framework skips to the next model in such events. Unlike several AutoML frameworks that concurrently train multiple models on the same instance, AutoGluon adopts a sequential training approach. It relies on individual implementations of models to efficiently leverage multiple cores, allowing successful training on larger datasets without encountering frequent out-of-memory errors as observed in parallel training scenarios. It bags all models, and each model is asked to produce out-of-fold (OOF) predictions on the chunk it did not see during training. As every training example is OOF for one of the bagged model copies, this allows to obtain OOF predictions from every model for every training example.

### 1.3. Base models

Eight base models play a pivotal role in this study, encompassing ensemble methods, boosting algorithms, deep neural networks, and traditional regression techniques, collectively contributing to a comprehensive and diverse predictive framework. The models were selected based on their proven effectiveness in regression tasks, their ability to complement each other by capturing different aspects of the data, and their diversity in learning approaches. Specifically, we chose models with varying complexity and mechanisms, such as linear regression for its simplicity and interpretability, decision trees for their ability to handle non-linear relationships, and neural networks for their capacity to model complex patterns. This diversity ensures a robust ensemble capable of handling the intricate and varied nature of the data, ultimately improving the overall prediction performance. Ensemble based models include Random Forest (RF) and Extremely Randomized Trees (XT) that combine multiple models to enhance predictive accuracy by aggregating their outputs. Random Forest is a popular ensemble learning method that constructs a multitude of decision trees and combines their outputs to improve accuracy and reduce overfitting, offering robust performance across diverse datasets (Upadhyay et, al., 2022). Like RF, XT further diversifies the learning process by randomizing the feature selection for each split in the decision trees, enhancing robustness and reducing variance (Sagi & Rokach, 2021).

Boosting algorithms encompass XGBoost (XGB), LightGBM (GBM), and CatBoost (CAT) models. XGB is a gradient boosting framework designed to handle diverse data types and exhibit exceptional predictive power known for its efficiency and performance. Its core strength lies in sequentially combining weak learners, optimizing the model through gradient-based boosting (Sagi & Rokach, 2021; Ke et al., 2017b). GBM is another gradient boosting framework that excels in scalability and speed, making it particularly suited for large datasets like in telco domain. Its unique feature is the efficient implementation of tree-based learning, leading to faster training times and reduced memory consumption (Ke et al., 2017a). CAT stands out for its ability to handle categorical features seamlessly. It employs a robust algorithm that minimizes the need for extensive pre-processing, making it an ideal choice for real-world datasets with mixed data types (Dorogush et al., 2018).

On the neural network front, FastAI and Neural Networks in PyTorch (NN_TORCH) represents novel architectures capable of learning intricate patterns and representations from complex tabular data. FastAI framework built on PyTorch simplifies complex deep learning tasks, providing a high-level interface for rapid experimentation and model deployment. It empowers users to achieve state-of-the-art results with minimal code (Mendoza et al., 2016). NN_TORCH model implemented in PyTorch as well allows for fine-grained control over the architecture and training process. It's the PyTorch's flexibility that makes it a popular choice for researchers and practitioners in deep learning (Guo & Berkhahn, 2016).

AutoGluon uses a feedforward neural network architecture that is suitable for tabular data as shown in Fig. 3. The network contains separate input layers for numerical and categorical features. Numerical features are directly connected to dense layers, while categorical features are first embedded then connected. The network contains dense blocks with batch normalization and ReLU activation. It has dropout for regularization. The last dense layer outputs the predictions. This architecture allows the network to learn relationships between mixed numerical and categorical features for tabular data (Liang et al., 2019). AutoGluon trains the neural network as one of its base models. It then ensembles the neural network with other base models like tree-based models. This provides diversity since neural networks and trees have different types of decision boundaries. The ensemble usually performs better than any individual model. In summary, AutoGluon's neural network architecture and training approach allow it to leverage deep learning for tabular data, while also benefiting from ensemble methods for robust and accurate AutoML (Kotthoff et, al., 2017).

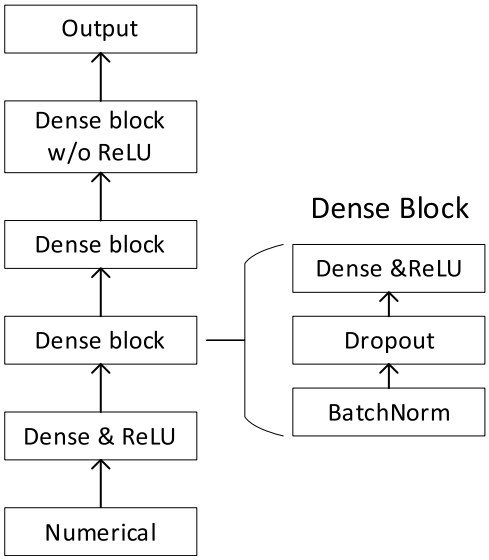



**Fig. 3.** AutoGluon's Neural Network Architecture.

Linear Regression (LR) serves as a foundational method for predicting continuous outcomes by modeling the relationship between dependent and independent variables through a linear equation. Its simplicity and transparency make it a valuable tool for certain prediction tasks. All these base models are exemplars used predominantly for regression tasks, contributing to predictive modeling across various domains by capturing intricate patterns and relationships within datasets (Wong & Michaels, 2022).

## 2. Experimental setup

### 2.1. Real-world data

The datasets employed in this research stem from a live LTE and NR network belonging to a US-based mobile network operator. Capturing session data records from both eNBs and gNBs, the comprehensive dataset is built by aggregating per unique data session. Encompassing user sessions throughout a typical weekday in December 2023 over a 24-hour period, the data is sourced from diverse sites in the Seattle, WA area, presenting varying load scenarios across urban and rural landscapes. In the NR data, implemented in non-standalone (NSA) mode, LTE-specific features are included alongside NR-specific attributes.

Entries are meticulously filtered for non-guaranteed bit rate (non-GBR) class channel quality indicators. Both LTE and NR technologies represent typical data sessions excluding voice. With 5 G implemented as NSA, lacking native voice or Guaranteed Bit Rate (GBR) services, data sessions are meticulously collected to maintain parity between both technologies. The NR network boasts mid-band NR layers featuring N41 (2500 MHz) as its mid-band layer and N71 (600 MHz) as its low-band NR layer. LTE, on the other hand, is implemented across various layers, including mid-band layers of AWS (2100 MHz) and PCS (1900 MHz), and low-band layers of B12 (700 MHz) and B71 (600 MHz). As all considered sites belong to a single RAN vendor with a consistent network configuration, shared NR, and LTE feature sets guarantee data consistency across both technologies. The final dataset from Session Records comprises approximately 25,000 entries for LTE and about 36,000 sessions for NR, sampled after eliminating entries with missing network attributes and ensuring data consistency. Table 1 provides a detailed list of all features collected in the real-world dataset per technology, marked as 'X' where applicable and 'N/A' where not applicable. Additionally, 5 G introduces 10 extra features highlighted at the end of the table, distinguishing it from LTE. To handle missing data and

**Table 1**
List of features collected in real-world datasets.

| Feature | NR | LTE |
|---|---|---|
| Subscriber Identifier | X | X |
| Device Software [SVN] | X | X |
| Device Model | X | X |
| Device Make | X | X |
| Service Type | X | X |
| Start Time | X | X |
| Environment [Indoor / Outdoor] | X | X |
| End Time | X | X |
| Start Type | X | X |
| Start eNB | X | X |
| Establishment Cause | X | X |
| RRC Setup Result | X | X |
| LTE Setup Time | X | X |
| S1 Release Cause | X | X |
| RSRP [dBm] | X | X |
| RSRQ [dB] | X | X |
| Start Timing Advance [Miles] | X | X |
| UE Category | X | X |
| QCI List | X | X |
| ARP List | X | X |
| UE Power Headroom [dB] | X | X |
| PUSCH SINR [dB] | X | X |
| Mean MAC Throughput UL [kbps] | X | X |
| Mean MAC Throughput DL [kbps] | X | X |
| Mean CQI | X | X |
| MAC Volume DL [bytes] | X | X |
| MAC Volume UL [bytes] | X | X |
| Max Number of LTE Carrier Components during aggregation | X | X |
| Avg Number of LTE Carrier Components during aggregation | X | X |
| 5 G EN-DC Downlink Volume [bytes] | X | N/A |
| 5 G EN-DC Uplink Volume [bytes] | X | N/A |
| 5 G EN-DC Downlink Throughput [kbps] | X | N/A |
| 5 G EN-DC Uplink Throughput [kbps] | X | N/A |
| 5 G NR RSRP [dBm] | X | N/A |
| 5 G NR RSRQ [dB] | X | N/A |
| 5 G NR DL SINR [dB] | X | N/A |
| 5 G EN-DC Setup Time | X | N/A |
| Max Number of NR Carrier Components | X | N/A |
| Avg Number of NR Carrier Components | X | N/A |

outliers, we performed data cleaning steps such as removing entries with missing values in critical columns and applying interquartile range (IQR) methods to detect and eliminate outliers (Vinutha et al., 2018).

Duration is calculated as difference between start time and end time features. To provide a contextual understanding, NR data in comparison to LTE has relatively larger duration interval with average session duration of 66.65 s while minimum and maximum are 0.76 s and 64,088.21 s (~17.8 h) respectively. For LTE data, average session duration is 12.7 s while minimum is 0.53 s and maximum is 4068.29 s (~1.1 h). The computational infrastructure employed for executing predictions and obtaining performance results on these datasets comprises a virtual Central Processing Unit (vCPU) configuration of 64 cores, coupled with a Memory allocation of 52 gigabytes.

### 2.2. Evaluation metrics

The evaluation metrics serve as crucial benchmarks to gauge the efficacy of the predictive models, offering insights into their performance across various dimensions. Total six standard and key metrics are used for comparison purpose across both the data sets to add to the relevance and implications for this study.

SHAP (SHapley Additive exPlanations) is employed as a powerful tool for interpreting and explaining the predictions of ML models. SHAP values as expressed in (3) provide a comprehensive understanding of feature contributions to individual predictions, shedding light on the factors that influence the model's output. For this work, SHAP values is used to quantify the impact of each feature on the predicted session duration across both data sets. Such interpretability framework facilitates transparency in model's decision-making process, allowing stakeholders to gain valuable insights into the key drivers behind predicted session durations in both technologies.

$$\phi_i(f) = \frac{1}{N!}\sum_{S\subseteq N\{i\}} \frac{|S|!.(N-|S|-1)!}{N!}\ [f(S\cup\{i\}\ -f(S)] \tag{3}$$

where $N$ is the number of features, $f(S)$ is the model's prediction given the set of features. $S$ represents a coalition of features excluding feature $i$. $|S|$ denotes the cardinality of the set $S$.

Mean Absolute Error (MAE) measures the average absolute difference between the actual and predicted values as expressed in (4). It provides a straightforward indication of the magnitude of errors without considering their direction, with a lower MAE indicating better predictive accuracy.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \tag{4}$$

where $n$ represents the number of observations in the dataset. $y_i$ represents the actual values and $\widehat{y_i}$ represents the predicted values. Mean Squared Error (MSE) measures the average squared difference between the actual and predicted values as shown in (5). Squaring the differences emphasizes larger errors, making it more sensitive to outliers compared to Mean Absolute Error (MAE).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \tag{5}$$

Root Mean Squared Error (RMSE) is like Mean Squared Error (MSE), but the square root is taken to bring the error metric back to the original scale of the dependent variable. It provides a measure of the average magnitude of the errors between actual and predicted values. Lower values of MSE and RMSE signify better model performance.

$$RMSE = \sqrt{MSE} \tag{6}$$

The coefficient of determination, often denoted as $R^2$ and shown in (7) is a metric used to assess the goodness of fit of a regression model. It indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. The $R^2$ value ranges from 0 to 1, where 0 indicates that the model does not explain any variability, and 1 indicates perfect prediction.

$$R^2 = 1 - \frac{(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \overline{y_i})^2} \tag{7}$$

where $n$ is the number of observations in the dataset. $y_i$ represents the actual values, $\widehat{y_i}$ represents the predicted values and $\overline{y_i}$ is the mean of the actual values. Also, Mean Absolute Percentage Error (MAPE) is a commonly used metric to measure the accuracy of a predictive model, especially in time series and regression contexts. MAPE expresses the prediction error as a percentage, which makes it intuitive and easy to interpret.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y_i}}{y_i}\right| \times\ 100 \tag{8}$$

where $n$ represents the number of observations in the dataset, $y_i$ represents the actual value and $\widehat{y_i}$ represents the predicted values. MAPE measures the average absolute percentage error between the actual and predicted values. y expressing the errors as percentages, MAPE allows for a relative comparison of error magnitudes across different datasets or models. It is particularly useful when the scale of the data varies significantly. Lower values of MAPE indicate better model performance. However, MAPE can disproportionately penalize overestimates and underestimates differently, and may not be suitable for datasets with highly variable scales of actual values. Despite this limitation, MAPE

**Table 2**
Model weightage for LTE data.

| Model Name | Ensemble Weights |
|---|---|
| NeuralNet FastAI (Fast AI) | 0.726316 |
| XGBoost (XGB) | 0.242105 |
| CatBoost (CAT) | 0.031579 |

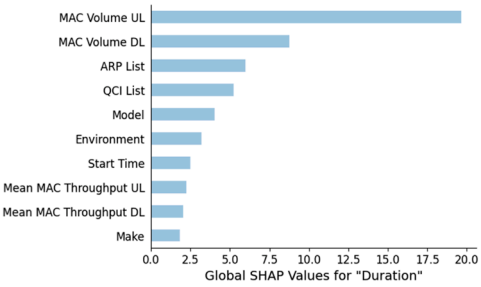



**Fig. 4.** Feature Importance for LTE Data.

remains a valuable metric for providing an easily interpretable measure of prediction accuracy in percentage terms.

Inference latency refers to the time it takes for a system or model to process input data and produce an output (inference). It is a crucial metric in real-time applications where prompt responses are essential. In machine learning, particularly with models deployed for inference tasks, latency is a critical consideration. Its magnitude may differ depending on the specific compute instance employed and the size of the overall dataset.

## 3. Results

### 3.1. LTE data

Based on the approach outlined in Section-I, AutoGluon implemented weighted ensemble approach on LTE and identified that among all 8-base model, only 3 models proved to be effective as shown in Table 2 along with their weights while Fig. 4 shows the feature importance.

Table 3 shows the values of other performance metrics along with their standard deviation. AutoGluon uses all the training data at least once to estimate the performance of a model. Scores are calculated using k-fold cross-validation resampling method that train a machine learning algorithm on different subsets of the dataset. A score is then calculated for overall performance by averaging the resulting performance metrics for each trial.

The actual vs predicted plot shown in Fig. 5 is the difference between actual and predicted model values. The solid red line is a linear line of best fit. If the model were 100 % accurate, each predicted point would equal its corresponding actual point and lie on this line of best fit. The distance away from the line of best fit is a visual indication of model error. The larger the distance away from the line of best fit, the higher the model error.

The standardized residual plot as shown in Fig. 6 measures the strength of the difference between observed and expected values. A

**Table 3**
Performance metrics for LTE data.

| Metric | Value | Standard Deviation |
|---|---|---|
| MAE | 3.6818 | 0.0464 |
| MSE | 64.6589 | 2.4005 |
| RMSE | 8.0410 | 0.1493 |
| $R^2$ | 0.9878 | 0.0058 |
| Inference Latency | 0.160s | – |
| MAPE | 28.97 % | – |

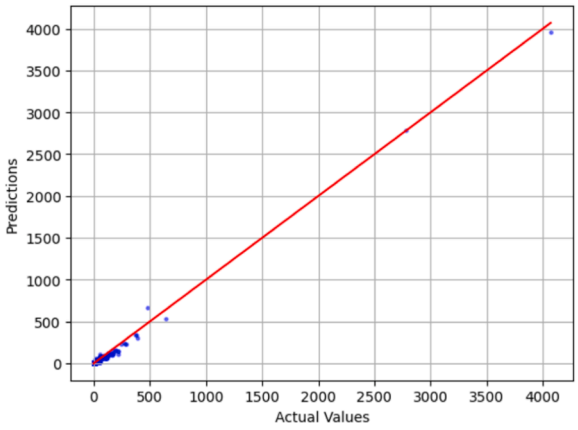



**Fig. 5.** Actual vs Predicted plot for LTE Data.

point shows a value larger than an absolute value of 3 is commonly regarded as an outlier.

The residual histogram of Fig. 7 shows the distribution of standardized residual values. When the histogram is distributed in a bell shape and centered at zero, it indicates that the model does not systematically over or under predict any range of target values. Overall, the weighted ensemble demonstrates strong performance with LTE data.

### 3.2. NR data

Like LTE, NR data also has same AutoGluon configuration of training and testing ratio. Model implementation on NR data identified 3 models (XGB, LR and GBM) proved to be effective as shown in Table 4. In comparison to LTE, XGB is the only common model with weightage while the other two are different.

With NR having 10 additional features when compared to LTE, among total 39 different input features, Fig. 8 displays the top 10 SHAP values, highlighting the attributes with the most significant contributions to the model's predictions. Downlink Volume followed by Start Time and Uplink Volume metrics has the highest SHAP values. In comparison to LTE, high SHAP value for Volume metrics is observed to be standing out.

Table 5 shows the values of other performance metrics of NR along with their standard deviation. Like LTE approach, all the training data is used at least once to estimate the performance of a model to calculate the scores using k-fold cross-validation resampling method. With R2 close to 1 and other metrics at reasonable values, model is performing strong.

The standardized residual plot in Fig. 10 shows most of the predicted

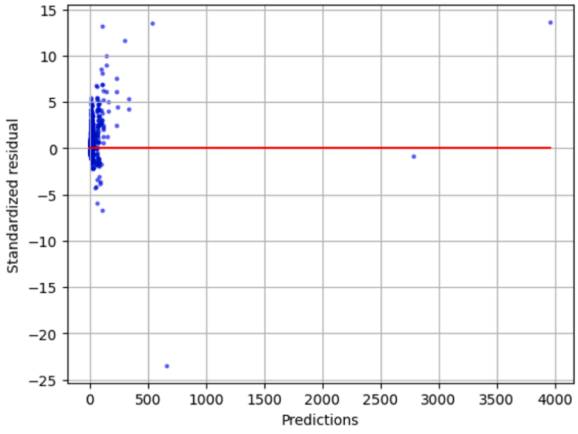



**Fig. 6.** Standardized Residual Plot for LTE Data.

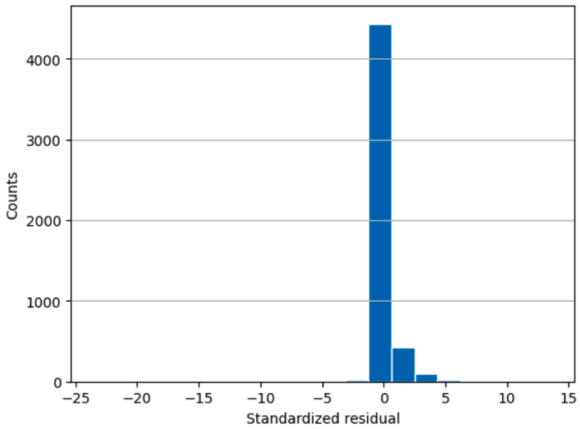



**Fig. 7.** Standardized Residual Histogram for LTE Data.

**Table 4**
Model weightage for NR data.

| Model Name | Ensemble Weights |
|---|---|
| XGBoost (XGB) | 0.963855 |
| Linear Regression (LR) | 0.024096 |
| Light GBM (GBM) | 0.012048 |

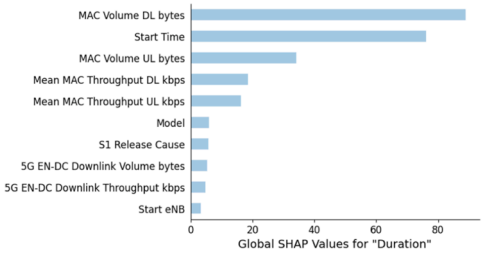



**Fig. 8.** Feature Importance for NR Data.

points to be within +/−3 threshold. The actual vs predicted plot of Fig. 9 shows the difference between actual and predicted model values. In comparison to LTE, both the axes are of high order due to high variation in session durations across both technologies. With most points close to the best fit, model error is minimal. Like LTE, the residual histogram in Fig. 11 shows the distribution of standardized residual values to be centered at zero, indicating that the model does not systematically over or under predict any range of target values.

### 3.3. LTE vs NR

Table 6 is summarization of Tables 3 and 5 to compare evaluation metrics of LTE and NR datasets next to each other. Model performance on LTE is relatively better as all the error metrics of NR are higher than LTE. However, R2 metric is slightly better for NR. NR's higher Inference Latency can partly be attributed to relatively larger data set.

To further explain the performance delta in both technologies, NR

**Table 5**
Performance metrics for NR data.

| Metric | Value | Standard Deviation |
|---|---|---|
| MAE | 8.2879 | 0.1290 |
| MSE | 793.2269 | 193.2641 |
| RMSE | 28.1642 | 3.8137 |
| $R^2$ | 0.9988 | 0.0005 |
| Inference Latency | 0.172s | – |
| MAPE | 12.44 % | – |

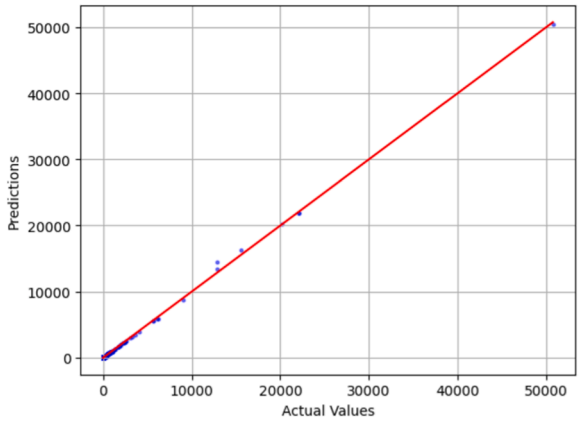



**Fig. 9.** Actual vs Predicted plot for NR Data.

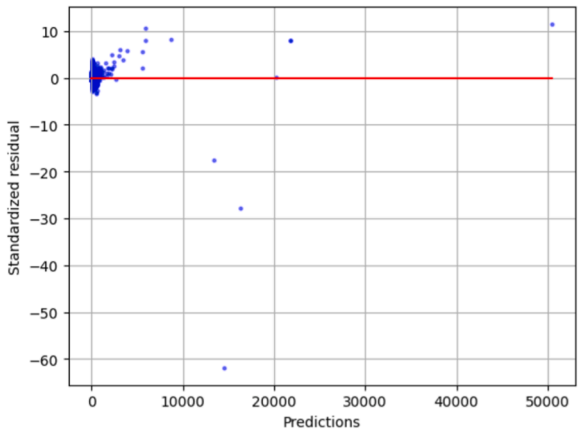



**Fig. 10.** Standardized Residual Plot for NR Data.

data set is further explored, and the large session durations are attributed to a newly introduced device types in NR referred to as Fixed Wireless Access (FWA). These devices are introduced because of excess capacity with introduction of NR with larger bandwidths. Unlike traditional smartphone devices, FWA devices are mostly stationary devices in a residential setup that in provides connectivity to multiple variety of devices like TVs, Laptops, Sensors, which results in relatively longer session durations. While LTE has a lower MAE, the higher MAPE indicates that the relative errors are larger when considering the actual values. This is because LTE has smaller average actual values compared to NR; thus, even small absolute errors can result in large percentage

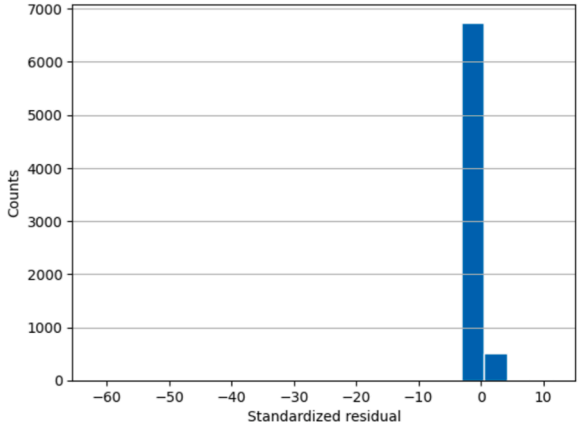



**Fig. 11.** Standardized Residual Histogram for NR Data.

**Table 6**
Model performance of LTE vs NR data.

| Metric | LTE | NR |
|---|---|---|
| Base Models | FastAI, XGB, CAT | XGB, LR, GBM |
| MAE | 3.6818 | 8.2879 |
| MSE | 64.6589 | 793.2269 |
| RMSE | 8.0410 | 28.1642 |
| $R^2$ | 0.9878 | 0.9988 |
| Inference Latency | 0.160s | 0.172s |
| MAPE | 28.97 % | 12.44 % |

errors, inflating MAPE. On the other hand, NR, having larger average actual values, would see smaller percentage errors for the same absolute error, leading to a lower MAPE. Despite having a higher MAE, the MAPE for NR is slightly lower, suggesting that the errors are smaller in percentage terms relative to the larger actual values.

Table 7 helps us to compare the quantitative metrics of these FWA devices to regular smartphones that explains the model performance. FWA devices have a larger session duration and carries more than double the data volumes of smartphones by its nature while being stationary with good radio conditions (RSRP) and no uplink power limitations. While smartphones have different classes of 5 G Quality Indicators (5QI), FWA is always assigned a single 5QI value.

The T-test was employed to rigorously examine the statistical significance of differences in session durations between FWA and regular smartphone users. The obtained T-statistic of −3.1175, with a corresponding p-value of 0.00186, rejects the null hypothesis that there is no significant difference in session durations between the two groups. The negative T-statistic suggests that FWA sessions exhibit longer durations compared to regular smartphone sessions. This statistical analysis provides robust evidence supporting the contention that the observed differences in session durations between FWA and regular smartphone users are statistically significant, emphasizing the distinct characteristics of these user groups within the NR network. Fig. 12

To further assess the model's performance on like-to-like device types in both LTE and NR datasets, only smartphone-specific data is extracted from NR, and the same weighted ensemble model is applied as in the LTE dataset. Table 8 provides a summary of the weights assigned to the base models, showing a more distributed weightage across models compared to previous results. Additionally, six models contribute to the final prediction, a higher number than observed in earlier results. Analyzing the top 10 SHAP attributes, as depicted in Fig. 13, reveals a mix of influential factors. This aligns with LTE's observation, where Uplink volume appears relatively more impactful than downlink volume, while Start time remains a top attribute, consistent with NR data.

The performance metrics for smartphone-only data has been summarized in Table 9 to be consistent with earlier results shown in Tables 3 and 5. This table shows improved performance across all metrics for smartphone-only data compared to NR, while still showing relatively poor performance compared to LTE. This further confirms that the model's accuracy has been affected by the large variations in session durations caused by FWA devices. When excluding FWA data from NR,

**Table 7**
Difference between FWA vs smartphone data.

| Session Metrics | FWA | Smartphone |
|---|---|---|
| Duration Feature | Min: 0.825 s | Min: 0.76 s |
| | Max: 64,088.22 s | Max:9139.07 s |
| | Avg: 82.17s | Avg: 48.34s |
| Device Make Count | 6 | 25 |
| Avg RSRP | −96dBm | −102.01 dBm |
| Mean MAC UL Thpt | 39.61 Mbps | 35.97 Mbps |
| Mean MAC DL Thpt | 14.65 Mbps | 64.79 Mbps |
| Avg MAC UL Volume | 0.46MB | 0.25MB |
| Avg MAC DL Volume | 1.02MB | 0.49MB |
| 5QI | 9 | 6,7,8,9 |

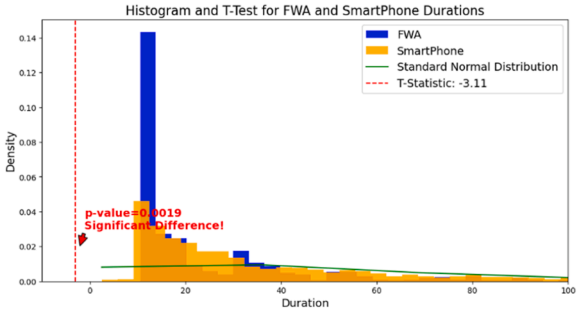



**Fig. 12.** T-Test Results.

**Table 8**
Model weightage for NR data.

| Model Name | Ensemble Weights |
|---|---|
| NeuralNetFastAI | 0.341772 |
| ExtraTreesMSE | 0.303797 |
| XGBoost | 0.227848 |
| LightGBM | 0.063291 |
| CatBoost | 0.050633 |
| LinearModel | 0.012658 |

despite having a higher MAE, the MAPE is still lower than LTE. This indicates that the relative errors in percentage terms are smaller for NR without FWA data, even though the absolute errors are higher. This suggests that NR's larger average session durations result in relatively smaller percentage errors compared to LTE.

### 3.4. Ablation study

The ablation study systematically investigates the impact of individual components within the machine learning model, providing insights into the specific contributions and effectiveness of each element. In this context, it enables a nuanced comparison between the weighted ensemble approach and individual base models like XGBoost and Random Forest, offering a deeper understanding of their unique strengths in predicting session durations.

Table 10 summarizes the best-performing base models for both

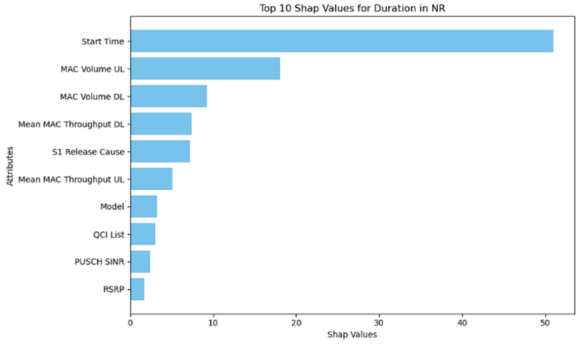



**Fig. 13.** Feature Importance for NR-Smartphone Only Data.

**Table 9**
Performance metrics for NR-smartphone only data.

| Metric | Value | Standard Deviation |
|---|---|---|
| MAE | 9.3362 | 0.1372 |
| MSE | 483.9649 | 84.5460 |
| RMSE | 21.9992 | 2.0526 |
| $R^2$ | 0.99276 | 0.0017 |
| Inference Latency | 0.280s | – |
| MAPE | 19.31 % | – |

**Table 10**
Performance on LTE and NR data without weighted ensemble.

| Metric | LTE | NR |
|---|---|---|
| Model | XGBoost (XGB) | RandomForest (RF) |
| MAE | 3.2139 | 16.3907 |
| MSE | 178.9418 | 262,508.7070 |
| RMSE | 26.8130 | 512.3560 |
| $R^2$ | 0.8644 | 0.6089 |
| Inference Latency | 0.111s | 0.159s |
| MAPE | 25.30 % | 24.59 % |

datasets without the weighted ensemble approach. Comparison of individual models on both LTE and NR datasets reveals higher performance on LTE, consistent with previous results obtained with the weighted ensemble approach. However, contrasting individual model performance with that of the weighted ensemble approach clearly indicates that the weighted ensemble consistently outperforms in all metrics except for inference latency, reflecting the delays introduced by the stacked approach at the expense of improved accuracy. This emphasizes the significance of the weighted ensemble approach in telecom networks, where accuracy is crucial for achieving optimal network performance gains.

## 4. Conclusion and future work

The research presented demonstrates the effectiveness of a weighted ensemble model using AutoGluon in predicting RRC session durations in LTE and NR network environments, emphasizing its potential to optimize resource allocations and improve network performance. The ensemble model notably outperforms individual base models, with LTE exhibiting higher accuracy than NR. The 'Data Volume' metric is identified as a crucial feature in both technologies, underscoring its significance in network management, especially in high-demand scenarios.

The study also highlights the impact of FWA devices, which display a broader range of session durations, on NR data predictions. A T-test confirms significant differences in session durations between smartphones and FWA devices, pointing to the necessity of tailored data analysis and categorization by device type to boost predictive accuracy. Despite achieving superior prediction performance with the ensemble model, the research notes limitations due to the regional and temporal scope of the dataset, suggesting the need for more extensive data to better capture network behavior and user dynamics.

Further, the research lays the groundwork for two pivotal areas of future research: Dynamic Resource Allocation for Mixed Networks and Energy-Efficient Resource Management. These initiatives aim to develop adaptive allocation strategies in heterogeneous network settings and refine energy usage in NR networks, particularly for FWA devices, leveraging predictive insights into session durations. Such advancements could significantly enhance network efficiency and sustainability, meeting the evolving demands of diverse technologies and user needs. This work not only proposes mechanisms for adaptive resource allocation across LTE and NR devices but also explores strategies to optimize energy efficiency in resource management for NR networks, specifically tailored to the unique requirements of FWA devices based on their session duration predictions.

## Authors and contributions

**Roopesh Kumar Polaganga** - As the corresponding author, Roopesh Kumar Polaganga was primarily responsible for the conception and design of the study, data collection, and analysis. He implemented the AutoGluon-based weighted ensemble model, conducted the comparative analysis and ablation study, and drafted the manuscript. Roopesh also handled the manuscript submission process.

**Qilian Liang** - Professor Qilian Liang contributed to the theoretical framework and the interpretation of data. He provided critical revisions that were important for the intellectual content of the manuscript. His guidance was crucial in shaping the research direction and ensuring the rigorous application of machine learning techniques to the telecommunications data.

## CRediT authorship contribution statement

**Roopesh Kumar Polaganga:** Methodology, Software, Writing – original draft, Visualization. **Qilian Liang:** Supervision, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The data that has been used is confidential.

*Acknowledgments*

This work was supported in part by the U.S. National Science Foundation under grant CCF-2219753.

*Data Availability Statement*

The raw data used in this study is not publicly available to ensure confidentiality and cannot be shared for public use.

## References

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.
Brezov, Danail & Burov, Angel & Brezov, Danail & Burov, Angel. (2023). Ensemble Learning Traffic Model for Sofia: A Case Study. 10.3390/app13084678.
Caruana, R., Niculescu-Mizil, A., Crew, G., & Ksikes, A. (2004). Ensemble selection from libraries of models. In , *18. Proceedings of the 21st International Conference on Machine Learning*.
Dietterich, T. G. (2000). Ensemble methods in machine learning. *International workshop on multiple classifier systems* (pp. 1–15). Springer.
Dorogush, Anna & Ershov, Vasily & Gulin, Andrey. (2018). CatBoost: Gradient boosting with categorical features support.
Erickson, Nick & Mueller, Jonas & Shirkov, Alexander & Zhang, Hang & Larroy, Pedro & Li, Mu et al.. (2020). AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data.
Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In , *96. Proceedings of the 13th International Conference on Machine Learning* (pp. 148–156). Citeseer.
Guo, C. Berkhahn, F. (2016). Entity Embeddings of Categorical Variables.
How, K., Pang, Y., Ooi, S. Y., Wang, L.-Y.-K., & Poh, Q. (2023). Predictive churn modeling for sustainable business in the telecommunication industry: Optimized weighted ensemble machine learning. *Sustainability, 15*, 8631. https://doi.org/10.3390/su15118631
Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W. et al. LightGBM: A highly efficient gradient boosting decision tree. In Advances in neural information processing systems, pp. 3146–3154, 2017.
Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W. et al. LightGBM: A highly efficient gradient boosting decision tree. In Advances in neural information processing systems, pp. 3146–3154, 2017.
Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2017). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in weka. *Journal of Machine Learning Research, 18*(25), 1–5.
Liang, F., Shen, C., & Yu, W. (2019). Towards optimal power control via ensembling deep neural networks. In *IEEE Transactions on Communications*. https://doi.org/10.1109/TCOMM.2019.2957482, 1-1.
Luo, C., Zeng, J., Yuan, M., Dai, W., & Yang, Q. (2016). Telco user activity level prediction with massive mobile broadband data. *ACM Transactions on Intelligent Systems and Technology, 7*, 1–30. https://doi.org/10.1145/2856057
Mendoza, H., Klein, A., Feurer, M., Springenberg, J. T., & Hutter, F. (2016). Towards automatically-tuned neural networks. *Workshop on automatic machine learning* (pp. 58–65).
Mohr, F., Wever, M., & Hüllermeier, E. (2018). ML-plan: Automated machine learning via hierarchical planning. *Machine Learning, 107*(8), 1495–1515.

Parmanto, B., Munro, P. W., & Doyle, H. R. (1996). Reducing Variance of committee prediction with resampling techniques. *Connection Science, 893-4*, 405–425.
Rekkas, V., Sotiroudis, S., Sarigiannidis, P., Wan, S., Karagiannidis, G., & Goudos, S. (2021). Machine learning in beyond 5G/6G networks—State-of-the-art and future trends. *Electronics, 10*, 2786. https://doi.org/10.3390/electronics10222786
Sagi, O., & Rokach, L. (2021). Approximating XGBoost with an interpretable decision tree. *Information Sciences, 572*. https://doi.org/10.1016/j.ins.2021.05.055
Song, Yimeng, Xu, Yong, Chen, Bin, He, Qingqing, Tu, Ying, Wang, Fei, et al. (2022). *Dynamic population mapping with AutoGluon, 1* p. 13). https://doi.org/10.1007/s44212-022-00017-x
Upadhyay, Deepak & Tiwari, Pallavi & Mohd, Noor & Pant, Bhaskar. (2022). A Machine Learning Approach in 5G User Prediction. 10.1007/978-981-19-3571-8_59.
Van der Laan, M.J., Polley, E.C., and Hubbard, A.E. Super learner. Statistical applications in genetics and molecular biology, 6(1), 2007.
Vinutha, H.P. & Poornima, B. & Sagar, B. (2018). Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. 10.1007/978-981-10-7563-6_53.
Wilhelmi, F., Carrascosa, M., Cano, C., Jonsson, A., Ov, V., & Bellalta, B. (2021). Usage of network simulators in machine-learning-assisted 5G/6G networks. *IEEE Wireless Communications, 28*, 160–166. https://doi.org/10.1109/MWC.001.2000206
Wong, L., & Michaels, A. (2022). Transfer learning for radio frequency machine learning: A taxonomy and survey. *Sensors, 22*, 1416. https://doi.org/10.3390/s22041416

**ROOPESH KUMAR POLAGANGA** received his B.Tech degree in Electronics and Communication Engineering (ECE) from Pondicherry Engineering College, Pondicherry, India, in 2013, the M.S. (Thesis) degree in Electrical Engineering (EE) from the University of Texas at Arlington, Texas, USA, in 2015 and the Master of Business Administration (MBA) degree from Capella University, Minneapolis, USA in 2019. While at UTA, he served as Graduate Research Assistant in the Communication and Networking Lab under the guidance of Dr. Liang where his research was focused on Ultra-Wide Band and LTE technologies.

He is currently working as Principal Systems Architect Engineer at T-Mobile US since 2015 where he successfully designed several features and solutions in 5G-NR, LTE/LTE-Advanced and IoT for everyday customer use. Besides technology development, he also contributed towards multiple M&A projects to realize network synergies and improved overall customer experience. He has authored several journal papers and filed more than 80 U.S. patent applications. He has also received numerous corporate awards at T-Mobile US, including peak nominations, the company's highest honor. His technical areas of interests include Wireless Telecommunications, Cloud Networks, Internet of Things, AI/ML in Telco networks.

**QILIAN LIANG** received the B.S. degree in electrical engineering from Wuhan University, Wuhan, China, in 1993, the M. S. degree in electrical engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 1996, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2000.

He was a member of Technical Staff with Hughes Network Systems Inc., San Diego, CA, USA. He is a Distinguished University Professor with the Department of Electrical Engineering, The University of Texas at Arlington (UTA), Arlington, TX, USA. He has authored or coauthored over 350 journals and conference papers, seven book chapters, and has six U.S. patents pending. His current research interests include machine learning, wireless sensor networks, wireless communications, smart grids, signal processing for communications, and fuzzy logic systems and applications. Dr. Liang was a recipient of the 2002 IEEE Transactions on Fuzzy Systems outstanding Paper Award, the 2003 U.S. Office of Naval Research Young Investigator Award, the 2005 UTA College of Engineering Outstanding Young Faculty Award, the 2007, 2009, and 2010 U.S. Air Force Summer Faculty Fellowship Program Award, the 2012 UTA College of Engineering Excellence in Research Award, and the 2013 UTA Outstanding Research Achievement Award. He was inducted into the UTA Academy of Distinguished Scholars, in 2015. He is a Fellow of the IEEE, AIIA, and AAIA.