Enhancing network resource management through machine learning for spatio-temporal beam-level traffic forecasting

Stephen Kolesh[1], Thomas Basikolo[2]

[1]Sportserve, Nairobi, Kenya, [2]International Telecommunication Union, Telecommunication Standardization Bureau, Geneva, Switzerland

Corresponding author: Stephen Kolesh, Koleshjr@gmail.com

Accurate forecasting of Downlink Throughput Volume (DLThpVol) at the beam level is essential for improving resource management in modern communication networks. This study addresses the challenges posed by complex, high-dimensional spatio-temporal traffic data, leveraging multivariate time series that include critical factors such as Physical Resource Block (PRB) utilization and user count. Recent benchmarks on traditional and deep learning models (e.g., iTransformer, PatchTST, DLinear) achieve Mean Absolute Errors (MAEs) ranging from 0.1967 to 0.2005 on short-term targets and up to 0.2352 on longer-term forecasts, but opportunities remain for improvement through domain-informed feature engineering.

We propose a dual-pipeline Gradient Boosting Decision Tree (GBDT)-based framework for beam-level DLThpVol prediction that incorporates carefully engineered temporal and spatial features (e.g., PRB utilization dynamics, beam-level user clustering). Our models achieve MAEs of 0.1919 (short-term) and 0.2261 (long-term), outperforming several deep learning benchmarks by up to 11.4% on short-term forecasts. These results demonstrate that interpretable, feature-driven ensemble learners can provide competitive forecasting performance while maintaining computational efficiency.

Although the work does not directly implement congestion-aware resource allocation, the improved forecast accuracy lays the foundation for future studies on predictive resource management, such as PRB provisioning and energy-efficient beam scheduling. Our findings highlight the importance of combining domain knowledge with interpretable machine learning for advancing spatio-temporal traffic forecasting in communication networks.

Keywords – Beam-level, downlink throughput volume, ensemble learning, feature engineering spatio-temporal forecasting

1. INTRODUCTION

The exponential growth of mobile data traffic, driven by the proliferation of smart devices and bandwidth-intensive applications, has placed unprecedented strain on communication network management, particularly in the areas of traffic forecasting and dynamic resource allocation [1, 2]. Accurate spatio-temporal forecasting of network traffic is essential to ensure Quality of Service (QoS), minimize congestion, and optimize the use of network resources [3]. While traditional cell-level traffic prediction methods have provided foundational insights [4], the emergence of beamforming and massive Multiple-Input Multiple-Output (MIMO) technologies which enable the use of multiple directed beams per cell have facilitated fine-grained traffic management at the beam level, where multiple directional beams are deployed per base station cell to serve users more efficiently [5, 6]. These technologies demand granular forecasting at the beam level to optimize performance and energy efficiency. As a result, the ability to accurately predict traffic at the beam level has become increasingly important, especially in the context of ultra-dense 5th Generation (5G) networks and beyond [7]. This paradigm shift introduces new complexities, as beam-level traffic exhibits finer spatial variability and stronger interdependencies between neighboring beams, necessitating advanced modeling approaches [8].

Existing traffic forecasting methodologies, such as Autoregressive Integrated Moving Average (ARIMA) [9] and Holt-Winters exponential smoothing [10], have demonstrated utility in low-dimensional, stationary time series contexts. However, these models often struggle with high-dimensional, nonlinear, and noisy data, which are common in modern communication networks. Consequently, Machine Learning (ML) techniques have gained attention for their ability to model complex, nonlinear dependencies in spatio-temporal data [11, 12].

Recent advances in machine learning offer promising alternatives, particularly GBDT [13] architectures such as LightGBM [14] and CatBoost [15]. These ML models excel at capturing complex feature interactions and temporal dependencies while maintaining computational efficiency which is a critical advantage for real-time network operations [16]. These ML models have demonstrated strong performance on tabular data and multivariate time series tasks. GBDT models offer robust feature handling, interpretability, and scalability, making them well-suited for forecasting tasks involving structured network telemetry.

Accurate forecasting of Downlink Throughput Volume (DLThpVol) at the beam level is essential for optimizing resource allocation and enhancing user experience in modern wireless networks. Despite significant progress in spatio-temporal traffic prediction, most existing approaches focus on cell or sector-level forecasting, neglecting the fine-grained spatial resolution enabled by beamforming [8]. Furthermore, other methods underutilize multivariate operational metrics such as Physical Resource Block (PRB) utilization that are strongly correlated with throughput dynamics. Additionally, few studies explicitly link improved forecasting accuracy to tangible network outcomes, such as energy savings or congestion reduction.

Benchmarking efforts by [17, 18] have established strong performance baselines using both traditional and deep learning models. For example, models such as iTransformer [19], PatchTST [20], and DLinear [21] achieved MAEs ranging from 0.1967 to 0.2005 on short-term targets, and up to 0.2352 on longer-term forecasts (see Table 1). While these results represent significant progress in the field, the complex and high-dimensional nature of spatio-temporal traffic data leaves room for additional accuracy improvements particularly through domain-specific feature engineering and interpretable model architectures.

While these deep learning models benefit from automated feature extraction, this often comes at the cost of interpretability and significant computational overhead. In contrast, a structured feature engineering approach, as pursued in this work, allows for the explicit encoding of known domain knowledge (e.g., diurnal and weekly cycles) into the model and studying the feature importance later after training. This strategy aims to create a framework that is not only more accurate but also computationally efficient and transparent, which are critical requirements for operational deployment in real-world network management systems.

Table 1 – Comparative MAE scores for short and long-term traffic forecasting by [17]

Target	Hist.Avg.	iTransformer	PatchTST	DLinear	Transformer
Week 6 Short Term	0.2108	0.1967	0.1973	0.2005	0.2166
Week 11 Long Term	0.2431	0.2348	0.2343	0.2352	0.2331

To address these gaps, we challenge the trend of increasing model complexity by proposing a GBDT-based framework that demonstrates the superior performance of domain-specific feature engineering combined with computationally-efficient models. Our approach, leveraging LightGBM and CatBoost, is designed not just for performance but for practical deployment, prioritizing efficiency and transparency over the 'black-box' nature of more complex alternatives. Our approach integrates structured feature engineering with lightweight, interpretable model design, achieving superior performance compared to both traditional machine learning models and state-of-the-art deep learning benchmarks. Specifically, our framework attains MAEs as low as 0.1919 for short-term predictions (Week 6) one week after the training period and 0.2261 for long-term forecasts (Week 11) 6 weeks after the training period, outperforming existing baselines. By explicitly modeling intra-beam temporal patterns and inter-beam spatial correlations using high-resolution hourly traffic data across multiple base stations, our model captures the complex drivers of traffic variability with extraordinary granularity.

The contributions of this work are two-fold:

- We demonstrate that a GBDT-based forecasting framework, when combined with deliberate, domain-specific feature engineering, achieves consistent performance gains over state-of-the-art deep learning baselines. Our work provides empirical evidence that prioritizing an interpretable and computationally-efficient design can yield superior results compared to more complex, automated architectures in this domain**.** Our LightGBM, CatBoost, and ensemble models outperform leading automated architectures such as iTransformer, PatchTST, DLinear, and Transformer by up to 4.21 percentage points for short-term forecasting (Week 6) and 5.97 percentage points for long-term forecasting (Week 11), on average across all evaluated horizons.
- We introduce a dual-pipeline forecasting design tailored to short-term and long-term horizons. While the first contribution focuses on model performance and interpretability, this second contribution emphasizes temporal adaptability offering a flexible blueprint for optimizing forecast accuracy across varying time scales. This design paradigm generalizes well to other spatio-temporal prediction domains, including network traffic, energy load, and mobility forecasting.

The remainder of this paper is structured as follows: Section 2 reviews related work; Section 3 details the proposed methodology; Section 4 presents the data and feature engineering; Section 5 reports the experimental results; and Section 6 provides the conclusion.

2. LITERATURE REVIEW

Beam-level traffic forecasting has become increasingly critical with the emergence of 5G and next generation networks, where highly directional beams enable unprecedented spatial control over signal distribution [22]. Unlike traditional cell-level prediction, beam-level forecasting captures usage patterns at a much finer granularity, facilitating intelligent resource allocation, energy optimization, and congestion mitigation [23], [24]. Accurate prediction at the beam level is particularly vital for ultra-dense network deployments, where user mobility and interference management present significant operational challenges [25]. Despite this potential, much of the prior research on traffic forecasting has focused on broader spatial resolutions such as cells or base stations thereby lacking the specificity required for modern, beam-centric architectures [26].

Time Series Forecasting (TSF) plays a foundational methodology for network traffic prediction [27]. Classical statistical approaches, including ARIMA [9] and Holt-Winters exponential smoothing [10], have traditionally been used to model time-dependent phenomena. While these models are effective at capturing seasonality and linear trends, they are inherently limited in high-dimensional, nonstationary environments. Their univariate nature further restricts their ability to account for complex interactions among multiple correlated variables, a critical requirement for cellular network traffic analysis [9].

To address these limitations, Multivariate Time Series Forecasting (MTSF) has emerged as a more expressive paradigm, enabling the modeling of multiple interrelated time-dependent variables [27]. MTSF techniques allow for simultaneous analysis of various traffic features, such as Physical Resource Block (PRB) utilization, user count, and throughput volume. Early work in this space employed multivariate extensions of ARIMA (VARIMA) or state-space models [28], but these techniques often fail to scale or generalize effectively to nonlinear and high-dimensional domains. In response, machine learning models particularly Gradient Boosting Decision Trees (GBDT) such as LightGBM [14] and CatBoost [15] have shown strong predictive performance when paired with extensive feature engineering [29], [30]. Features such as lag values, rolling means, expanding statistics, and temporal encodings provide the model with a rich representation of past behaviors, allowing it to learn complex nonlinear dependencies in multivariate settings [30].

In recent literature, spatio-temporal forecasting models have increasingly incorporated beam-level data to improve prediction granularity and accuracy [31]. Deep learning methods, including Recurrent Neural Networks (RNNs) [32], Long Short-Term Memory (LSTM) networks [33], and more recently, transformer-based architectures [34], have been widely adopted for modeling time-dependent sequences with spatial embeddings. While these models achieve state-of-the-art accuracy, they are often resource-intensive and difficult to interpret. Conversely, studies have demonstrated that GBDT models, though less complex, can rival deep models when equipped with well-designed spatio-temporal features especially in contexts where real-time inference, computational efficiency, and model interpretability are essential [30], [35]. However, relatively little work has focused specifically on beam-level traffic forecasting using GBDTs, leaving a gap in understanding their comparative effectiveness at this fine spatial resolution.

This study aims to address this gap by proposing a dual-pipeline approach for beam-level traffic forecasting using GBDT models. Specifically, we employ two feature selection strategies tailored to short-term (Week 6) and long-term (Week 11) prediction horizons, respectively. The use of *stratified K-fold cross-validation* across base stations ensures robust model evaluation, while advanced feature engineering including target encoding, lag features, rolling statistics, and weekly aggregations enhances predictive fidelity. Unlike existing work that either relies heavily on deep learning [29], [36] or operates at coarser spatial resolutions [16], our approach demonstrates that GBDTs can provide competitive performance in beam-level forecasting when supported by carefully curated features.

In summary, this study introduces a novel dual-pipeline GBDT-based framework specifically tailored for beam-level traffic forecasting, thereby addressing several significant gaps in the current literature. While prior work has largely focused on deep learning-based spatio-temporal models or limited itself to coarser spatial resolutions, our approach bridges the methodological divide by demonstrating that efficient, interpretable tree-based learners when coupled with domain-specific feature engineering can achieve competitive performance even at the fine-grained beam level. The proposed framework is informed by the need for lightweight, deployable models that maintain high accuracy and interpretability in real-world network environments, an area that has received limited attention despite its practical importance. By systematically evaluating GBDT models with advanced feature selection and cross-validation strategies, this work not only fills the research gap concerning interpretable forecasting at beam granularity but also establishes a foundation for future exploration of hybrid or transformer-based enhancements in network traffic prediction. Thus, the present study advances both methodological innovation and practical applicability, offering actionable insights for network operators and guiding future research toward more efficient, sustainable, and intelligent communication systems.

3. METHODOLOGY

3.1 Problem formulation

We aim to forecast Downlink Throughput Volume (DLThpVol) for each base station at hourly intervals. Let:

$$\mathcal{D} = \{(\mathbf{X}_t, y_t)\}_{t=1}^{N}$$

denote the dataset, where:

$$\mathbf{X}_t \in \mathbb{R}^d$$

is the vector of observed network features at time ttt (e.g., PRB utilization, active user counts, categorical beam IDs) and

$$y_t \in \mathbb{R}$$

is the corresponding DLThpVol.

To capture temporal dependencies, we augment $\mathbf{X}_t$ with derived features to obtain:

$$\mathbf{x}_t \in \mathbb{R}^p$$

(where ***p*>*d***), including lagged values, rolling statistics, and periodic encodings. The forecasting problem then reduces to learning a mapping function:

$$f : \mathbb{R}^p \to \mathbb{R}$$

that predicts future DLThpVol:

$$y_t = f(\mathbf{x}_t) \qquad (1)$$

The model is trained to minimize the empirical loss over all observations. We adopt the Mean Absolute Error (MAE) as the loss function:

$$L(f) = \frac{1}{N}\sum_{t=1}^{N} |y_t - f(\mathbf{x}_t)| \qquad (2)$$

3.2 Gradient boosting framework

We model ***f*** as an additive ensemble of ***M*** regression trees:

$$y_t = \sum_{m=1}^{M} h_m(\mathbf{x}_t) \qquad (3)$$

Here, $(h_m(\mathbf{x}_t))$ predicts an adjustment to the previous ensemble for DLThpVol at time ***t***, and ***N*** denotes the total number of hourly observations across all base stations.

At each boosting iteration, a new tree is trained on the pseudo-residuals, i.e., the negative gradient of the loss function:

$$r_t^{(m)} = -\frac{\partial L(y_t, \hat{y}_t^{(m-1)})}{\partial \hat{y}_t^{(m-1)}} \qquad (4)$$

The new tree approximates these residuals:

$$h_m(\mathbf{x}_t) \approx r_t^{(m)} \qquad (5)$$

This procedure ensures that each subsequent tree focuses on the prediction errors of the previous ensemble.

3.3 Temporal feature engineering

To capture sequential dependencies in DLThpVol, we construct several temporal features:

- **Lag features** – capture immediate past network load:

$$x_t^{\text{lag-}k} = y_{t-k} \qquad (6)$$

- **Rolling mean** – averages throughput over the previous www hours:

$$x_t^{\text{roll-mean}} = \frac{1}{w}\sum_{i=1}^{w} y_{t-i} \qquad (7)$$

- **Expanding mean** – accumulates historical trends:

$$x_t^{\text{exp-mean}} = \frac{1}{t-1}\sum_{i=1}^{t-1} y_i \qquad (8)$$

- The target encoding (with caution) of a categorical feature such as beam ID is defined as:

$$TE_{\text{beam}} = \mathbb{E}[y_t \mid \text{beam ID}] \qquad (9)$$

Target encoding is computed in a time-aware manner to avoid leakage, e.g., using out-of-fold schemes or expanding means over past observations.

These features allow the model to capture both short-term fluctuations and long-term trends in network throughput.

3.4 Dual-pipeline strategy for temporal forecasting

A single, monolithic forecasting model often struggles to optimize for both short-term and long-term prediction horizons simultaneously. Short-term forecasting is highly dependent on recency and autoregressive features. Long-term forecasting is dominated by stable, long-term periodicities. By employing a dual-pipeline, we can create a specialized feature set and model for each task. This architectural choice prevents a suboptimal trade-off and allows each model to excel at its specific horizon.

- **Short-term forecasting (Week 6)**: ~1 week after 5 weeks of training data. Uses fine-grained temporal features such as lag values, rolling statistics, and short-term target encodings to capture hourly and daily fluctuations.
- **Long-term forecasting (Week 11)**: ~11th week after a 5 week gap. Uses a reduced, temporally stable feature set, including aggregated trends and static variables (e.g., PRB utilization trends, user counts, categorical encodings). Highly time-sensitive variables are excluded to improve generalization.

Week 6 and Week 11 were chosen based on the challenge objective.

3.5 Final objective and regularization

The overall training objective combines the MAE loss with a regularization term to control model complexity:

$$L_{\text{total}} = \sum_{t=1}^{N} |y_t - f(\mathbf{x}_t)| + \Omega(f)$$

- In LightGBM, $\Omega(f)$ penalizes large leaf weights and tree depth.
- In CatBoost, ordered boosting, depth limits, and learning rate adjustments achieve similar regularization.

This ensures strong predictive performance while avoiding overfitting on noisy or high-dimensional network data.

4. DATA AND FEATURE ENGINEERING

4.1 Dataset description and forecasting problem

This study leverages a *high-resolution multivariate time series dataset* collected from an operational cellular network over five consecutive weeks. Captured at hourly intervals, this dataset provides granular insights into network dynamics across *2 880 unique directional beams* distributed over *30 base stations*, with each station comprising 3 cells and 32 beams (30 stations × 3 cells/station × 32 beams/cell = 2 880 beams). This hierarchical structure enables detailed analysis of spatio-temporal traffic patterns at unprecedented resolution [17].

The primary objective is to forecast the *DLThpVol*, which represents the total volume of data transmitted to users within a beam's coverage area during a one-hour period. This target variable is the central indicator of user-perceived network performance and resource consumption. To support a robust multivariate forecasting approach, the target variable is contextualized with several key exogenous variables, provided in separate but time-aligned files.

- **Downlink Throughput Time (*DLThpTime*)**: This variable measures the duration within each hour that the downlink channel was active. It provides insight into the temporal consistency of data transmission.
- **PRB utilization** This metric quantifies the percentage of available frequency-time resource blocks that were allocated for data transmission. PRB utilization is a critical indicator of network load and resource contention, serving as a primary explanatory variable for throughput.
- **User count (MR_number)** This variable records the number of unique user devices served by a beam in a given hour. It directly reflects the spatial distribution of demand and is a key driver of traffic volume.

Together, these variables form a rich multivariate framework for modeling beam-level traffic dynamics. The forecasting challenge is formally defined as the task of predicting the hourly *DLThpVol* for each of the 2 880 individual beams. The prediction problem is structured around two distinct temporal horizons, designed to evaluate different aspects of model generalization. The *short-term forecasting* which predicts *DLThpVol* values for Week 6, immediately following the training period to assess the model's ability to capture recent temporal patterns and near-term trends and the *long-term forecasting* which predicts *DLThpVol* values for Week 11, five weeks beyond the end of the training window. This scenario is intended to evaluate the model's capacity to generalize over extended time intervals and to learn stable, long-range dependencies that persist beyond immediate historical contexts.

These dual forecasting objectives introduce varying levels of complexity, particularly with respect to temporal drift and feature relevance, making the task well-suited for assessing robustness in time-series modeling under real-world constraints.

4.2 Exploratory data analysis and methodological implications

A comprehensive Exploratory Data Analysis (EDA) was conducted to uncover the underlying structure, patterns, and statistical properties of the dataset. The findings from this analysis were instrumental in justifying the subsequent feature engineering and modeling choices.

(a) Temporal patterns and periodicity: Fig. 1 presents the mean hourly DLThpVol across three representative base stations over a full weekly cycle. The visualizations reveal pronounced and recurring temporal patterns, confirming the presence of strong periodic behavior in cellular traffic. Two dominant cycles are clearly identifiable.

(b) Diurnal cycle (24-hour): Each base station exhibits a regular daily rhythm. Traffic volumes consistently rise to a peak during the afternoon and evening hours and fall to a distinct trough in the early morning, typically between 3 AM and 5 AM. This bimodal or unimodal daily pattern directly mirrors daily human activity.

(c) Weekly cycle (168-hour): The trendlines also demonstrate a clear weekly periodicity, where weekday traffic patterns (Monday to Friday) differ significantly from weekend patterns (Saturday and Sunday). During weekdays, the daily peaks are sharp and highly regular. In contrast, the weekend shows a more varied and less predictable pattern. For example, the sharp drop-off in traffic on Friday night is followed by a more erratic and sustained, lower-level usage on Saturday and a slightly different recovery pattern on Sunday.

Additionally, cross-station comparisons highlight significant spatial heterogeneity in traffic demand.

- Base station 2 consistently handles the highest traffic volume, showing the most pronounced peaks and troughs throughout the week.

- Base station 1 shows a moderate traffic load, generally following the same temporal pattern but at a lower amplitude than Base station 2.
- Base station 0 consistently experiences the lowest traffic volume of the three, though it still follows the same fundamental diurnal and weekly cycles.

The distinct and consistent ordering of traffic volumes (Base station 2 > Base station 1 > Base station 0) underscores the necessity of treating each base station individually in the modeling process.

These temporal and spatial insights validate the importance of explicitly encoding time-based features (such as hour of the day and day of the week) and station-specific identifiers. Capturing these features allows a predictive model to learn the cyclical structure of network demand for each location, which is crucial for improving forecasting accuracy.

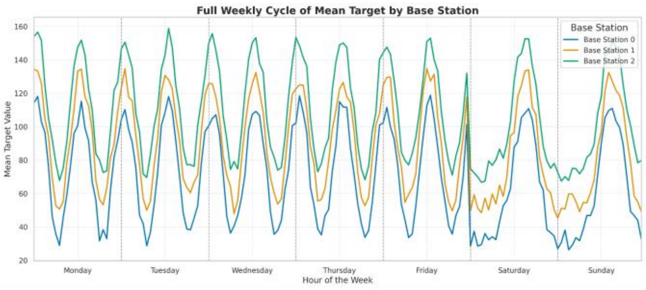

Figure 1 – Full weekly cycle of DLThpVol for base stations 0,1 and 2

(d) Spatial heterogeneity: Fig. 2 presents a box plot of *DLThpVol* by base station, revealing substantial variation in both the average throughput volume and its variability across locations. This variation exemplifies ***spatial heterogeneity***, which refers to differences in a variable across geographic space. The observed disparities are likely influenced by contextual factors such as whether a base station serves a residential or commercial area, differences in user density, or variations in physical environment and infrastructure.

Furthermore, a heatmap as shown in Fig. 3, depicting traffic volume across beams, within a single base station illustrates that certain beams consistently handle disproportionately high or low shares of traffic. These spatial imbalances underscore the limitations of adopting a uniform, "one-size-fits-all" modeling approach, which fails to account for localized usage patterns.

As a result, the findings motivated the development of group-based feature engineering strategies, enabling the model to learn distinct behavioral patterns for each base station and beam. This approach enhances the model's ability to capture spatial variability and improves predictive accuracy in heterogeneous network environments.

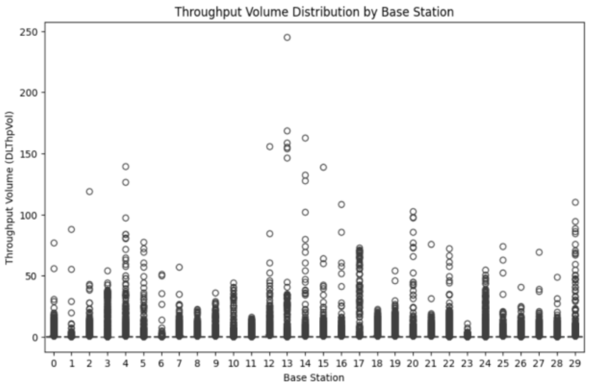

Figure 2 – Throughput volume distribution by base station

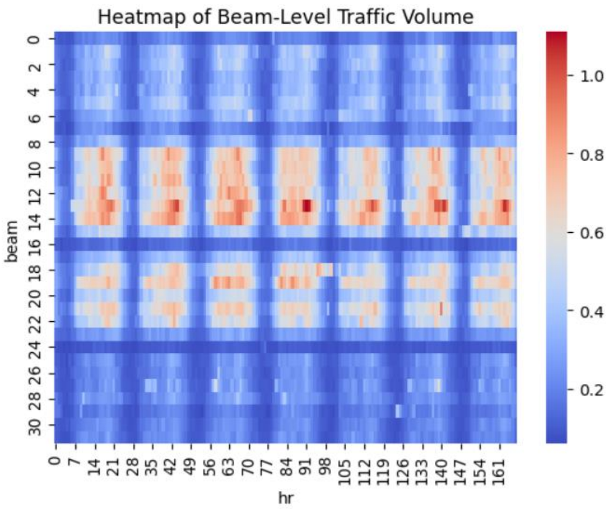

Figure 3 – Heatmap of beam-level traffic volume

(e) Target variable distribution: The distribution of the target variable *DLThpVol* was found to be strongly right-skewed, characterized by a large concentration of observations with low or zero traffic volume and a long tail representing rare but high-volume events, as shown in Fig. 4. Common transformations, such as the square root transformation, were applied to reduce skewness, but only provided partial normalization, as shown in Fig. 5.

This distributional characteristic has two important implications for model development and evaluation. Firstly, it indicates that evaluation metrics such as Mean Squared Error (MSE), which are highly sensitive to outliers, would disproportionately penalize errors on infrequent high-volume instances. As a result, MAE was selected as the primary evaluation metric due to its greater robustness to skewed distributions. Secondly, the non-Gaussian nature of the data underscores the need for models capable of handling complex, non-linear relationships and irregular distributions, properties for which tree-based models like LightGBM are particularly well-suited.

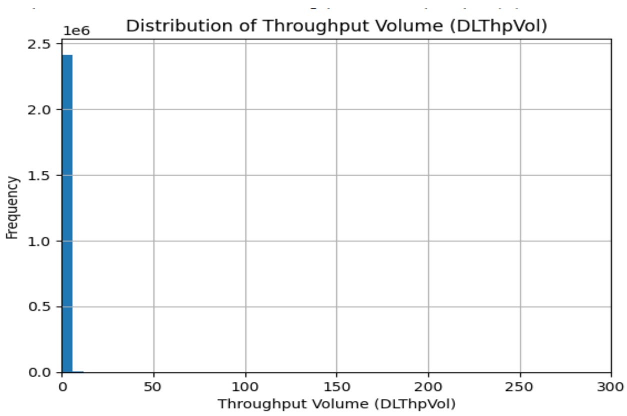

Figure 4 – Distribution of throughput volume

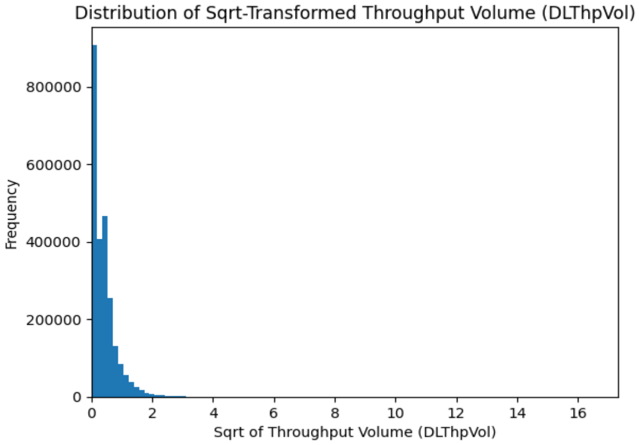

Figure 5 – Sqrt distribution of throughput volume

4.3 Feature engineering strategy

Guided by insights from the EDA and building on the formal definitions introduced in Section 3 (equations 6–9), a comprehensive feature engineering pipeline was designed to transform the raw time-series data into a rich, tabular feature set. This process was essential for enabling the GBDT models to learn the complex spatio-temporal dependencies in the network data.

The engineered features were grouped into distinct categories, each encoding different types of information relevant to the forecasting task.

(a) Lag and rolling window features: As formulated in equations (6) and (7), lag and rolling window features were constructed to capture short-term temporal dependencies and smoothed local trends in DLThpVol.

- **Lag features**: Lagged values of the target variable, PRB utilization, and active user counts from the previous 1 to 4 hours were created to provide the model with immediate historical context and momentum in network load.
- **Rolling window statistics**: To provide a more stable view of recent history, statistical aggregations were computed over moving windows of 168 hours (1 week) and 336 hours (2 weeks) to align with the observed weekly seasonality.

To prevent data leakage, these windows were shifted by one week, ensuring that each feature at time $\boldsymbol{t}$ was computed strictly using data from times – $\boldsymbol{t-168}$ and earlier. Within each window, the mean, median, standard deviation, and 25th/75th percentiles were computed to capture both central tendency and variability.

(b) Expanding window features: Following Equation (8), expanding window features were designed to capture long-term, cumulative trends in the data. Unlike rolling features that emphasize short-term patterns, these compute statistics using all historical observations available up to time ttt, effectively encoding the cumulative “memory” of network behavior. Specifically, expanding mean and standard deviation were calculated for both the target variable and key exogenous predictors, embedding information about their evolving central tendency and variability.

This approach allowed the model to account for gradual changes and persistent effects across the entire observation period.

(c) Group-based aggregations and fold-aware target encoding: As formalized in Equation (9), target encoding [37] was used to represent categorical variables (e.g., beam ID, base station, cell type) through aggregated statistics of the target variable. This step was crucial to capturing spatial heterogeneity and entity-specific behavioral patterns identified during EDA.

However, a naive implementation where the encoding for a row is computed using its own target value leads to data leakage and overfitting. To avoid this, a robust fold-aware target encoding scheme was implemented.

The procedure was as follows:

1. The training data was divided into $\boldsymbol{K}$ folds, stratified by a key categorical feature (e.g., base_station).
2. For each fold $\boldsymbol{i}$, category-level statistics (mean, standard deviation, skewness, min, max, percentiles) were computed using only the remaining $\boldsymbol{k-1}$ folds.
3. These out-of-fold statistics were then used to encode the categorical features within fold $\boldsymbol{i}$.
4. The process was repeated until all folds were encoded, ensuring that each encoded feature was leak-free.
5. Finally, the statistics computed from the full training data was used to encode the unseen test set.

This method was applied across multiple grouping hierarchies such as ['base_station', 'beam'], ['base_station', 'daily_hr'], and ['base_station', 'cell_type', 'beam'], allowing the model to capture nuanced spatial-temporal interactions.

By systematically applying this fold-aware target encoding, the model gained a richer and more generalizable representation of contextual dynamics, a critical factor in achieving high forecasting accuracy.

5. EXPERIMENTS

This section presents a comprehensive experimental framework designed to evaluate the proposed forecasting methodology. We outline the experimental setup, comparative performance against state of the art baselines, evaluating feature selection and model selection strategies, performance and evaluation analysis and, lastly, feature importance and interpretation.

5.1 Experimental setup

The experimental phase is structured around three interrelated analyses designed to comprehensively evaluate the effectiveness, robustness, and practical applicability of the proposed forecasting methodology.

The first component involves *benchmarking the proposed model against a suite of state-of-the-art baseline models* to assess its predictive performance relative to existing approaches in the field. The second analysis focuses on evaluating the impact of different *feature selection strategies* on model accuracy and generalization capability. This includes assessing the contribution of temporal, spatial, and contextual features through controlled ablation experiments. The final component examines key *operational metrics* such as computational efficiency i.e model size and inference latency to evaluate the deployment feasibility of the model under real-world constraints.

The overall experimental design is intended to rigorously evaluate model performance under realistic operational conditions, with particular emphasis on both short-term prediction accuracy (*e.g., next-hour forecasts*) and long-term trend capture (*e.g., multi-day forecasting horizons*). This dual focus ensures that the methodology is not only accurate in immediate predictions but also reliable in capturing evolving patterns over extended periods.

5.2 Evaluating feature selection and model selection strategies

This section presents an ablation study designed to assess the impact of different feature engineering strategies on model performance. The primary objective is to identify which approaches to feature construction and selection yield the most accurate and robust forecasts.

5.2.1 Forecasting scenarios and feature strategies

To understand how temporal distance impacts predictability and feature relevance, we defined two distinct forecasting tasks, each with a tailored feature engineering strategy. The first task is a short-term *prediction scenario* (Week 6 prediction). This scenario simulates operational, near-future forecasting, requiring the model to predict the immediate following week (Week 6) using training data from weeks 1-5. For this task, we hypothesized that recent temporal dynamics are highly predictive. We therefore employed a dynamic feature set, which incorporates the full suite of engineered features detailed in Section 4. This includes highly time-sensitive predictors such as hourly lags, short-term rolling window statistics, and expanding features that capture the most recent system state and momentum.

The second task is *long-term prediction scenario* (Week 11 prediction). This scenario tests the model's ability to generalize over a significant temporal gap, a common challenge in strategic network planning. The model must predict Week 11 using only training data from weeks 1-5, contending with a five-week data gap where *temporal distribution shift* (or concept drift) [38] is a major concern. To mitigate this, we curated a stable feature set. This set explicitly excludes features most susceptible to drift, such as short-term lags, expanding statistics, and certain volatile target encodings. The underlying hypothesis is that by forcing the model to rely on fundamental, time-invariant patterns such as stable weekly/daily cycles and core spatial hierarchies it will achieve better generalization over extended horizons.

5.2.2 Model selection and implementation

Two state-of-the-art Gradient Boosting Decision Tree (GBDT) models were chosen for their proven efficacy on structured, tabular data.

- **LightGBM**: A highly efficient GBDT framework utilizing a leaf-wise growth strategy, enabling it to converge quickly and capture complex patterns. Its speed is a significant advantage for experiments involving large datasets and extensive cross-validation.
- **CatBoost**: A GBDT framework distinguished by its novel handling of categorical features and its use of ordered boosting. This permutation-based approach inherently reduces target leakage during the training process, often leading to more robust and generalizable models.

Both models were implemented in Python. To address the severe right-skew of the target variable (*DLThpVol*), a ***square-root transformation*** was applied prior to training to stabilize variance and make the error distribution more amenable to learning. All predictions were inversely transformed back to the original scale before evaluation.

5.2.3 Evaluation protocol and metrics

A robust evaluation protocol was established to ensure the reliability and reproducibility of our findings. We employed a *10-fold stratified cross-validation* [39] methodology. Critically, stratification was performed based on the *base_station* identifier. This ensures that each fold contains a proportionally representative sample of data from all 30 base stations, preventing situations where a model is trained without seeing data from certain geographical clusters. This spatial stratification is essential for obtaining a reliable estimate of generalization performance in a real-world, heterogeneous network. The *MAE* [40] was selected as the primary performance metric. Its choice is motivated by two key properties of the data. First, MAE is less sensitive to the extreme outliers present in the long-tailed *DLThpVol* distribution compared to the Root Mean Squared Error (RMSE), providing a more stable measure of typical model performance. Second, MAE is directly interpretable in the original units of the target variable (i.e., throughput volume), facilitating clear communication of the model's accuracy.

5.2.4 *Hyperparameter optimization and model training*

Optimal model performance is highly dependent on hyperparameter configuration [41]. Separate hyperparameter tuning was conducted for each model and for each of the two forecasting scenarios using Optuna [42]. We utilized a systematic approach to find configurations that balance model complexity (e.g., *max_depth, num_leaves*) with regularization (e.g., *lambda_l1, lambda_l2*) to prevent overfitting. The final, optimized hyperparameters used for generating our results are detailed in Table 2 below.

During training within each cross-validation fold, the models were trained on the training partition and evaluated on the validation partition at each boosting iteration. This allows for the use of early stopping [43], a technique where training is halted if the validation performance does not improve for a specified number of rounds, preventing the model from overfitting to the training data. The learning curves from this process provide insight into model convergence.

Table 2 – Model parameters used

LightGBM (Long Term)	**LightGBM (Short Term)**	**Catboost (Long Term)**	**Catboost (Short Term)**
• learning_rate: 0.0205 • num_leaves: 254 • max_depth: 10 • feature_fraction: 0.6697 • bagging_fraction: 0.7229 • bagging_freq: 8 • min_child_samples: 100 • lambda_l1: 2.49e-6 • lambda_l2: 1.69e-8 • n_estimators: 1000	• learning_rate: 0.0830 • num_leaves: 151 • max_depth: 9 • feature_fraction: 0.7095 • bagging_fraction: 0.9362 • bagging_freq: 1 • min_child_samples: 22 • lambda_l1: 6.42 • lambda_l2: 0.0034 • n_estimators: 5000	• learning_rate: 0.020218465729343698 • depth: 9 • l2_leaf_reg: 1.339103723284128e-06 • random_strength: 6.000809910512735e-07 • bagging_temperature: 0.38040823680407604 • leaf_estimation_iterations: 7 • iterations: 15000	• learning_rate: 0.020218465729343698 • depth: 9 • l2_leaf_reg: 1.339103723284128e-06 • random_strength: 6.000809910512735e-07 • bagging_temperature: 0.38040823680407604 • leaf_estimation_iterations: 7 • iterations: 15000

5.3 Performance evaluation and analysis

This section presents the core empirical results of the study, focusing on the predictive performance of LightGBM, CatBoost, and a model ensemble combining both approaches. The evaluation is conducted across both short-term and long-term forecasting scenarios to assess the robustness and generalization capability of each method.

5.3.1 *Comparative performance against other provided baselines*

To contextualize the performance of the proposed models, we benchmark them against publicly available baseline results provided by the competition organizers [17]. These baselines encompass both classical and deep learning approaches, including iTransformer, PatchTST, DLinear, and transformer.

Fig. 6 illustrates the MAE between benchmark models (Hist.Avg, iTransformer, PatchTST, DLinear and transformer) and our proposed models (LightGBM and CatBoost) across two forecasting horizons; ***short-term predictions*** for Week 6 and ***long-term predictions*** for Week 11. These visual comparisons demonstrate the strong competitiveness of our approach relative to benchmark models.

In the short-term forecasting task, our proposed GBDT-based LightGBM and CatBoost models outperform all deep learning baselines. The proposed ensemble model achieves an MAE of 0.1919, surpassing the best- performing baseline model (iTransformer, MAE = 0.1967) by approximately 2.4%. This margin of improvement highlights the effectiveness of gradient-boosted decision trees when paired with carefully engineered temporal and contextual features.

In the case of the long-term forecasting scenario, our proposed ensemble model again attains the lowest MAE of 0.2261, outperforming all baselines, including the transformer architecture (MAE = 0.2331). This consistent outperformance across both forecasting horizons underscores the advantages of domain-informed feature engineering and ensemble learning in capturing temporal dynamics and mitigating model drift over extended periods.

These findings emphasize the importance of model interpretability, feature expressiveness, and computational efficiency, particularly in real-world deployment scenarios where inference latency and resource constraints are critical considerations. Overall, the results affirm the suitability of gradient-boosted tree models as a powerful alternative to deep learning architectures in structured time-series forecasting tasks.

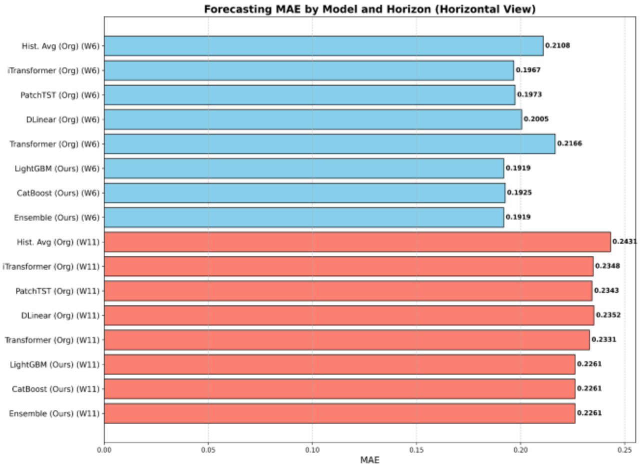

Figure 6 – Comparison of MAE between benchmark models and our proposed models (LightGBM and CatBoost)

5.3.2 Baseline: single-pipeline modeling results (no feature strategy split)

As an initial benchmark before implementing the dual-feature strategy described in Section 3.5, we trained GBDT models, specifically LightGBM and CatBoost, using a unified feature set that did not distinguish between short-term and long-term forecasting requirements. This single-pipeline approach serves as a useful benchmark to evaluate the effectiveness and added value of the dual-feature strategy.

We applied standard GBDT models using all engineered features, regardless of temporal horizon. The results, summarized in Table 3, demonstrate acceptable performance across both forecasting horizons. However, we observed notably diminished generalization accuracy on the long-term forecasting task, as shown in Table 3. This suggests that features informative for short-term predictions may lose relevance or introduce noise when applied to extended temporal horizons.

These limitations motivated the development of the dual-pipeline architecture, which enables model specialization based on temporal context and feature stability. By tailoring feature sets to each forecasting horizon, the proposed design aims to improve both predictive accuracy and robustness over time.

Table 3 – Results of the single pipeline

Model	Scenario	Feature Set	CV MAE	Leaderboard MAE	Δ (LB-CV)
LightGBM	Short Term (W6)	Single Pipeline	0.1923	0.1926	+0.0003
LightGBM	Long Term (W11)	Single Pipeline	0.1923	0.2302	+0.0379
Catboost	Short Term (W6)	Single Pipeline	0.1917	0.1918	+0.0001
Catboost	Long Term (W11)	Single Pipeline	0.1917	0.2356	+0.0439

5.3.3 Dual pipeline modelling results (with feature splits)

Table 4 presents an overview of the predictive performance achieved by the models under the dual-pipeline framework. The table reports two key evaluation metrics: the average MAE obtained from 10-fold cross-validation (referred to as CV MAE), which reflects the model's ability to generalize within the training distribution, and the final MAE on the held-out competition test set (referred to as Leaderboard MAE), which evaluates out-of-distribution generalization over time. These results provide insight into both the consistency of model performance during training and its robustness when applied to unseen temporal data.

Table 4 – Model performance comparison

Model	Scenario	Feature Set	CV MAE	Leaderboard MAE	Δ (LB-CV)
LightGBM	Short Term (W6)	Dynamic	0.1913	0.1925	+ 0.0012
Catboost	Short Term (W6)	Dynamic	0.1918	0.1919	+ 0.0001
Ensemble	Short Term (W6)	Dynamic	_	0.1919	_
LightGBM	Long Term (W11)	Stable	0.1971	0.2262	+ 0.0291
CatBoost	Long Term (W11)	Stable	0.1972	0.2262	+ 0.0290
Ensemble	Long Term (W11)	Stable	_	0.2261	_

5.3.4 Single-pipeline vs. dual-pipeline performance

In this study, we proposed a dual-modeling strategy to improve forecasting performance across different temporal horizons. To evaluate its effectiveness, we compare the results obtained under the dual-pipeline approach with those from the single-pipeline baseline. In the ***single-pipeline approach***, a unified feature set was used to train both short and long-term forecasts together, whilst in the ***dual-pipeline approach***, features were explicitly split into dynamic (short-term) and stable (long-term) sets, and separate models were trained for each forecasting horizon.

In the single-pipeline setup, as reported in Fig. 7, both models achieved reasonable performance across both tasks. However, we observed a notable drop in generalization accuracy for the long-term forecasting scenario. This suggests that features effective for short-term predictions introduced noise when applied to extended horizons, limiting model robustness.

By contrast, the dual-pipeline strategy significantly improved model specialization. In the short-term task (Week 6), the ensemble achieved a final Leaderboard MAE of 0.1919, outperforming all individual baselines and demonstrating strong competitiveness. For the long-term task (Week 11), the ensemble improved the best individual model results slightly, achieving a Leaderboard MAE of 0.2261 suggesting that both models successfully leveraged the more constrained and temporally stable feature set.

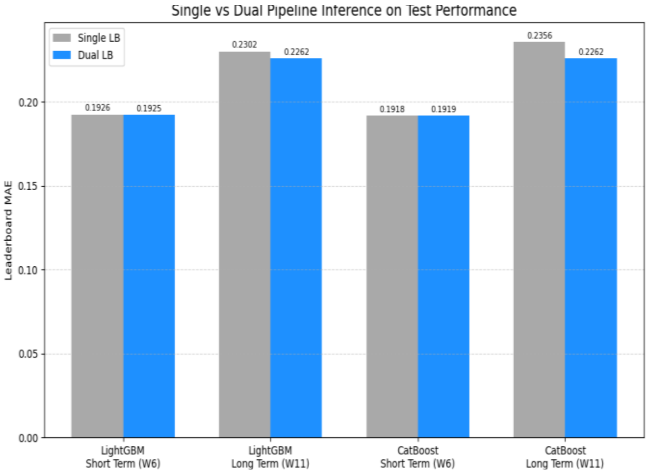

Figure 7 – Comparison between dual and single pipeline

This comparison confirms the effectiveness of separating temporal contexts during training. The dual-pipeline design not only enhances predictive accuracy but also improves interpretability and robustness by aligning feature relevance with forecasting horizons.

5.3.5 Analysis of results and model convergence (understanding beam-level traffic forecasting)

The results summarized in Table 4 offer several key insights into the characteristics and challenges of beam-level traffic forecasting.

Efficacy in short-term forecasting

In the Week 6 (short-term) forecasting scenario, both LightGBM and CatBoost models demonstrate strong and nearly identical performance. The small discrepancy between Cross-Validation MAE (CV MAE) and the final Leaderboard MAE (Δ < 0.001) indicates excellent generalization capability. This supports our hypothesis that the dynamic feature set enriched with recent temporal patterns is highly effective for near-future predictions, enabling accurate modeling of short-term traffic behavior.

Impact of temporal drift in long-term forecasting

The Week 11 (long-term) forecasting scenario highlights the significant challenge posed by temporal distribution shift. While the CV MAE (~0.197) suggests that the models effectively capture patterns within the training data, the Leaderboard MAE increases to 0.226, indicating a notable performance drop (Δ ≈ 0.029). This degradation reflects the diminishing relevance of short-term patterns over extended time horizons. However, the stable feature set plays a crucial role in maintaining model reliability by leveraging time-invariant structural features, thereby preventing complete model breakdown despite the five-week gap in temporal context.

Model convergence behavior

Figures 8 and 9 depict the learning curves of the CatBoost and LightGBM models, respectively, for both the short and long-horizon forecasting tasks.

In the long-horizon scenario, both models show a rapid decline in training and validation errors during the early boosting rounds, followed by a smooth plateau. The close alignment between the curves indicates stable convergence and effective generalization, suggesting that the models adequately capture longer-term temporal dependencies without substantial overfitting.

In contrast, the short-horizon models exhibit a gradual but continuous reduction in training error while the validation error stabilizes early, revealing mild overfitting. This pattern implies that, beyond a certain number of boosting iterations, further training primarily benefits the fit on the training data rather than improving predictive performance on unseen samples.

Overall, both CatBoost and LightGBM demonstrate consistent convergence behavior across forecasting horizons, with early stopping effectively preventing divergence and ensuring that training terminates near the point of optimal validation performance.

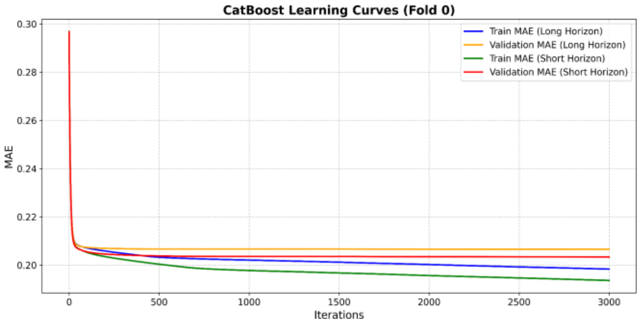

Figure 8 – Catboost training and validation curves for fold 0

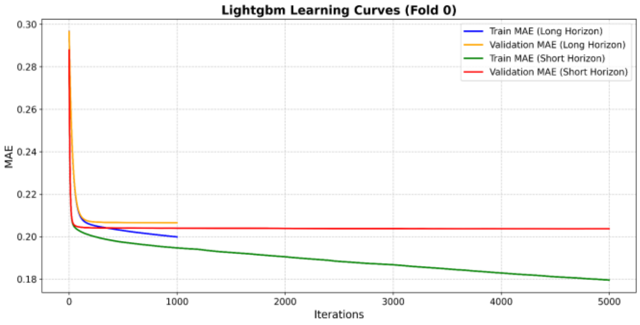

Figure 9 – Lightgbm training and validation curves for fold 0

Visualization of predicted hourly traffic volume trend

To gain a deeper understanding of the model's temporal behavior, we visualize the predicted hourly downlink throughput volume across three representative base stations: *Base station 0, Base station 1,* and *Base station 2*. This analysis highlights the model's ability to capture fine-grained spatio-temporal dynamics across geographically distributed network nodes.

As shown in Fig. 10, the model effectively learns distinct diurnal traffic patterns, characterized by pronounced peaks during daytime hours and troughs during the night. These periodic fluctuations reflect realistic user behavior and align with known daily activity cycles. (i) *Base station 0* exhibits the highest traffic intensity with clear, sharp peaks recurring every 24 hours, indicative of a densely populated or highly utilized area. (ii) *Base station 1* demonstrates a more stable and lower-volume profile, with moderate peaks and minimal variability. (iii) *Base station 2* shows an intermediate pattern, with elevated traffic during business hours but less intensity compared to Base station 0.

This visualization confirms that the forecasting model adapts its predictions to the unique traffic rhythms of individual base stations, supporting the hypothesis that *beam and cell-level traffic exhibit heterogeneous usage profiles*. Such insights are critical for operators seeking to implement targeted optimization strategies for specific network regions.

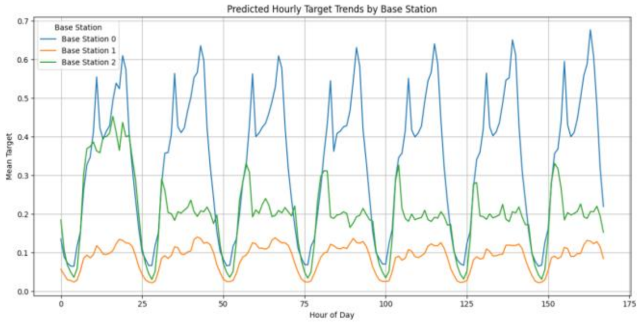

Figure 10 – Predicted hourly volume trends for base station 0, 1 and 2

Relative performance gains over baseline models

To quantify the efficacy of the proposed forecasting models, we conducted a comparative analysis against baseline results published by the competition organizers. These baselines include both traditional statistical techniques (e.g., historical average) and recent deep learning models such as iTransformer, PatchTST, DLinear, and transformer.

Table 5 reports the percentage improvement in *MAE* achieved by each of our models, LightGBM, CatBoost, and their ensemble on the official test set (Leaderboard MAE), relative to each baseline. Results are shown for both the short-term (Week 6) and long-term (Week 11) forecasting scenarios.

The findings reveal consistent performance gains across all baselines. In the short-term scenario, the ensemble model achieved up to 2.44% improvement over the best baseline (iTransformer) indicating the model's effectiveness in capturing recent temporal patterns. Similarly, in the long-term scenario, the ensemble approach delivered gains of nearly 3% over the best baseline in that horizon(transformer).

These results demonstrate that our GBDT-based framework, particularly the dual-pipeline strategy, not only matches but in many cases outperforms state-of-the-art deep learning methods. The improvements validate the strength of domain-informed feature engineering and ensemble learning in real-world traffic forecasting tasks.

Table 5 – Percentage improvement over benchmark models (MAE)

Benchmark Models	Light GBM W6	Light GBM W11	Catboost W6	Catboost W11	Ensem. W6	Ensem. W11
Hist. Avg	8.68%	6.95%	8.97%	6.95%	8.97%	6.99%
iTransformer	2.14%	3.66%	2.44%	3.66%	2.44%	3.71%
PatchTST	2.43%	3.46%	2.74%	3.46%	2.74%	3.50%
Dlinear	3.99%	3.83%	4.29%	3.83%	4.29%	3.87%
Transformer	11.13%	2.96%	11.40%	2.96%	11.40%	3.00%

5.3.6 Ensemble strategy

Given the complementary performance characteristics of LightGBM and CatBoost, a final weighted ensemble model was constructed by linearly combining their predictions. The ensemble was formulated as:

Ensemble prediction = 0.6 × CatBoost + 0.4 × LightGBM,

with weights derived from a combination of cross-validation performance and an analysis of feature importance patterns, which revealed distinct yet complementary sensitivities of the two models to different input features.

For the short-term forecasting task (Week 6), this ensemble approach achieved a final Leaderboard MAE of 0.1919, outperforming individual model results and yielding a competitive submission. In the long-term forecasting scenario (Week 11), the ensemble demonstrated performance comparable to the best individual models, suggesting that both models had converged toward similar solutions when constrained by the more stable and temporally invariant feature set.

This result highlights the value of ensemble learning in leveraging model diversity while reinforcing the importance of feature stability in long-horizon forecasting tasks.

5.4 Feature importance and interpretation

To understand the key drivers of the predictions and validate our feature engineering strategy, we analyzed the gain-based feature importances as calculated by LightGBM and CatBoost. This metric quantifies the total reduction in the loss function attributable to a given feature across all splits in the ensemble of trees.

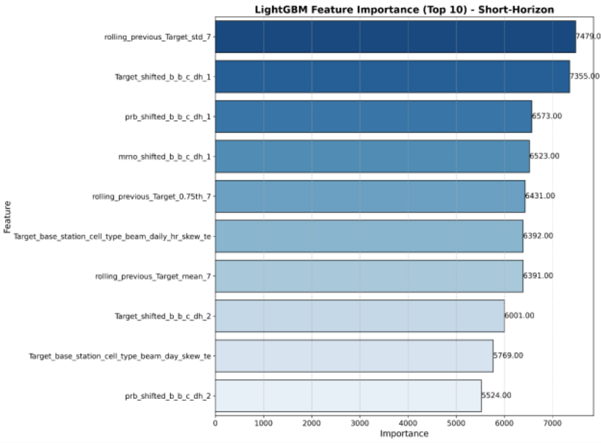

Figure 11 – Top 10 features for the short-term scenario

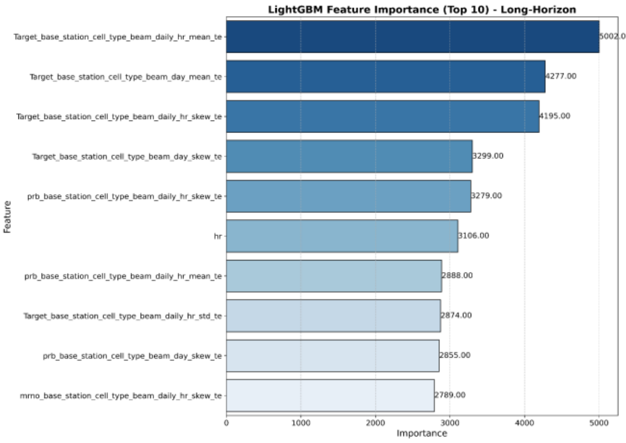

Figure 12 – Top 10 features for the long-term scenario

The feature importance analysis for the short-term forecasting scenario in Fig. 11 confirms that the model predominantly relies on recent, dynamic patterns in the data. The most influential features include indicators of weekly seasonality specifically, values from the same hour in the previous week which reflect the model's ability to capture recurring temporal rhythms in traffic behavior. Highly ranked features also include granular spatio-temporal interactions derived from target-encoded combinations of *beam*, *base_station*, and *daily_hr*. These engineered features enable the model to learn fine-grained behavioral patterns, such as "the average traffic for beam 5 at base station 10 during the 9 AM hour," demonstrating its capacity to localize predictions based on both spatial and temporal contexts.

Additionally, features extracted from short-term rolling windows such as the standard deviation of the target variable over the past seven hours rank among the most important. These statistics allow the model to adapt its predictions dynamically in response to recent load fluctuations, further enhancing its responsiveness to evolving network conditions.

In stark contrast to the short-term forecasting scenario, the feature importance ranking for the long-term task shown in Fig. 12 reveals a clear and meaningful shift in the model's reliance on different types of predictors. Deprived of access to recent dynamic features, the model adapts by prioritizing structurally stable and temporally invariant characteristics of the data.

This explicit shift in feature reliance between the two scenarios provides powerful validation for our dual-strategy approach. It demonstrates that the models successfully adapted their learning to the nature of the forecasting task, exploiting transient dynamics when available and relying on stable structural patterns when forced to generalize over a long temporal gap.

The most influential features fall into three primary categories.

Core spatial hierarchy: Target-encoded features based on *beam* and *base_station* identity dominate the importance rankings. These features serve as a stable baseline, capturing the average traffic level associated with each spatial entity independently of short-term fluctuations.

Fundamental cyclical patterns: Basic temporal features such as *daily_hr* and *day_of_week* emerge as critical predictors. In the absence of fine-grained dynamic signals, the model leverages these reliable, long-term periodicities as its primary temporal reference.

Long-term averages: Aggregated weekly mean features provide a robust historical context, offering generalized insights into traffic behavior that are less sensitive to transient noise or outlier events.

This explicit shift in feature dependence between the two forecasting scenarios offers strong empirical validation of our dual-strategy modeling approach. It demonstrates that the models successfully adapt their learning mechanisms to the nature of the prediction task leveraging high-resolution, dynamic patterns when available, and falling back on stable, structural relationships when generalizing over extended time horizons.

6. CONCLUSION

This study challenged the prevailing trend of increasing model complexity in network traffic forecasting. We demonstrated that a GBDT-based framework, grounded in deliberate, domain-specific feature engineering, can achieve superior performance over state-of-the-art deep learning benchmarks for the task of beam-level DLThpVol prediction. Our approach prioritizes interpretability, computational efficiency, and practical deployability, critical requirements for real-world network management that are often overlooked by more complex “black-box” models.

Our proposed dual-pipeline architecture, leveraging LightGBM and CatBoost, proved highly effective. By specializing separate models for distinct forecasting horizons, one sensitive to short-term autoregressive patterns and another focused on long-term seasonal stability, the framework achieved state-of-the-art accuracy. Our ensemble model attained a Mean Absolute Error (MAE) of 0.1919 for short-term (Week 6) and 0.2261 for long-term (Week 11) forecasts, outperforming established benchmarks including iTransformer, PatchTST, and DLinear.

The primary contribution of this work is the empirical evidence that for structured time-series data with strong periodicities, a method that explicitly encodes domain knowledge can be more powerful than generic automated feature extraction. This work provides a robust and efficient forecasting blueprint that serves as a critical enabler for downstream network optimization. While we do not propose a new resource allocation algorithm, the high accuracy of our forecasts provides the foundation needed to reduce safety margins in resource provisioning, mitigate network congestion, and enhance energy efficiency in 5G and future cellular systems.

Ultimately, our findings advocate for a pragmatic approach to machine learning in network management, one that proves high performance and actionable insight are not mutually exclusive.

REFERENCES

[1] Imielinski, T., & Badrinath, B. R. (1994). Mobile wireless computing: Challenges in data management. *Communications of the ACM*, *37*(10), 18-28.

[2] Zhu, C., Shu, L., Hara, T., Wang, L., Nishio, S., & Yang, L. T. (2014). A survey on communication and data management issues in mobile sensor networks. *Wireless Communications and Mobile Computing*, *14*(1), 19-36.

[3] Aouedi, O., Le, V. A., Piamrat, K., & Ji, Y. (2025). Deep learning on network traffic prediction: Recent advances, analysis, and future directions. *ACM Computing Surveys*, *57*(6), 1-37.

[4] Li, F., Zhang, Z., Chu, X., Zhang, J., Qiu, S., & Zhang, J. (2023). A meta-learning based framework for cell-level mobile network traffic prediction. *IEEE Transactions on Wireless Communications*, *22*(6), 4264-4280.

[5] Ali, E., Ismail, M., Nordin, R., & Abdulah, N. F. (2017). Beamforming techniques for massive MIMO systems in 5G: overview, classification, and trends for future research. *Frontiers of Information Technology & Electronic Engineering*, *18*, 753-772.

[6] Ahmed, I., Khammari, H., Shahid, A., Musa, A., Kim, K. S., De Poorter, E., & Moerman, I. (2018). A survey on hybrid beamforming techniques in 5G: Architecture and system model perspectives. *IEEE Communications Surveys & Tutorials*, *20*(4), 3060-3097.

[7] Andreev, S., Petrov, V., Dohler, M., & Yanikomeroglu, H. (2019). Future of ultra-dense networks beyond 5G: Harnessing heterogeneous moving cells. *IEEE Communications Magazine*, *57*(6), 86-92.

[8] Zhang, C., Patras, P., & Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials*, *21*(3), 2224-2287.

[9] Moayedi, H. Z., & Masnadi-Shirazi, M. A. (2008, August). Arima model for network traffic prediction and anomaly detection. In *2008 International Symposium on Information Technology* (Vol. 4, pp. 1-6). IEEE.

[10] Tikunov, D., & Nishimura, T. (2007, September). Traffic prediction for mobile network using Holt-Winter’s exponential smoothing. In *2007 15th International Conference on Software, Telecommunications and Computer Networks* (pp. 1-5). IEEE.

[11] Wikle, C. K., & Zammit-Mangion, A. (2023). Statistical deep learning for spatial and spatiotemporal data. *Annual Review of Statistics and Its Application*, *10*(1), 247-270.

[12] Gómez, J. A., Patiño, J. E., Duque, J. C., & Passos, S. (2019). Spatiotemporal modeling of urban growth using machine learning. *Remote Sensing*, *12*(1), 109.

[13] Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*, 21.

[14] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *30*.

[15] Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

[16] Cortez, P., Rio, M., Rocha, M., & Sousa, P. (2012). Multi-scale Internet traffic forecasting using neural networks and time series methods. *Expert Systems*, *29*(2), 143-155.

[17] "Spatio-Temporal Beam-Level Traffic Forecasting Challenge," ITU AI/ML Challenge, 2024. [Online]. Available: https://zindi.africa/competitions/spatio-temporal-beam-level-traffic-forecasting-challenge

[18] L. Fechete et al., Goal-Oriented Time-Series Forecasting: Foundation Framework Design, arXiv:2504.17493 (2025).

[19] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748-1764.

[20] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021, May). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligenc* (Vol. 35, No. 12, pp. 11106-11115).

[21] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, *34*, 22419-22430.

[22] Agiwal, M., Roy, A., & Saxena, N. (2016). Next generation 5G wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *18*(3), 1617-1655.

[23] Mao, Q., Hu, F., & Hao, Q. (2018). Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *20*(4), 2595-2621.

[24] Aouedi, O., Le, V. A., Piamrat, K., & Ji, Y. (2025). Deep learning on network traffic prediction: Recent advances, analysis, and future directions. *ACM Computing Surveys*, *57*(6), 1-37.

[25] Wang, X., Zhou, Z., Xiao, F., Xing, K., Yang, Z., Liu, Y., & Peng, C. (2018). Spatio-temporal analysis and prediction of cellular traffic in metropolis. *IEEE Transactions on Mobile Computing*, *18*(9), 2190-2202.

[26] Wang, Z., Hu, J., Min, G., Zhao, Z., Chang, Z., & Wang, Z. (2022). Spatial-temporal cellular traffic prediction for 5G and beyond: A graph neural networks-based approach. *IEEE Transactions on Industrial Informatics*, *19*(4), 5722-5731.

[27] Jiang, W. (2022). Cellular traffic prediction with machine learning: A survey. *Expert Systems with Applications*, *201*, 117163.

[28] Pavlyuk, D. (2017). Short-term traffic forecasting using multivariate autoregressive models. *Procedia Engineering*, *178*, 57-66.

[29] Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. Philosophical *Transactions of the Royal Society A*, *379*(2194), 20200209.

[30] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). The M5 competition: Background, organization, and implementation. *International Journal of Forecasting*, *38*(4), 1325-1336.

[31] Kaya, A. Ö., & Viswanathan, H. (2021, March). Deep learning-based predictive beam management for 5G mmWave systems. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 1-7). IEEE.

[32] Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), 388-427.

[33] Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019, December). The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 3285-3292). IEEE.

[34] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2022). Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

[35] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*(1), 54-74.

[36] Han, Z., Zhao, J., Leung, H., Ma, K. F., & Wang, W. (2019). A review of deep learning models for time series prediction. *IEEE Sensors Journal*, *21*(6), 7833-7848.

[37] Pargent, F., Pfisterer, F., Thomas, J., & Bischl, B. (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics*, *37*(5), 2671-2692.

[38] Zhang, Z., Wang, X., Zhang, Z., Li, H., Qin, Z., & Zhu, W. (2022). Dynamic graph neural networks under spatio-temporal distribution shift. *Advances in Neural Information Processing Systems*, *35*, 6074-6089.

[39] Browne, M.W., 2000. Cross-validation methods. *Journal of Mathematical Psychology*, *44*(1), pp. 108-132.

[40] Hodson, T.O., 2022. Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development Discussions*, *2022*, pp.1-10.

[41] Feurer, M., & Hutter, F. (2019). *Hyperparameter* optimization (pp. 3-33). Springer International Publishing.

[42] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019, July). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2623-2631).

[43] Prechelt, L. (2002). Early stopping-but when?. In *Neural Networks: Tricks of the trade* (pp. 55-69). Berlin, Heidelberg: Springer Berlin Heidelberg.

AUTHORS

Stephen Kolesh received his B.Sc. in computer science in 2023. He has held roles in various data science and machine learning capacities, most recently at Sportserve, where he works on applied ML experimentation, model deployment, and evaluation. His professional achievements include leading Zindi globally as the top ranked competitive machine learning engineer and data scientist. His current research interests lie in scalable machine learning pipelines, model interpretability, and responsible deployment of AI systems.

Thomas Basikolo is a programme coordinator in the Study Groups and Policy Department of the ITU Telecommunication Standardization Bureau (TSB). He coordinates and manages the AI for Good’s Machine Learning activities (including ML5G), is ITU lead of the Green Computing Pillar of the Green Digital Action, and is an advisor to the ITU-T Focus Group on AI-Native Networks. Prior to joining ITU, he worked as a research engineer in the Engineering Department of Microwave Factory Co., Ltd, Tokyo, Japan.

He received a PhD in electrical and computer engineering from Yokohama National University, Japan. He is a recipient of multiple best paper awards, the IEEE AP–S Japan Student Award and the Young Engineer of the year award by IEEE AP-S Japan in 2018.

He has co-authored peer-reviewed journal and conference papers, predominantly in the areas of wireless communications and antenna engineering. He serves as a reviewer of IEEE and IEICE Journals. His interests include machine learning, deep learning, artificial intelligence and network science, and their applications in wireless networks, as well as how technology can be used to advance the UN SDGs.