*applied sciences*

MDPI

*Article*

# A Meta-Learning-Based Framework for Cellular Traffic Forecasting

**Xiangyu Liu, Yuxuan Li, Shibing Zhu, Qi Su * and Changqing Li**

School of Space Information, Space Engineering University, Beijing 101416, China; liuxypaper@163.com (X.L.); liyuxuan@hgd.edu.cn (Y.L.); zsbpaper@163.com (S.Z.); lcqqcl5577@hgd.edu.cn (C.L.)
* Correspondence: suqi@hgd.edu.cn

**Abstract**

The rapid advancement of 5G/6G networks and the Internet of Things has rendered mobile traffic patterns increasingly complex and dynamic, posing significant challenges to achieving precise cell-level traffic forecasting. Traditional deep learning models, such as LSTM and CNN, rely heavily on substantial datasets. When confronted with new base stations or scenarios with sparse data, they often exhibit insufficient generalisation capabilities due to overfitting and poor adaptability to heterogeneous traffic patterns. To overcome these limitations, this paper proposes a meta-learning framework—GMM-MCM-NF. This framework employs a Gaussian mixture model as a probabilistic meta-learner to capture the latent structure of traffic tasks in the frequency domain. It further introduces a multi-component synthesis mechanism for robust weight initialisation and a negative feedback mechanism for dynamic model correction, thereby significantly enhancing model performance in scenarios with small samples and non-stationary conditions. Extensive experiments on the Telecom Italia Milan dataset demonstrate that GMM-MCM-NF outperforms traditional methods and meta-learning baseline models in prediction accuracy, convergence speed, and generalisation capability. This framework exhibits substantial potential in practical applications such as energy-efficient base station management and resilient resource allocation, contributing to the advancement of mobile networks towards more sustainable and scalable operations.

**Keywords:** meta-learning; cellular network traffic forecasting; Gaussian mixture models; multi-component mechanism; few-shot learning; LSTM; negative feedback correction

check for updates

Academic Editor: Christos Bouras

Received: 8 October 2025
Revised: 24 October 2025
Accepted: 28 October 2025
Published: 30 October 2025

**Citation:** Liu, X.; Li, Y.; Zhu, S.; Su, Q.; Li, C. A Meta-Learning-Based Framework for Cellular Traffic Forecasting. *Appl. Sci.* **2025**, *15*, 11616. https://doi.org/10.3390/app152111616

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

With the rapid advancement of 5G/6G networks and Internet of Things (IoT) technologies, mobile network traffic has exhibited explosive growth. Within the current cellular network infrastructure, such factors as population mobility, public holidays, or special circumstances may lead to predictable network congestion, while certain cells may simultaneously exhibit power consumption far exceeding demand. To address escalating user requirements and reduce energy wastage, research into edge computing within a cellular-level traffic forecasting and drone-assisted framework is paramount. For instance, accurate forecasting of future traffic loads across multiple base stations enables dual-pronged optimisation: deploying drone swarms to assist overloaded sites, thereby enhancing user experience, while implementing sleep modes for temporarily idle base stations to reduce energy expenditure.

Currently, precise cellular-level traffic forecasting is a core enabling technology for dynamic network resource scheduling, base station energy management, and service quality assurance. It has consequently garnered significant attention in both academic research and engineering applications worldwide. However, existing forecasting methods still face three major challenges:

Firstly, the dilemma of model complexity. Whilst statistical models and shallow learning models exhibit lower complexity and training costs, real-world network traffic variations are vastly more intricate than any statistical or shallow learning model can capture, rendering them inadequate for complex spatio-temporal characteristics. Secondly, challenges stem from task heterogeneity: traffic patterns across different functional zones within cities exhibit significant divergence, leaving single models lacking in generalisation potential. Thirdly, the challenge of few-shot learning: the performance of deep learning-based forecasting methods heavily depends on the training dataset. When training data is insufficient, deep learning models exhibit overfitting.

To address the first challenge, the hierarchical optimisation architecture of meta-learning effectively reduces model complexity. By decomposing complex traffic distributions into K Gaussian components, each associated with a long short-term memory (LSTM)-based learner focused on learning specific patterns, the overall model complexity is diminished.

To address the second challenge, a Gaussian mixture model (GMM) is employed as the meta-learner within the meta-learning architecture. This captures the frequency-domain characteristics of traffic sequences, quantifies task differences, and probabilistically adapts prediction tasks, thereby significantly enhancing adaptation effectiveness across heterogeneous tasks.

To address the third challenge, this paper introduces an MCM upon the hierarchical meta-learning architecture and dedicated modelling for diverse traffic patterns. The MCM dynamically synthesises weight vectors across components, thereby effectively mitigating prediction errors under small sample sizes.

Given the above considerations, this paper incorporates meta-learning for cellular network traffic forecasting. The proposed model employs a multi-layer long short-term memory (LSTM) network as the base learner, with each base processor handling a distinct fundamental task—corresponding to each cellular network. The meta-task of the meta-learning model involves initialising the weight vectors for the base learners of each basic task according to the distinct meta-features of each cellular network. To accomplish this meta-task, a GMM is adopted as the meta-learner. Employing the meta-learning model effectively balances prediction accuracy and training cost. Numerical experiments demonstrate that the proposed algorithm significantly improves post-training prediction accuracy and learning efficiency.

The principal contributions of this paper are summarised as follows:

- Proposing a GMM-based meta-learner to replace the KNN meta-learner in ML-TP. GMM enables probabilistic modelling of the meta-feature space, capturing latent structures in task distributions.
- The MCM is introduced to overcome the limitations of KNN's rigid assignment in ML-TP. MCM initialises the base learner for new tasks by synthesising weight vectors from multiple Gaussian components.
- A prediction–correction negative feedback mechanism (NF) is designed to dynamically adjust GMM parameters during long-term predictions.

## 2. Related Work

Accurate cellular traffic forecasting is crucial for network operators to optimise resource allocation, reduce energy consumption, and enhance quality of service (QoS). Recent rapid advancements in machine learning, particularly deep learning techniques, have shifted traffic forecasting methods from traditional statistical models towards sophisticated intelligent frameworks. As systematically summarised in reviews by Jiang et al. [1] and Wang et al. [2], this research has become a prominent area of focus. A review of existing studies reveals that cellular traffic forecasting methods can be broadly categorised based on their core mechanisms and technical approaches:

- Hybrid prediction models based on signal decomposition. Their core concept employs a "decomposition–prediction–reconstruction" paradigm: first utilising signal processing techniques such as empirical mode decomposition or wavelet transforms to decompose non-stationary raw traffic sequences into relatively stationary sub-components, then predicting and fusing these separately to reduce modelling complexity. Such methods demonstrate unique advantages in handling complex fluctuation patterns.
- Deep learning-based end-to-end prediction models, which represent the current mainstream research direction. These can be further subdivided based on network architecture and spatio-temporal information processing approaches: recurrent neural networks (RNNs/LSTM/GRUs) and their hybrid variants, which focus on capturing temporal dependencies; convolutional neural networks (CNNs) and their spatio-temporal fusion variants, adept at extracting spatial features; and graph neural networks (GNNs), capable of directly modelling base station network topologies.
- Emerging models adapted from other domains, such as Transformers utilising self-attention mechanisms to capture long-range dependencies, and multi-task learning frameworks designed to enhance generalisation capabilities through cross-task knowledge sharing.

The aforementioned existing models generally suffer from limitations, including heavy reliance on large amounts of labelled data, insufficient generalisation capability in small-sample novel scenarios, and limited adaptability to dynamic changes in non-stationary traffic patterns. This precisely constitutes the starting point for this paper's proposed introduction of a meta-learning paradigm to construct a rapid adaptive forecasting framework tailored for cellular traffic.

### *2.1. Signal Decomposition-Based Hybrid Forecasting Models*

Cellular traffic data typically exhibits a mixture of periodic, trend, and random fluctuations across multiple timescales, presenting characteristic nonlinear and non-stationary properties. To effectively capture these complex patterns, numerous studies have adopted a "decomposition–prediction–reconstruction" approach. Such methods first employ signal decomposition techniques to split the original complex traffic sequence into several relatively stable sub-sequences. Predictions are then made independently for each sub-sequence, with the final prediction value synthesised from the combined results. Duan Amin et al. [3] proposed a hybrid neural network approach based on quadratic decomposition. This method first decomposes traffic into frequency and trend components using CEEMDAN and K-Shape clustering, followed by a secondary VMD decomposition of the frequency sequence. A neural network integrating CBAM, BiLSTM, and attention mechanisms then performs multidimensional forecasting. To address long-term forecasting challenges, Jiang Donghao et al. [4] employed Empirical Wavelet Transform (EWT) for data decomposition, creatively utilising the NeuralProphet model to forecast low-frequency components while processing medium-to-high frequencies via MLP. The work of Wei Yan [5] explored temporal similarity and periodicity in cellular traffic, laying theoretical foundations for

subsequent decomposition approaches. Gu, M.C. [6] combined Empirical Mode Decomposition (EMD) with LSTM, employing noise statistics to denoise the decomposed Intrinsic Mode Functions (IMFs), thereby enhancing model robustness. From another perspective, Zang et al. [7] employed wavelet transforms to preprocess flow data, providing cleaner inputs for subsequent machine learning models. Earlier research by Zhang Lin [8] also explored wavelet scale coupling and particle swarm optimisation for detecting multi-cluster network flows, reflecting early efforts to analyse flow characteristics using signal processing techniques. These works collectively demonstrate that refined data decomposition can effectively reduce the inherent complexity of raw sequences, enabling prediction models to better learn patterns across different frequency components and thereby enhance forecasting accuracy. This offers significant insight for our investigation into stationary versus non-stationary traffic patterns: for complex non-stationary traffic, preprocessing through pattern decomposition or component separation may assist meta-learning models in more efficiently capturing commonalities and idiosyncrasies across heterogeneous tasks.

### *2.2. Deep Learning-Based End-to-End Forecasting Models*

In recent years, deep learning models represented by RNNs, CNNs, and their variants have achieved significant success in traffic forecasting due to their powerful nonlinear feature learning capabilities, gradually becoming mainstream approaches.

#### 2.2.1. Recurrent Neural Networks and Their Variants

Table 1 summarises the current state of research on traffic forecasting models based on RNNs and their variants. Analysis indicates that RNNs, particularly their enhanced architectures like LSTM and GRU, are inherently suited to processing time series data due to their sequential modelling capabilities. The multi-channel sparse LSTM model designed by Zhang et al. [9] not only captures multi-source traffic information but also enables the model to adaptively focus on different time points through sparse connections. Research by Jaffry and Hasan [10] further validated RNNs' efficacy in cellular traffic forecasting. Moreover, Jaffry [11] compared LSTMs with traditional ARIMA and feedforward neural networks (FFNN), highlighting LSTMs' superior training speed, particularly suited for modelling scenarios requiring rapid response. To address more complex spatio-temporal dependencies, Li et al. [12] proposed a hybrid LSTM-TCN model incorporating a Multi-Head Self-Attention (MHSA) mechanism. This approach aims to strengthen internal data correlations through MHSA while leveraging LSTMs and TCNs to capture long-term and short-term dependencies and global features, respectively. Alsaade and Al-Adhaileh [13] combined single exponential smoothing (SES) with LSTM, first using SES to preliminarily smooth complex traffic data before passing it to LSTM for prediction, achieving favourable results. Comparative experiments by Azari et al. [14] demonstrate that LSTMs typically outperform ARIMA models when training on large datasets with high sampling frequencies. Kurri et al. [15] even applied LSTM models to blockchain-based 4G LTE network traffic forecasting, showcasing their adaptability. Furthermore, Li Huidong et al. [16] combined LSTMs with Adaptive Neuro-Fuzzy Inference Systems (ANFISs) to construct hybrid models enhancing predictive performance. These studies affirm the robust capability of RNNs and their variants in capturing temporal dependencies, yet also suggest to this paper that relying solely on temporal information may prove insufficient for complex city-level traffic forecasting. This has spurred researchers to explore more intricate network architectures.

**Table 1.** Research status of flow prediction models based on RNNs and their variants.

| Researcher (Year) | Model Used | Method/Features | Principal Contributions/Conclusions |
| --- | --- | --- | --- |
| Zhang Zhengwan et al. (2021) [9] | Multi-channel sparse LSTM | The introduction of multi-source inputs and sparse connections enables the model to adaptively focus on different time points. | Enhances the model's ability and flexibility to capture multi-source traffic information. |
| Jaffry and Hasan (2020) [10] | RNN | Validates the fundamental efficacy of RNN-based models for cellular traffic forecasting tasks. | Confirmed the potential of RNN-type models for handling such time series problems. |
| Jaffry (2020) [11] | LSTM | Conducted comparative experiments with ARIMA and FFNN. | Noted that LSTM possesses advantages in training speed, making it more suitable for scenarios requiring rapid response. |
| Li Weiyue et al. (2024) [12] | LSTM-TCN-MHSA | A hybrid model incorporating Multi-Head Self-Attention (MHSA), utilising LSTM and TCN to capture long-term/short-term and global dependencies, respectively. | Enhances spatio-temporal feature extraction by introducing an attention mechanism to handle complex dependencies. |
| Alsaade and Al-Adhaileh (2021) [13] | SES-LSTM | Data is preprocessed using Single Exponential Smoothing (SES) before prediction by LSTM. | Data smoothing preprocessing enhances LSTM prediction performance on complex traffic data. |
| Azari et al. (2019) [14] | LSTM | Compared with ARIMA models under conditions of large-scale, high-sampling-frequency data. | It was demonstrated that, when sufficient data is available, LSTM generally outperforms traditional ARIMA models. |
| Kurri et al. (2021) [15] | LSTM | Applied LSTM to blockchain-based 4G/LTE network traffic forecasting. | Demonstrated the flexibility and applicability of LSTM models across different network architectures and application scenarios. |
| Li Huidong et al. (2022) [16] | LSTM-ANFIS | Combining LSTM with Adaptive Neuro-Fuzzy Inference Systems (ANFISs) to construct a hybrid model. | Through hybrid modelling, the performance and robustness of the prediction model are further enhanced. |

#### 2.2.2. Convolutional Neural Networks and Spatio-Temporal Fusion Models

Table 2 summarises the current state of traffic forecasting research based on CNNs and their hybrid models. Analysis indicates that CNNs excel at extracting spatial features, such as images, hence their widespread application in processing grid-based cellular traffic data to capture spatial correlations between base stations. Zheng et al. [17] designed a lightweight CNN model employing a parallel branch architecture to extract spatio-temporal features from recent and periodic data, respectively. By incorporating base station density information derived from K-Means clustering as external features, they achieved low-complexity yet precise predictions. Huang, Dongyi et al. [18] designed the Spatio-Temporal Fully Connected Convolutional Network (ST-FCCNet), which employs a specialised unit structure to capture spatial dependencies between any two regions within a city, integrating external information for prediction. Zhang et al. [19] were among the earliest to propose using densely connected CNNs for urban cellular traffic forecasting, demonstrating the potential of CNNs in this domain. To better fuse spatio-temporal features, numerous studies have designed hybrid architectures. For instance, Zhang, Deyang et al. [20] employed a one-dimensional CNN to extract features from sequences of varying temporal granularity, subsequently aggregated via a Graph Attention Network (GAT). Feng et al. [21] designed DeepTP, an end-to-end neural network capable of effectively processing mobile cellular traffic. Zhang et al. [22] constructed the Hybrid Spatio-Temporal Network (HSTNet), incorporating deformable convolutions and attention mechanisms into the CNN model to enhance robustness. Concurrently, Ni Feixiang [23] combined k-NN algorithms to analyse spatio-temporal correlations and employed wavelet-Elman neural networks for prediction,

providing insights for early research on spatio-temporal information fusion. These works collectively emphasise the importance of simultaneously modelling temporal and spatial dimensions, offering conceptual guidance for designing the spatio-temporal heterogeneity meta-learning framework presented herein.

**Table 2.** Current status of traffic forecasting research based on CNNs and hybrid models.

| Researcher | Model Employed | Method/Core Approach | Principal Contributions/Characteristics |
| --- | --- | --- | --- |
| Zheng Songzhi et al. [17] | Lightweight CNN | Employing a parallel branch architecture to extract recent and periodic features, respectively, whilst incorporating base station density derived from K-Means clustering as an external feature. | Achieves low-complexity, high-precision forecasting whilst emphasising the effective integration of external features. |
| Huang Dongyi et al. [18] | ST-FCCNet (Spatio-Temporal Fully Connected Convolutional Network) | Designed with a special unit structure to capture spatial dependencies between any two regions within a city, while integrating external information. | Achieves modelling of extensive spatial dependencies, overcoming the limitations of local convolutions. |
| Zhang et al. [19] | Dense-Connected CNN (DenseNet) | Among the earliest to apply densely connected CNNs to city-level cellular traffic forecasting. | This validated the immense potential and effectiveness of CNN architectures in this domain. |
| Zhang Deyang et al. [20] | 1D-CNN + GAT (Graph Attention Network) | Employed a one-dimensional CNN to extract features at different temporal granularities, subsequently aggregated via a Graph Attention Network (GAT). | Effectively fuses temporal features with complex spatial (graph structure) correlations. |
| Feng et al. [21] | DeepTP (End-to-End Neural Network) | Designed an end-to-end deep learning model specifically for mobile cellular traffic forecasting. | Provides a complete end-to-end forecasting solution, simplifying the workflow. |
| Zhang et al. [22] | HSTNet (Hybrid Spatio-Temporal Network) | Extends CNNs by incorporating deformable convolutions and attention mechanisms. | Enhances the model's adaptability and robustness to complex spatio-temporal patterns. |
| Ni Feixiang et al. [23] | k-NN + Wavelet-Elman Neural Network | Combines k-NN algorithm analysis of spatio-temporal correlations with a wavelet-Elman neural network for prediction. | Provides valuable insights for hybrid models integrating spatio-temporal information at an early stage. |

#### 2.2.3. Graph Neural Networks (GNNs) and Network Topology Modelling

Table 3 summarises the current state of research on traffic prediction based on Graph Neural Networks (GNNs). Urban cellular networks can be naturally viewed as a graph structure, where base stations are nodes and the geographical or functional connections between them are edges. The advent of Graph Neural Networks (GNNs) provides powerful tools for learning directly on such non-Euclidean data. Li Zhehui et al. [24] proposed a

weighted multi-graph convolutional network approach. This constructs multiple graph relationships based on distances between base stations, data correlations, and attention weights, then fuses features from different graphs through a weighted mechanism. Guo Xinyu et al. [25] designed a spatio-temporal graph convolutional network (STGCN)-based prediction method to enhance forecasting of dynamic spatio-temporal correlations. Wang Yu et al. [26] and Fu Bohan et al. [27] employed GNN predictions (particularly T-GCN [27]) for dynamic base station switching decisions to achieve energy conservation. Wang et al. [28] applied GNNs to spatio-temporal traffic forecasting for 5G and future networks. Yao et al. [29] proposed a Multi-View Spatio-Temporal Graph Neural Network (MVSTGN) to capture complex spatio-temporal dependencies from multiple perspectives. Zhao et al. [30] designed spatio-temporal aggregated graph convolutional networks for more efficient traffic prediction. Zhou et al. [31] employed graph convolutional networks combined with transfer learning to address traffic forecasting in large-scale cellular networks. Zhao et al. [32] integrated GNNs with user handover information, further enhancing prediction accuracy. These GNN-based approaches model spatial dependencies between base stations more directly and flexibly than traditional grid-based CNN methods.

**Table 3.** Research status of flow prediction models based on RNNs and their variants.

| Researcher (Year) | Model/Method Used | Core Idea/Technical Features | Principal Contributions/Application Objectives |
| --- | --- | --- | --- |
| Li Zhehui et al. (2025) [24] | Weighted Multi-Graph Convolutional Network | Constructs multi-graph relationships through distance, correlation, and attention mechanisms, while weighting and integrating features. | More finely characterises the complex multi-spatial dependencies between base stations. |
| Guo Xinyu et al. (2023) [25] | Spatio-Temporal Graph Convolutional Network (STGCN) | Designs an STGCN-based framework to simultaneously capture spatio-temporal dynamic correlations. | Improves the modelling and prediction accuracy of dynamic spatio-temporal correlations. |
| Wang Yu et al. (2024) [26]/Fu Bohan et al. (2024) [27] | T-GCN (Temporal Graph Convolutional Network) | Utilises GNN prediction outputs (e.g., future load) as inputs to dynamically control base station switching. | Applies predictive models to network energy conservation, achieving a closed-loop system of prediction and optimisation. |
| Wang et al. (2022) [28] | GNN | Applies GNNs to traffic forecasting problems in 5G and B5G networks. | Validating the efficacy and potential of GNNs in modelling future advanced network traffic. |
| Yao et al. (2021) [29] | Multi-view Spatio-temporal Graph Network (MVSTGN) | Constructing graph structures from multiple perspectives (views) to capture complex spatio-temporal dependencies. | Enhances the model's understanding of complex spatio-temporal patterns through multi-view learning. |
| Zhao et al. (2021) [30] | Spatio-Temporal Aggregated Graph Convolutional Network | Designs novel aggregation mechanisms aimed at enhancing the efficiency of spatio-temporal information propagation and aggregation. | Achieve more efficient, computationally less costly flow prediction. |
| Zhou et al. (2022) [31] | Graph Convolutional Networks (GCN) + Transfer Learning | Employ GCNs to model spatial relationships, combined with transfer learning to address data scarcity issues. | Transferring learned knowledge to new domains to tackle the challenge of small-sample prediction in large-scale networks. |
| Zhao et al. (2020) [32] | GNN + User Handover Information | Integrates user handover behaviour information across different base stations into the graph neural network model. | Further enhances prediction accuracy by introducing mobility semantic features. |

### *2.3. Transformers and Other Emerging Models*

In recent years, the Transformer model, which has made significant strides in natural language processing, has been introduced into time series forecasting due to its self-attention mechanism's ability to capture long-range dependencies. Liu et al. [33] proposed ST-Transformer, a spatio-temporal Transformer model specifically designed for cellular traffic forecasting. Gu et al. [34] designed a more complex spatio-temporal Transformer

network for city-level traffic analysis and forecasting. Furthermore, several studies have begun to explore the potential of multi-task learning. Wei et al. [35] proposed a series of prediction models based on deep multi-task learning. These treat predictions at different temporal granularities or traffic forecasts for distinct service types as separate yet related tasks. By sharing the underlying network architecture, they learn common features, thereby enhancing accuracy while reducing computational complexity. Zhang Jiaoyang and Sun Li [36] also employed multi-dataset joint prediction methods, incorporating attention mechanisms to learn correlations across different services. JointSTNet [37] proposed a unified pre-training framework that captures shared patterns in spatio-temporal traffic data through multi-task learning, aligning its core philosophy with meta-learning's task generalisation objective. The Electric Vehicle Frequency-Response Framework [38] combines game-theoretic incentives with deep reinforcement learning (DRL) to dynamically optimise multi-agent system behaviour, demonstrating DRL's adaptability in complex dynamic environments. These works underscore the role of pre-training and multi-agent collaboration in enhancing model generalisation capabilities, providing theoretical underpinnings for this paper's meta-learning framework. Concurrently, online learning algorithms have garnered attention, such as the work by Mehri et al. [39] exploring their application in cellular traffic forecasting. These emerging models and approaches offer fresh perspectives on traffic prediction, particularly as multi-task learning and meta-learning share conceptual proximity—both aim to enhance learning efficiency and performance on target tasks by leveraging information from related tasks.

### *2.4. Summary*

Reviewing existing literature on cellular traffic forecasting reveals an evolving landscape of research methodologies. As demonstrated by the comparative study of Santos Escriche et al. [40], no single model achieves optimal performance across all scenarios; different models exhibit distinct strengths at varying prediction horizons. Although deep learning approaches—such as the cross-domain big data deep transfer learning model proposed by Zhang et al. [41] and the spatio-temporal attention convolutional network designed by Zhao et al. [42]—achieve high accuracy on specific datasets, existing research still faces several common challenges. These are precisely the issues addressed by this study. Firstly, prevailing models predominantly rely on extensive, high-quality training datasets. When deployed to novel, data-scarce base stations or performing granular predictions within confined areas, these models often struggle to generalise effectively due to the "few-shot" problem. Secondly, real-world urban network environments are dynamically evolving; shifts in functional zones, major events, or emergencies can induce abrupt and persistent alterations in traffic patterns, generating non-stationary time series. Existing models are predominantly trained under static or quasi-static assumptions, lacking rapid adaptive capabilities for such dynamic evolution patterns and exhibiting insufficient robustness. This research therefore aims to address these challenges by introducing a meta-learning framework. The objective is to enable models to "learn how to learn"—that is, to form "prior knowledge" capable of rapidly adapting to new tasks by drawing experience from a large number of diverse historical prediction tasks. Although existing meta-learning approaches have demonstrated some efficacy in cellular traffic forecasting—such as the ML-TP framework achieving knowledge transfer between tasks via KNN algorithms—they exhibit notable limitations: KNNs rely on the rigid assignment of Euclidean distances, are sensitive to noise and outliers, and cannot probabilistically model task distributions. This results in limited generalisation capabilities when handling heterogeneous tasks and small-sample scenarios. To address this, this paper introduces a GMM as the meta-learner, achieving soft partitioning of task distributions through probabilistic clustering of the

meta-feature space. GMM not only captures latent task patterns more robustly but also provides more reasonable initial weights for new vehicle tasks via posterior probabilities. Furthermore, this paper proposes a multi-component synthesis mechanism (MCM) and an NF to enhance the model's adaptability and robustness at both the initial weight allocation and dynamic optimisation levels.

## 3. GMM-ML-TP Cellular Traffic Forecasting Model

This section first outlines the traffic forecasting framework proposed herein—GMM-MCM-NF—before explicitly defining its three core components: the base learner tasked with predicting traffic load for individual cells, the meta-learner designed to enhance the base learner's predictive accuracy and learning efficiency, and the correction mechanism for long-term forecasting tasks.

As illustrated in Figure 1, the GMM-MCM-NF model's workflow comprises three core steps. The first step involves meta-feature extraction and meta-learner training: each cell's traffic load data is converted into a frequency-domain signal via the Fast Fourier Transform (FFT). Subsequently, following the methodology described in Section 3.1, the real and imaginary parts of five dominant frequency components are extracted to construct a 10-dimensional meta-feature vector. These vectors represent each cell's traffic pattern and are used to train Gaussian mixture models. The trained GMM meta-learners generate initialisation weights for base learners based on distinct cellular traffic characteristics. Step two involves meta-knowledge-based weight initialisation and prediction: GMM-generated weights serve as initial parameters for LSTM networks to forecast network traffic load. Step Three: Prediction-Correcting Feedback Mechanism: This mechanism dynamically evaluates the responsibility weights of each Gaussian component by calculating the LSTM's prediction error. It penalises components with larger errors and optimises the GMM parameters accordingly, thereby enhancing the quality of subsequent task weight selection.

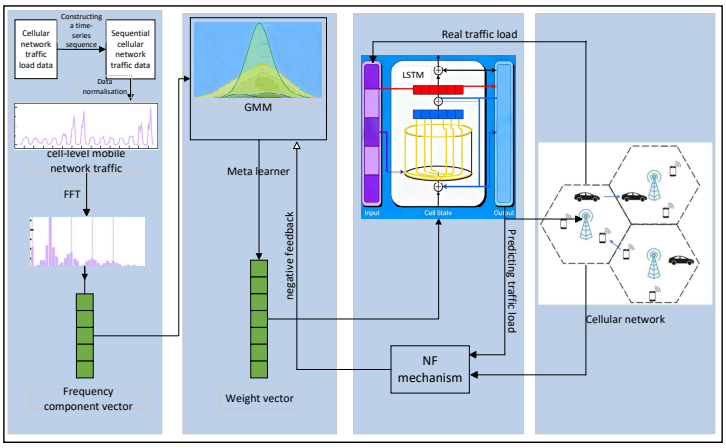


**Figure 1.** Schematic diagram of the GMM-MCM-NF model.

### 3.1. Dataset and Preliminary Analysis

This section primarily introduces the dataset utilised in this study, comprising real mobile network traffic records. Additionally, it presents mathematical analyses of cellular traffic in both the time and frequency domains.

(1) Spatial Gridding and Cell Definition

This study employs the mobile network dataset provided by Telecom Italia's Big Data Challenge [43] programme. The dataset comprises approximately 3 million traffic records collected in Milan between 1 November 2013 and 1 January 2014. The Milan area was divided into 10,000 grids, each representing a square region with a side length of 235 m,

serving as the fundamental spatial unit for this research. Each record contains a timestamp, grid ID, and mobile traffic load (i.e., traffic payload). Given that the grid size approximates the coverage area of a 5G base station, each grid is defined as a "cell" in this paper.

(2) Time Series Construction

This paper utilises the mobile network dataset provided by Telecom Italia's Big Data Challenge programme. The dataset encompasses approximately three million traffic records generated within the city of Milan between 1 November 2013 and 1 January 2014. The region was partitioned into a grid of 10,000 cells, each representing a square region with a side length of 235 m. Each record contains a timestamp, grid ID, and mobile traffic load (i.e., traffic payload). Hereafter, each grid is referred to as a cell, as its size is comparable to the coverage area of a 5G base station.

For analytical convenience, the entire dataset's temporal span was divided into consecutive one-hour time intervals: $\Delta t = 1h$ the $p$ cell in the $t$ traffic load for the $p$th cell during the $t$th time interval is calculated as follows:

$$\begin{gathered} l'_p[t] = \sum_{r \in \{Z^+ : ID_r = p, (t-1)\cdot\Delta t < t_r \le t\cdot\Delta t\}} vol_r \\ p \in \left\{1, ..., 10000\right\} \\ t \in \left\{1, 2, ..., N\right\} \\ r \in \left\{1, 2, ...\right\} \end{gathered} \tag{1}$$

Assuming there are $N$ time intervals, $r$ denotes the traffic record index, $ID_r$ represents the cell ID for record $r$, $t_r$ indicates the timestamp for record $r$, and $vol_r$ signifies the traffic load for record $r$. The time series of the traffic load for the $p$th cell is represented by the vector $l'_p = (l'_p[1], l'_p[2], ..., l'_p[N])$.

(3) Data Normalisation

To facilitate deep learning training, the traffic load time series for each cell is normalised. To prevent data leakage, normalisation parameters are strictly calculated within each cell's meta-training set time period. Specifically, for the $p$th cell, whose meta-training set time series is $\mathbf{X}_p^{meta-train}$, the normalisation formula is as follows:

$$l_p[t] = \frac{X - \min(\mathbf{X}_p^{meta-train})}{\max(\mathbf{X}_p^{meta-train}) - \min(\mathbf{X}_p^{meta-train})} \tag{2}$$

where $\min(\mathbf{X}_p^{meta-train})$ and $\max(\mathbf{X}_p^{meta-train})$ denote the minimum and maximum traffic load values within the cell's meta-training set, respectively. For the test task set (new cells), normalisation parameters are computed using their own limited fine-tuning set (data from the preceding week) to simulate the real-world scenario where new base stations lack future information. The normalised traffic load time series for the $p$th cell, denoted as $l_p = (l_p[1], l_p[2], \ldots, l_p[N])$, constitutes the "cell-level network traffic" data utilised in subsequent analyses.

Figure 2 illustrates the normalised traffic load patterns across three distinct cells during the same two-week period. It is evident that traffic load exhibits distinct temporal characteristics across different cells.

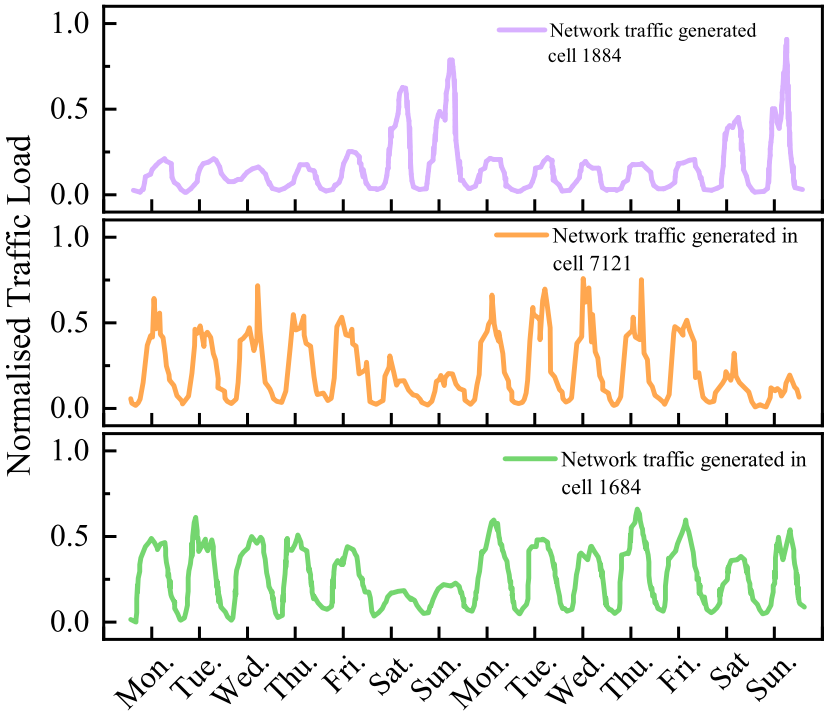


**Figure 2.** Time-domain traffic load (normalised) for cells 1884, 7121, and 1684.

(4) Meta-feature extraction

Within each cell, although daily variations in traffic load differ, they exhibit fixed periodic patterns on a weekly basis. To quantify the temporal correlation of cellular traffic load, the autocorrelation coefficients of the normalised traffic load vectors for cells $p$ can be calculated as follows:

$$cor_{p,Tlag} = \frac{\sum_{l=1}^{N-T_{lag}} (l_p[t] - \bar{l}_p)(l_p[t + T_{lag}] - \bar{l}_p)}{\sum_{t=1}^{N} (l_p[t] - \bar{l}_p)^2} \tag{3}$$

Figure 3 displays the autocorrelation coefficients of the normalised traffic load vectors for different cells. Calculations reveal that the autocorrelation coefficients across cells exhibit similar results. As shown in Figure 2, we observe that the autocorrelation coefficient of the normalised cell traffic load vector peaks when the time lag $T_{lag}$ is an integer multiple of 24 h.

Based on the above analysis, a discrete periodic signal may be constructed from the normalised traffic load vector of the cell, as follows:

$$\tilde{l}_p[t] = \begin{cases} l_p[t], 1 \le t < T \\ l_p[t \mod T], t < 1 \quad \text{or} \quad t \ge T \end{cases} \tag{4}$$

where $T = 168$ denotes the total number of hours in a week, then the FFT of this discrete periodic signal is obtained:

$$\begin{cases} F_p\left(k \cdot \frac{2\pi}{T}\right) = \sum_{t=0}^{T-1} \tilde{l}_p[t] (W_T)^{kt}, k = 0, 1, \cdots, T-1 \\ W_T = e^{-j\frac{2\pi}{T}} \end{cases} \tag{5}$$

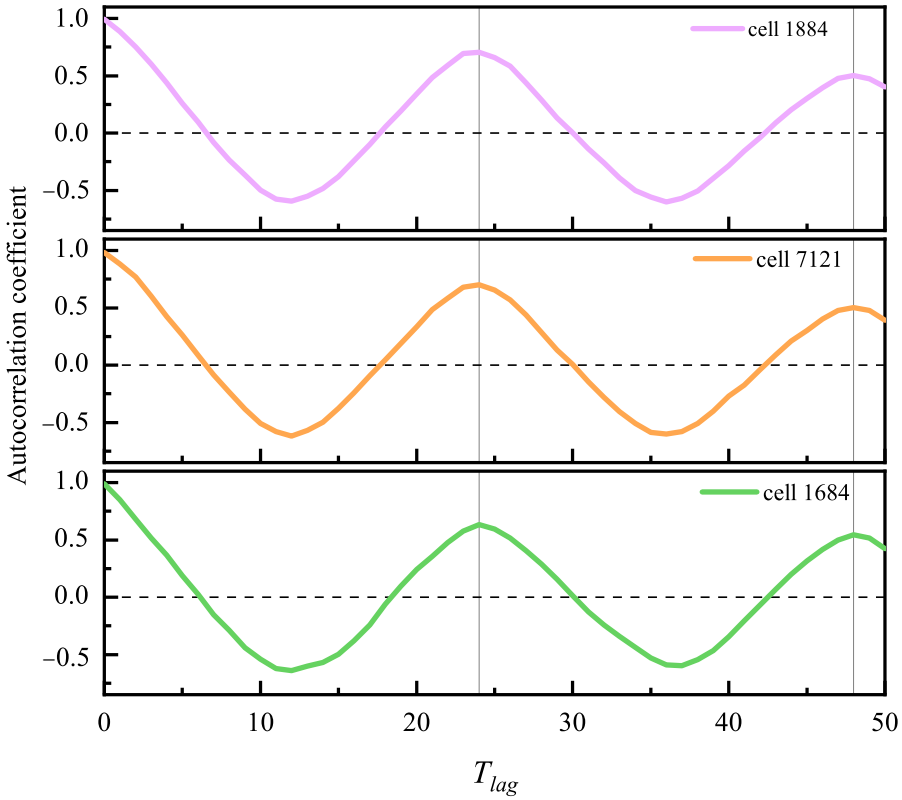


**Figure 3.** Autocorrelation coefficients of normalised traffic load vectors across different cells.

This paper selects five specific frequency components w = $\pi/84$, w = $\pi/12$, w = $\pi/6$, w = $\pi/4$, and w = $\pi/3$ corresponding to periods of 1 week, 1 day, 12 h, 8 h, and 6 h, respectively (sinewaves), as the meta-features for the cellular network. The rationale is as follows: This selection is based on a systematic analysis of the frequency-domain energy distribution across all cellular traffic sequences in the dataset. The specific selection rationale and validation process are as follows:

(1) Generation of feature candidate pool based on global energy spectrum

First, the FFT amplitude spectrum of the normalised traffic load vector was computed for all cells $|F_p(\omega)|$. Subsequently, the global average energy spectral density was calculated $\bar{S}(\omega)$

$$\bar{S}(\omega) = \frac{1}{N}\sum_{p=1}^{N}|F_p(\omega)|^2 \tag{6}$$

This energy spectrum quantifies the importance of each frequency component $\omega$ in terms of average significance across the entire dataset. As shown in Figure 4, $\bar{S}(\omega)$ it exhibits pronounced peaks near five frequency points, corresponding to periods confirmed as 1 week, 1 day, 12 h, 8 h, and 6 h, respectively. This constitutes the preliminary basis for the feature candidate pool in this paper.

(2) To ensure the universality of these five frequency components, we further examined their significance across different cellular subpopulations.

Step 1: We randomly sampled multiple subsets based on each cell's total traffic volume and geographical location (e.g., high-traffic cells, low-traffic cells, city-centre cells, suburban cells).

Step 2: For each subset, we repeated Step 1 to compute its average energy spectrum.

Results: Across all tested subsets, the aforementioned five frequency components consistently represented the most prominent peaks in the energy spectrum. We conducted paired t-tests comparing the average energy at these five frequency points with that at adjacent frequencies, revealing in all instances $p \leq 0.01$ that the energy at these five points was significantly higher than background noise and neighbouring frequencies, indicating they represent robust, cross-regional common patterns.

(3) Interpretability of physical significance

The selected frequency components possess explicit, human-activity-driven physical significance:

$\omega = \pi/84$ Weekly cycle: captures macro-level flow pattern differences between weekdays and weekends.

$\omega = \pi/12$ Diurnal cycle: Reflects fundamental human rhythms, constituting the core pattern of diurnal traffic alternation.

$\omega = \pi/6\pi/4\pi/3$(12/8/6 h cycles): These sub-diurnal and shorter cycles may correspond to refined intra-day activity patterns such as lunch breaks and commuting peaks (morning/evening), providing the model with richer temporal detail.

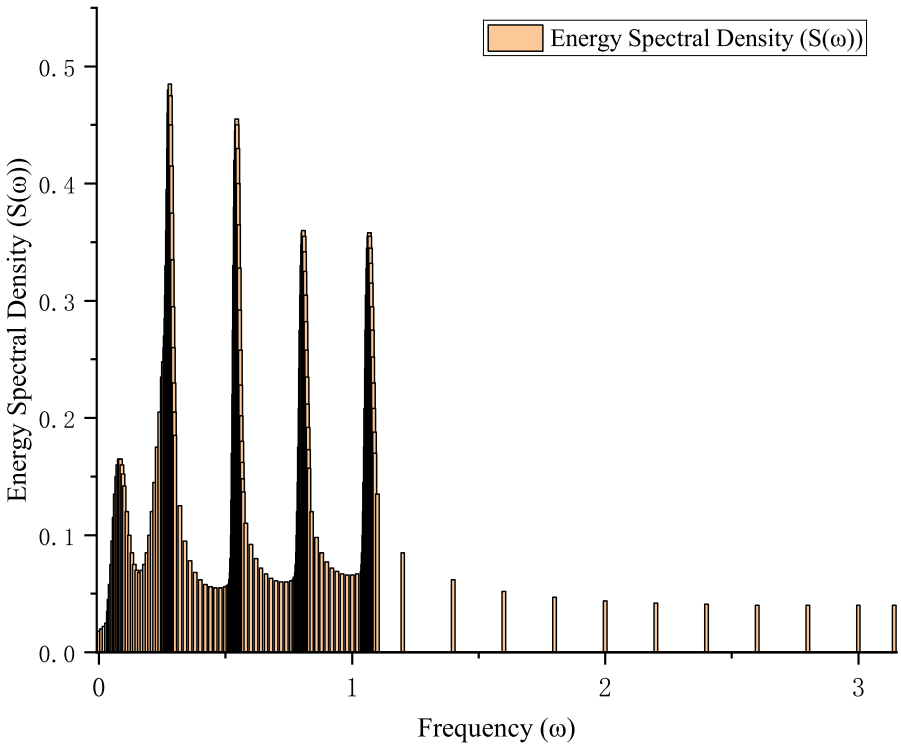


**Figure 4.** FFT results of normalised traffic load vectors for cells 1884, 7121, and 1684.

The amplitude of the FFT results for the cellular traffic load in Figure 2 is shown in Figure 4. The real and imaginary parts of the five principal frequency components of cell $p$ can form a 10-dimensional principal frequency component vector, i.e., the cell's metaphysical feature:

$$\begin{aligned}\Gamma_p = &[R(F_p(\pi/84)), S(F_p(\pi/84)),\\ &R(F_p(\pi/12)), S(F_p(\pi/12)),\\ &R(F_p(\pi/6)), S(F_p(\pi/6)),\\ &R(F_p(\pi/4)), S(F_p(\pi/4)),\\ &R(F_p(\pi/3)), S(F_p(\pi/3))]\end{aligned} \tag{7}$$

where $R(\cdot)$ and $S(\cdot)$ represent the real and imaginary parts of the complex numbers, respectively.

By calculating the Pearson correlation coefficient $\rho_{p,q}$ between any two honeycomb normalised flow load vectors $p$ and $q$, we obtain

$$\rho_{p,q} = \frac{cov(l_p, l_q)}{\sigma_{l_p}\sigma_{l_q}} \tag{8}$$

where $cov(\cdot)$ denotes the covariance. $\sigma_{l_p}$ and $\sigma_{l_q}$ represent the standard deviation of the normalised traffic load for cell $p$ and $q$, respectively.

Figure 5 depicts data sources comprising 1000 randomly selected cellular grid pairs from the complete dataset. This diagram aims to illustrate the overall statistical relationships across the sampled dataset. Due to the substantial sample size and random sampling methodology employed, individual cellular identifiers are not directly annotated within the figure. Figure 5 was derived by calculating the Pearson correlation coefficient between the normalised traffic load vectors of these 1000 cellular pairs, alongside the Euclidean distance between their principal frequency component vectors. As illustrated, the Pearson correlation coefficient between two cells' normalised traffic load vectors diminishes as the Euclidean distance between their principal frequency component vectors increases. This indicates that if the principal frequency component vectors of two cells are proximate in Euclidean space, the two cells tend to exhibit similar traffic patterns in the time domain. Conversely, if the principal frequency component vectors of two cells are distant in Euclidean space, this suggests that the traffic patterns of the two cells in the time domain are markedly different.

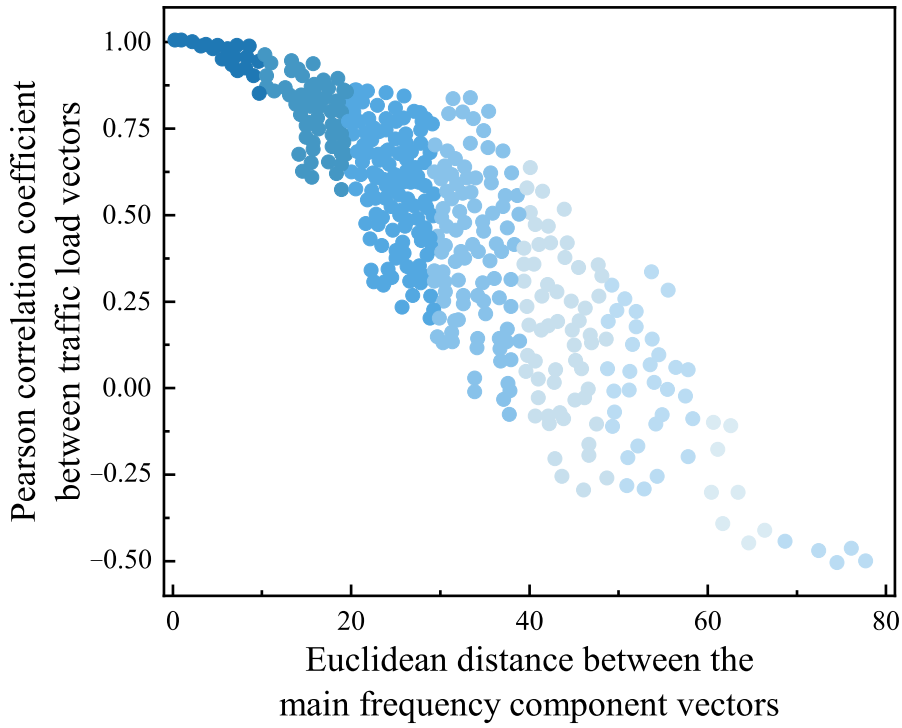


**Figure 5.** Relationship between the Pearson correlation coefficient of two cellular traffic vectors and the Euclidean distance of their corresponding principal frequency component vectors.

It should be noted that the Telecom Italia dataset employed herein originates from 2013–2014, reflecting traffic patterns characteristic of 3G/4G networks. With the proliferation of 5G/6G and IoT technologies, contemporary network traffic may exhibit heightened volatility, low-latency requirements, and diverse service types (e.g., video streaming, IoT

device communications). Nevertheless, this study focuses on validating the meta-learning framework's generalisability under task heterogeneity and small-sample scenarios, rather than precisely simulating the latest traffic patterns. The proposed method does not rely on specific data distributions. In Section 4.5.7, the Telecom Shanghai Dataset is employed to validate the model's cross-dataset generalisation capability [44].

### *3.2. Probabilistic Modelling in Feature Space*

Let the meta-feature vector for the historical baseline task be denoted as $\{\Gamma_p\}_{p=1}^{N} \subset R^D(D = 10)$, where $N$ represents the total number of historical tasks. This paper assumes that the distribution of meta-feature vectors adheres to a Gaussian mixture model (GMM), which follows a mixture model composed of $K$ Gaussian distributions:

$$p(\Gamma_p \mid \theta) = \sum_{k=1}^{K} \pi_k N(\Gamma_p \mid \mu_k, \sum_k) \tag{9}$$

The model parameters are $\theta = \{\pi_k, \mu_k, \sum_k\}_{k=1}^{K}$ , where $\pi_k$ represents the mixture coefficients in the $k$th Gaussian component, denoting the probability of selecting the kth Gaussian model during data generation. $\mu_k$ is the mean of the $k$th Gaussian model, and $\sum_k$ denotes the variance of the $k$th Gaussian model. Each Gaussian distribution corresponds to a category of base station traffic features with similar traffic patterns (e.g., commercial areas, residential areas, etc.).

For the $k$th Gaussian component, the corresponding set of optimal weight vectors is denoted as

$$W_k = \left\{ w_i \mid \Gamma_i \in \quad \theta_k \right\} \tag{10}$$

The class centres of the weight vectors are

$$\overline{w}_k = \frac{1}{|W_k|} \sum_{w_i \in W_k} w_i \tag{11}$$

When inputting the meta-feature $\theta_k$ for a new task $q$, the posterior probabilities for each component are as follows:

$$\gamma_k(\Gamma_q) = \frac{\pi_k N(\Gamma_q \mid \mu_k, \sum_k)}{\sum_{j=1}^{K} \pi_j N(\Gamma_q \mid \mu_j, \sum_j)} \tag{12}$$

The component $k^* = \arg\max_k \gamma_k(\Gamma_q)$ with the highest posterior probability is selected, and initial weights are allocated based on the following two rules.

The first rule, termed as the SCM, operates on the principle that "if the meta-features of two tasks are probabilistically similar, their optimal model weights should also be similar". It selects the base model weight with the closest probability distribution to the meta-feature vector as the initial weight:

$$w_q^{init} = arg_{w_i \in W_k^*} \|\Gamma_q - \Gamma_i\|_{\sum_{k^*}^{-1}} \tag{13}$$

where $\|\cdot\|_{\sum_{k^*}^{-1}}$ is used to compute the Mahalanobis distance, enhancing the measurement of local similarity by introducing the covariance structure of the component $k^*$.

The second approach employs an MCM. Its initial weight allocation rule does not directly select the parameter set corresponding to a single Gaussian component. Instead, it synthesises a new parameter set based on posterior probability. This mechanism fully

leverages the probabilistic modelling advantages of GMMs, avoiding information loss caused by rigid allocation. The allocation rule is as follows:

$$w_q^{init} = \sum_{k=1}^{K} \gamma_k \theta_k \tag{14}$$

where $\theta_k$ is the initial weight vector associated with the $k$th Gaussian component.

### *3.3. Verification of Feature Distribution Assumptions*

The validity of Gaussian mixture models rests upon the assumption that the meta-feature vector $f_p$ can be adequately approximated by multiple multivariate Gaussian distributions. This paper employs the following multi-faceted, quantitative methods to validate this core assumption.

#### 3.3.1. Intra-Component Multinormality Test

Given that GMM assumes data within each component follow multivariate Gaussian distributions, we first utilise the trained GMM to assign each sub-feature vector $f_p$ to its most probable component (i.e.) $\arg\max_k \gamma(z_{pk})$, then conduct multivariate normality tests on data within each component $k$ separately. We employ the Mardia test, which is based on multivariate skewness and kurtosis.

The multivariate skewness statistic is defined as

$$b_{1,d} = \frac{1}{N_k^2} \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \left[ (f_i - \hat{\mu}_k)^T \widehat{\Sigma}_k^{-1} (f_j - \hat{\mu}_k) \right]^3 \tag{15}$$

whose asymptotic distribution is $\chi^2$ distributed with degrees of freedom $d(d+1)(d+2)/6$, where $d = 10$ is the number of parameters.

The multivariate kurtosis statistic is defined as

$$b_{2,d} = \frac{1}{N_k} \sum_{i=1}^{N_k} \left[ (f_i - \hat{\mu}_k)^T \widehat{\Sigma}_k^{-1} (f_i - \hat{\mu}_k) \right]^2 \tag{16}$$

Its asymptotic distribution is normal.

(For all components) $k = 1, \ldots, K$. The test results indicate that at the $\alpha = 0.01$ significance level under the significance level, the majority of components (>95%) fail to reject the null hypothesis of multivariate normality, supporting the assumption of component-wise distribution within GMM.

#### 3.3.2. Model Comparison and Goodness-of-Fit Assessment

This paper compares the goodness-of-fit between GMM and other potential distribution models to demonstrate the superiority of GMM.

Likelihood Ratio Test (LRT): This paper compares GMM with a single multivariate Gaussian distribution (i.e., GMM with K = 1). The LRT statistic is

$$\Lambda = -2 \log \left( \frac{L(M_{Single})}{L(M_{GMM})} \right) \tag{17}$$

where $L(\cdot)$ is the maximum likelihood value of the model. The statistic $\Lambda$ follows an approximate $\chi^2$ distribution with degrees of freedom equal to the increase in parameters from K = 1 to the optimal K value (determined by BIC). The test result ($p$-value $< 0.001$) strongly rejects the hypothesis of a single Gaussian model, supporting that the data originate from a mixture distribution.

Bayesian Information Criterion (BIC) comparison: We further contrasted the GMM with a mixture model based on the t-distribution (MoT), which is more robust to heavy-tailed distributions. BIC is defined as:

$$\mathrm{BIC} = \ln(N) \cdot |\theta| - 2\ln(L) \tag{18}$$

Results indicate that the GMM's BIC value (−12,450) is significantly lower than the MoT's BIC (−11,980), demonstrating that the GMM represents a superior choice in balancing model complexity and fit.

3.3.3. Cluster Structure Validation

GMM essentially performs probabilistic clustering. We employ the silhouette coefficient to assess clustering quality. For a data point $i$, its silhouette coefficient $s(i)$ is computed as follows:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d(f_i, f_j) \tag{19}$$

$$b(i) = \lim_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(f_i, f_j) \tag{20}$$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{21}$$

where $C_i$ is the cluster to which point $i$ belongs, $d$ is the Euclidean distance. Based on the posterior probability $\gamma(z_{pk})$, a hard assignment is performed (assigning the point to the component with the highest probability). The average silhouette coefficient calculated across all principal feature vectors is 0.65 (>0.5), indicating a clear and reasonable clustering structure, with compactness within components and good separation between them.

In summary, through distribution verification, model comparison, and clustering evaluation, this paper validates the rationality and effectiveness of probabilistic modelling using Gaussian mixture models for frequency-domain feature vectors.

*3.4. Prediction Correction Mechanism Based on the SCM*

Within the SCM, when assigning initial weights for new forecasting tasks, we select only the weight vector corresponding to the Gaussian component with the highest posterior probability. This rigid allocation method is simple and efficient but may overlook useful information from other components. To optimise the model's performance in long-term forecasting, we introduce a correction mechanism that dynamically adjusts the GMM's mixture coefficients based on forecast errors. The steps for operating the SCM mechanism are as follows:

(1) Initial weight allocation

$$w_{\text{init}} = w_{k^*}, \quad k^* = \arg\max_k \gamma_k \tag{22}$$

where $\gamma_k$ represents the posterior probability of task $q$ belonging to the $k$th Gaussian component, reflecting the similarity between the task and the component. $k^*$ denotes the index of the component with the highest posterior probability, i.e., the most similar task pattern category. The weight vector corresponding to the Gaussian component with the highest posterior probability is selected as the initial weight. This equates to selecting the "expert experience" most similar to the current task from historical tasks.

(2) Prediction Error Calculation

Assume a new task $q$ with a validation set $D_q = \{(x_i, y_i)\}_{i=1}^{M}$. The prediction error $\varepsilon_q$ is defined as the mean squared error (MSE):

$$\varepsilon_q = \frac{1}{|D_q|} \sum_{(x,y)\in D_q} (f(x; w_{\text{init}}) - y)^2 \tag{23}$$

where $f()$ is the base learner parameterised by $w_{\text{init}}$. The error $\varepsilon_q$ quantifies the base learner's underperformance on task $q$, directly reflecting the suitability of the initial weights. A larger error indicates that the selected initial weights deviate further from the true requirements of task $q$, while a smaller error indicates a closer match to the true requirements of task $q$.

(3) Weighting Coefficient Update

This paper adjusts the mixing coefficient based on error, penalising poorly performing components:

$$\pi_k^{\text{new}} = \frac{\pi_k \cdot \exp(-\beta \cdot \delta_{k,k^*} \cdot \varepsilon_q)}{\sum_{j=1}^{K} \pi_j \cdot \exp(-\beta \cdot \delta_{j,k^*} \cdot \varepsilon_q)} \tag{24}$$

where $\delta_{k,k^*}$ (is an indicator function, equal to 1 when) $k = k^*$ and 0 otherwise; $\beta$ is a sensitivity coefficient, used to control the adjustment magnitude. When $\varepsilon_q \to 0$, $\exp(-\beta \cdot \varepsilon_q) \to 1$, (the weight of component) $k^*$ (remains largely unchanged). When $\varepsilon_q \to \infty$, $\exp(-\beta \cdot \varepsilon_q) \to 0$, the weight of component $k^*$ the weight is significantly reduced. The denominator ensures the sum of all mixing coefficients equals 1, maintaining the normality of the probability distribution. Through error-driven mixing coefficient adjustment, the model can automatically reduce the priority of poorly performing components, enhancing the overall robustness of the meta-learner.

### *3.5. Prediction Correction Mechanism Based on MCM*

The prediction correction mechanism based on the multi-component synthesis mechanism primarily achieves collaborative multi-component correction by distributing error responsibility through the posterior probability distribution of the GMM. Weighted experience from multiple similar tasks is typically more robust than that from a single most-similar task. Through soft allocation and collaborative optimisation, it better handles situations with blurred task boundaries. The steps for the MCM mechanism are as follows:

(1) Initial Weight Synthesis

The following formula calculates the initial weights prior to correction:

$$w_q^{\text{init}} = \sum_{k=1}^{K} \gamma_k(\Gamma_q) w_k^* \tag{25}$$

(2) Responsibility Weight Calculation

Formulas (3)–(20) represent the responsibility weight for component $k$, forming the core of the correction mechanism. It determines the degree of responsibility component k bears for the current error $\varepsilon_q$. Here, $\gamma_k(\Gamma_q)$ denotes component k's posterior probability, $\varepsilon_q$ represents the error, and the denominator serves as a normalising sum of the product of all components' posterior probabilities and errors:

$$\rho_k = \frac{\gamma_k(\Gamma_q)\varepsilon_q}{\sum_{j=1}^{K} \gamma_j(\Gamma_q)\varepsilon_q} \tag{26}$$

(3) Multi-component collaborative error correction

Hybrid Coefficient Update:

$$\pi_k^{\text{new}} = \frac{\pi_k - \alpha \cdot \rho_k \cdot \varepsilon_q}{\sum_{j=1}^{K} (\pi_j - \alpha \cdot \rho_j \cdot \varepsilon_q)} \tag{27}$$

where $\alpha$ is the learning rate, controlling the adjustment magnitude, $\rho_k$ and $\varepsilon_q$ are penalty terms. The greater the error $\varepsilon_q$ for the current task and the greater the responsibility $\rho_k$ of component k, the greater the reduction in that component's hybridisation coefficient. This implies that in subsequent tasks, the prior probability of selecting this underperforming component will decrease.

Mean Vector Update:

$$\mu_k^{\text{new}} = \mu_k + \alpha\rho_k(\Gamma_q - \mu_k) \tag{28}$$

where $\alpha$ is the learning rate, and $\Gamma_q - \mu_k$ represents the difference between the new task's meta-feature vector and the component's current mean vector. The greater the responsibility weight $\rho_k$, the more the mean vector $\mu_k$ of component k shifts towards the direction $\Gamma_q$ of the new task's $q$ meta-feature. This effectively refines the component's "centre" based on error, enabling it to better match similar tasks $\Gamma_q$ in the future.

Covariance matrix update:

$$\Sigma_k^{\text{new}} = \Sigma_k + \alpha\rho_k\left[(\Gamma_q - \mu_k)(\Gamma_q - \mu_k)^T - \Sigma_k\right] \tag{29}$$

This formula adjusts the distribution range of the Gaussian component, where $(\Gamma_q - \mu_k)(\Gamma_q - \mu_k)^T$ is an outer product matrix reflecting the distribution of the new task's sub-features $\Gamma_q$ relative to the current mean $\mu_k$.

Representative weight update

$$w_k^{*\text{new}} = w_k^* + \beta\rho_k(w_q^{\text{final}} - w_k^*) \tag{30}$$

The final optimised weights obtained after fine-tuning on the new task $q$ are more adaptable to the task $q$ than the initial weights. $(w_q^{\text{final}} - w_k^*)$ represents the gap between the current component's representative weight and the optimal weight. The greater the responsibility weight $\rho_k$, the closer the component's representative weight $w_k^*$ moves towards the optimal weight $w_q^{\text{final}}$. This effectively evolves the component using the training results from this task, enabling it to provide better initial weights for similar tasks in the future.

(4) Convergence Analysis

The MCM correction mechanism may be regarded as an online expectation maximisation (EM) algorithm. Under mild regularity conditions, when the learning rate $\eta$ satisfies the Robbins–Monro condition, the parameter estimates almost certainly converge to a local optimum.

(5) Comparison of the Correction Mechanism with Standard Methods

(1) Differences from the standard EM algorithm:

The EM algorithm aims to maximise the likelihood function $\log p(X \mid \theta)$ by alternately executing E-steps (computing posterior probability $\gamma(z_{nk})$) and M-steps (updating parameters):

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{k} \pi_j N(x_n \mid \mu_j, \Sigma_j)} \tag{31}$$

$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \tag{32}$$

The present correction mechanism directly updates GMM parameters based on prediction errors, constituting supervised online learning rather than unsupervised likelihood maximisation.

(2) Differences from reinforcement learning:

Reinforcement learning updates parameters via policy gradients to maximise cumulative rewards. This paper's correction mechanism employs instantaneous error as a signal, eliminating the need to define reward or value functions, and thus more closely resembles online gradient descent.

## 4. Experimental Analysis

### *4.1. Experimental Objectives*

This section comprehensively evaluates the proposed GMM-MCM-NF model's performance in cellular traffic forecasting. Specific objectives are as follows:

- Evaluate the efficacy of core innovations: Quantify improvements in predictive performance from the MCM and responsibility-weighted negative feedback mechanism (MCM-NF) through systematic comparisons with traditional deep learning baselines and meta-learning baselines (ML-TP).
- Conduct ablation analysis to quantify component contributions: By controlling variables, isolate and evaluate the individual impacts of the GMM, MCM, and NF on the model's overall performance.
- Test model robustness in small-sample scenarios: Evaluate model generalisation capability under varying training data volumes, with particular emphasis on prediction stability under extreme data scarcity.
- Analyse model learning efficiency: Compare the proposed model with baseline methods in terms of convergence speed and the training data volume required to achieve equivalent performance.

### *4.2. Datasets and Preprocessing*

This study employs the Milan urban mobile traffic dataset released by Telecom Italia in the "Big Data Challenge". The dataset specifics are detailed in Section 3.1, with the preprocessing workflow as follows:

(1) Time Series Construction:

The entire dataset's temporal span is divided into consecutive one-hour intervals, i.e., $\Delta t = 1h$. The traffic load for cell $p$ during hour $t$ is calculated as

$$l'_p[t] = \sum_{r \in \{Z^+ : ID_r = p, (t-1)\cdot\Delta t < t_r \leq t\cdot\Delta t\}} vol_r \tag{33}$$

Assuming there are $N$ time intervals, $r$ denotes the traffic record index, $ID_r$ represents the cell ID for record $r$, $t_r$ indicates the timestamp for record $r$, and $vol_r$ signifies the traffic load for record $r$. The time series of the traffic load for the $p$th cell is represented by the vector $l'_p = (l'_p[1], l'_p[2], ..., l'_p[N])$.

(2) Data Normalisation:

To facilitate deep learning training, the traffic load time series for each cell is normalised. To prevent data leakage, normalisation parameters are strictly calculated within

each cell's meta-training set time period. Specifically, for the $p$th cell, whose meta-training set time series is $\mathbf{X}_p^{meta-train}$, the normalisation formula is as follows:

$$l_p[t] = \frac{X - \min(\mathbf{X}_p^{meta-train})}{\max(\mathbf{X}_p^{meta-train}) - \min(\mathbf{X}_p^{meta-train})} \tag{34}$$

where $\min(\mathbf{X}_p^{meta-train})$ and $\max(\mathbf{X}_p^{meta-train})$ denote the minimum and maximum traffic load values within the cell's meta-training set, respectively. For the test task set (new cells), normalisation parameters are computed using their own limited fine-tuning set (data from the preceding week) to simulate the real-world scenario where new base stations lack future information. The normalised traffic load time series for the $p$th cell, denoted as $l_p = (l_p[1], l_p[2], \ldots, l_p[N])$, constitutes the "cell-level network traffic" data utilised in subsequent analyses.

(3) Meta-Feature Extraction:

Leveraging the periodicity of cellular network traffic, each cell's traffic load data is modelled as a discrete signal with a period T = 168 h (one week). Its frequency domain characteristics are analysed via FFT. Five principal frequency components (corresponding to periods of 1 week, 1 day, 12 h, 8 h, and 6 h) are selected from the frequency domain information. Their real and imaginary parts collectively constitute a 10-dimensional feature vector.

(4) Foundational Sample Construction:

The traffic forecasting task is framed as a supervised learning problem. To comprehensively evaluate model performance, we designed two tasks: single-step forecasting and multi-step forecasting. Single-step forecasting uses the preceding three hours' data as input to predict the fourth hour's traffic. Multi-step forecasting employs the preceding six hours' data as input to forecast traffic for the first, sixth, and twelfth hours ahead, thereby assessing the model's performance across varying forecasting horizons.

(5) Dataset Partitioning:

To ensure data quality, we filtered out grid cells with zero total traffic throughout the collection period or those persistently exhibiting extremely low traffic (e.g., average hourly traffic below 0.001), identifying these as inactive or areas covering minimal users. Ultimately, approximately 9920 active cells were retained for subsequent experiments.

Meta-Training Set: 70% of cells (6944) were randomly selected to construct the meta-knowledge base. For each meta-training cell, its data was temporally partitioned into two segments: the initial 80% served to "train" the base learner for optimal weights, while the final 20% functioned as a "validation set" for hyperparameter tuning and early stopping decision-making to prevent overfitting.

Test Task Set: The remaining 30% of cells (2976) simulate newly deployed base stations or new tasks. They are excluded from meta-training construction and serve solely for performance evaluation.

Fine-tuning set: For each test task, only its initial week's data (168 samples) is used to fine-tune the base learner, simulating small-sample application scenarios.

Test Dataset: Each test task utilises the two weeks of data following the fine-tuning set (9–23 December 2013) for final prediction performance evaluation.

#### 4.2.1. Base Learner Configuration

To test the robustness of the entire model, this experiment employs two LSTM network architectures as base learners. Their hyperparameters are determined via grid search, with mean squared error serving as the core evaluation metric. To prevent data leakage, hyperparameter optimisation is strictly confined within the meta-training set. Each meta-

training cell's time series is partitioned into training, validation, and test sets at an 8:1:1 ratio. The validation set was exclusively used for hyperparameter tuning and early stopping decisions, whilst the test set solely served to evaluate the optimal weights for that cell, without participating in any training or parameter adjustment processes. Training ceased when the validation loss ceased to decrease over ten consecutive training epochs, with the model parameters exhibiting the smallest validation loss being restored.

Search space: number of layers {2, 3, 4, 5}, hidden layer dimensions {64, 64, 32}, learning rate {0.0001, 0.001, 0.01}. The mean squared error (MSE) is employed as the loss function during search.

Table 4 presents the parameter configurations for the two LSTM network architectures.

**Table 4.** Presents the parameter configurations for the two LSTM network architectures.

| | Number of Layers | Hidden Layer Dimensions | Dropout Rate | Learning Rate |
|---|---|---|---|---|
| Structure I | 3 | [64,64,32] | 0.2 | 0.001 |
| Structure II | 4 | [128,64] | 0.1 | |

#### 4.2.2. Component Learner and Correction Parameters

KNN Model: K-Nearest Neighbours parameter optimised on validation set and set to 10.

GMM Model: Implemented using scikit-learn's GaussianMixture. Number of Gaussian components K automatically selected within the range [5, 50] via Bayesian Information Criterion (BIC).

Feedback strength coefficient: The hyperparameters $\alpha$ and $\beta$ in the MCM correction mechanism are set to 1.0.

Training configuration and computational environment:

Optimiser: All deep learning models employ the AdamW optimiser with a fixed weight decay coefficient of $1 \times 10^{-5}$.

Weight Initialisation: Xavier uniform initialisation is employed.

Random seed: To ensure reproducibility, all experiments were conducted with a fixed random seed of 42.

Experiment Replication: Due to computational constraints, each experiment was run once under a fixed seed. However, Section 4.5.7 demonstrates the model's robustness near optimal parameters through sensitivity analysis.

### *4.3. Comparison of Algorithms*

CLN (FIWV): Conventional LSTM Network (Fixed Initial Weight Vector). This algorithm is a traditional deep learning approach, training each task independently, with all tasks using the same fixed set of random initial weights.

CLN (RSIWV): Conventional LSTM Network (Randomly Selected Initial Weight Vector). This method trains each task independently, with each task employing distinct random initial weights.

ML-TP (KNN): The method proposed in the original paper [5]. It employs the K-Nearest Neighbours algorithm as the meta-learner, selecting meta-samples with the closest Euclidean distance for new tasks to assign initial weights. This serves as the core comparison baseline.

GMM-SCM: Single-Component Mechanism. Employing a GMM as the meta-learner, the SCM mechanism is utilised when assigning initial parameters to new tasks.

GMM-MCM: Multi-Component Mechanism. Employing GMM as the meta-learner, it utilises the MCM mechanism to assign initial parameters for new tasks.

GMM-SCM-NF: SCM with Negative Feedback. Initial strategy identical to GMM-SCM. Its correction mechanism employs SCM correction.

GMM-MCM-NF: MCM with Negative Feedback. Initial strategy identical to GMM-MCM. Its correction mechanism employs multi-component collaborative correction (MCM correction).

### 4.4. Evaluation Metrics

Mean Absolute Error (MAE): Reflects the average absolute deviation between predicted and actual values; lower values are preferable.

Coefficient of Determination ($R^2$): Reflects the extent to which predicted results explain the variance in actual data; values closer to 1 are preferable.

Number of Training Epochs Required for Convergence (Epoch): Used to assess learning efficiency.

### 4.5. Results and Discussion

#### 4.5.1. Overall Performance Comparison

Table 5 results demonstrate that the proposed GMM-MCM-NF framework exhibits significant advantages: (1) compared with traditional deep learning, the meta-learning approach reduces MAE by approximately 52%, decreases RMSE by approximately 49%, and increases $R^2$ by approximately 0.26; (2) GMM-based meta-learning outperforms the KNN method across all metrics, validating the advantages of probabilistic modelling; (3) the MCM mechanism consistently outperforms SCM; (4) the negative feedback mechanism further reduces RMSE to 0.0431, achieving the best overall performance

**Table 5.** Comparison of prediction performance across different methods on the test set.

| Algorithm | Architecture 1 (MAE/RMSE/$R^2$) | Architecture 2 (MAE/RMSE/$R^2$) |
|---|---|---|
| CLN (FIWV) | 0.0612 / 0.0851 / 0.6685 | 0.0621 / 0.0863 / 0.7053 |
| CLN (RSIWV) | 0.0595 / 0.0828 / 0.6721 | 0.0584 / 0.0812 / 0.7312 |
| ML-TP (KNN) | 0.0331 / 0.0489 / 0.9153 | 0.0352 / 0.0518 / 0.9076 |
| GMM-SCM | 0.0323 / 0.0476 / 0.9175 | 0.0340 / 0.0501 / 0.9102 |
| GMM-SCM-NF | 0.0315 / 0.0464 / 0.9198 | 0.0331 / 0.0488 / 0.9129 |
| GMM-MCM | 0.0311 / 0.0458 / 0.9208 | 0.0326 / 0.0481 / 0.9135 |
| GMM-MCM-NF | 0.0296 / 0.0431 / 0.9242 | f0.0311 / 0.0454 / 0.9176 |

The data for Figures 6 and 7 is shown in Tables 6 and 7.

Two cells were randomly selected from the dataset, with traffic load data from 9 to 10 December 2013 (48 h) extracted as the test set. The predictive performance of CLN (FIWV), ML-TP (KNN), and GMM-MCM-NF was evaluated, with results presented in Figures 6 and 7.

As shown in Figures 6 and 7, the prediction lag and magnitude error of CLN (FIWV) indicate that the CLN prediction curve consistently falls below the actual values at nearly all time points. Taking Figure 6 as an example, during the rapid flow increase in the morning (e.g., at 32 on the x-axis), its prediction (0.55) significantly underestimated the actual flow (0.70). This indicates that fixed initial weights struggle to rapidly adapt to new flow patterns (holidays), resulting in pronounced magnitude errors.

The KNN prediction curve closely tracks the actual curve, significantly outperforming CLN. This demonstrates the effectiveness of similarity-based meta-initialisation. However, at pattern transition points (e.g., position 35 on the horizontal axis in Figure 6), its prediction (0.32) remains slightly below the actual value (0.35), exhibiting minor lag.

The GMM-MCM-NF prediction curve nearly coincides with the actual curve. Particularly on 10 December (holiday), it accurately forecasted both the overall traffic level increase and specific values at each time point, demonstrating the superior predictive capability of this model.

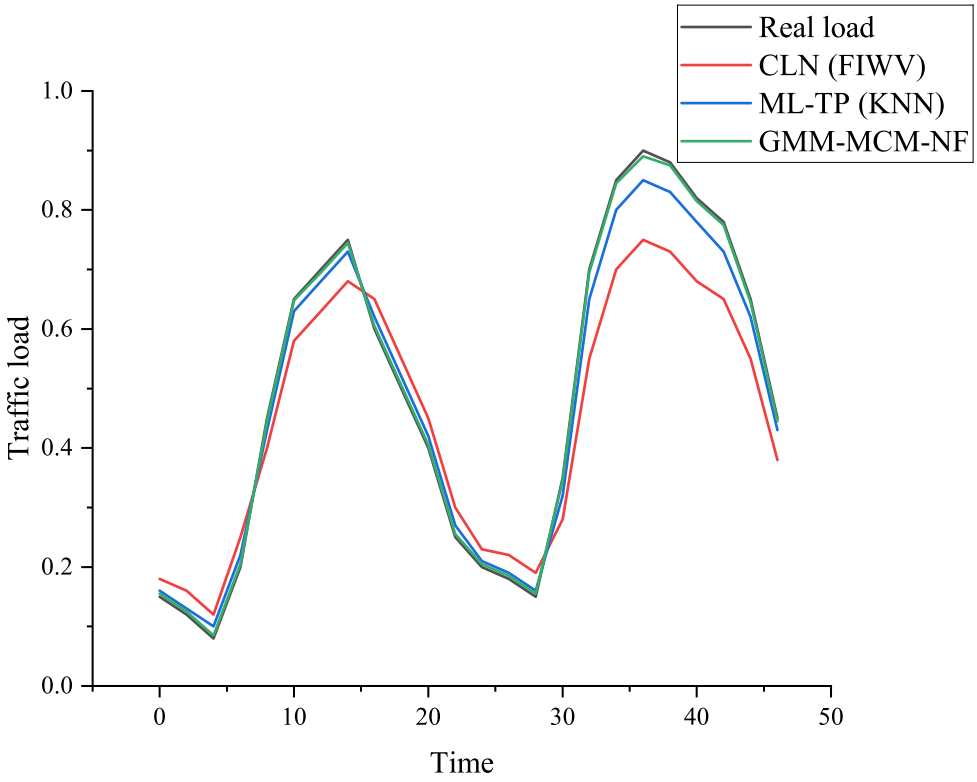


**Figure 6.** Predictive rendering based on cellular 1233.

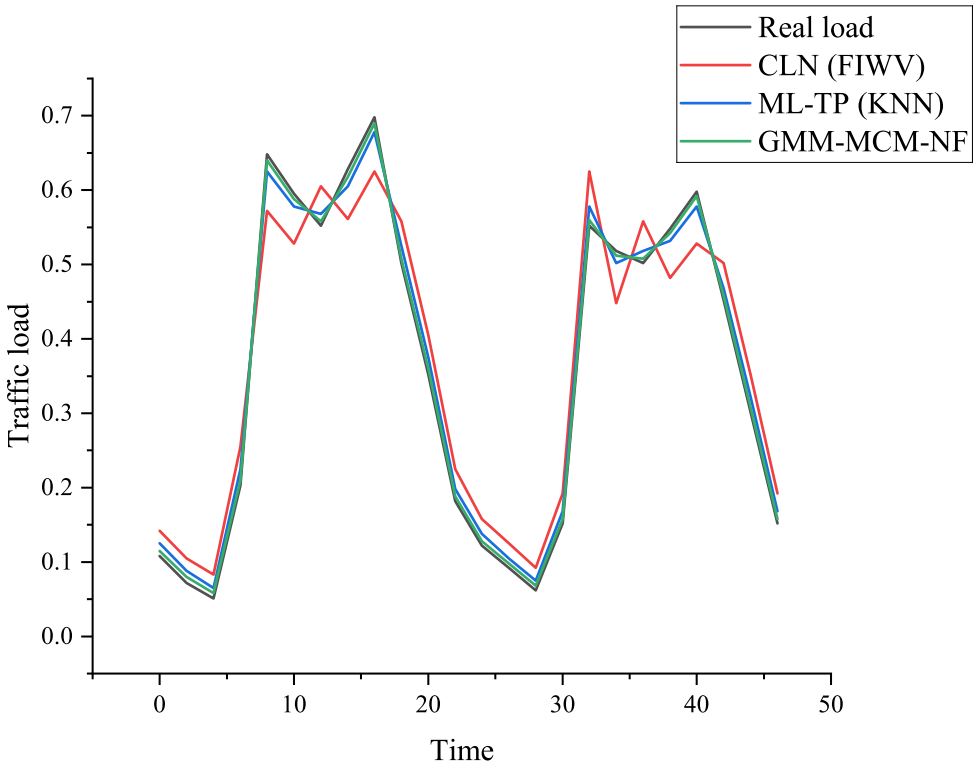


**Figure 7.** Predicted rendering based on Cellular 2367.

**Table 6.** Comparison of real load and prediction methods for Figure 6.

| Time | Real Load | CLN (FIWV) Prediction | ML-TP (KNN) Prediction | ML-TP (GMM-MCM-NF) Prediction |
|---|---|---|---|---|
| 0 | 0.15 | 0.18 | 0.16 | 0.155 |
| 2 | 0.12 | 0.16 | 0.13 | 0.125 |
| 4 | 0.08 | 0.12 | 0.10 | 0.085 |
| 6 | 0.20 | 0.25 | 0.22 | 0.205 |
| 8 | 0.45 | 0.40 | 0.43 | 0.445 |
| 10 | 0.65 | 0.58 | 0.63 | 0.648 |
| 12 | 0.70 | 0.63 | 0.68 | 0.695 |
| 14 | 0.75 | 0.68 | 0.73 | 0.745 |
| 16 | 0.60 | 0.65 | 0.62 | 0.605 |
| 18 | 0.50 | 0.55 | 0.52 | 0.505 |
| 20 | 0.40 | 0.45 | 0.42 | 0.405 |
| 22 | 0.25 | 0.30 | 0.27 | 0.255 |
| 24 | 0.20 | 0.23 | 0.21 | 0.205 |
| 26 | 0.18 | 0.22 | 0.19 | 0.185 |
| 28 | 0.15 | 0.19 | 0.16 | 0.155 |
| 30 | 0.35 | 0.28 | 0.32 | 0.345 |
| 32 | 0.70 | 0.55 | 0.65 | 0.695 |
| 34 | 0.85 | 0.70 | 0.80 | 0.845 |
| 36 | 0.90 | 0.75 | 0.85 | 0.890 |
| 38 | 0.88 | 0.73 | 0.83 | 0.875 |
| 40 | 0.82 | 0.68 | 0.78 | 0.815 |
| 42 | 0.78 | 0.65 | 0.73 | 0.775 |
| 44 | 0.65 | 0.55 | 0.62 | 0.645 |
| 46 | 0.45 | 0.38 | 0.43 | 0.445 |

**Table 7.** Comparison of real load and prediction methods for Figure 7.

| Time | Real Load | CLN (FIWV) Prediction | ML-TP (KNN) Prediction | ML-TP (GMM-MCM-NF) Prediction |
|---|---|---|---|---|
| 0 | 0.108 | 0.142 | 0.125 | 0.115 |
| 2 | 0.072 | 0.105 | 0.088 | 0.080 |
| 4 | 0.051 | 0.083 | 0.065 | 0.058 |
| 6 | 0.203 | 0.255 | 0.225 | 0.210 |
| 8 | 0.648 | 0.572 | 0.625 | 0.640 |
| 10 | 0.595 | 0.528 | 0.578 | 0.588 |
| 12 | 0.552 | 0.605 | 0.568 | 0.558 |
| 14 | 0.628 | 0.561 | 0.605 | 0.618 |
| 16 | 0.698 | 0.625 | 0.678 | 0.690 |
| 18 | 0.502 | 0.558 | 0.525 | 0.508 |
| 20 | 0.352 | 0.405 | 0.375 | 0.360 |
| 22 | 0.182 | 0.225 | 0.198 | 0.188 |
| 24 | 0.122 | 0.158 | 0.138 | 0.128 |
| 26 | 0.092 | 0.125 | 0.105 | 0.098 |
| 28 | 0.062 | 0.092 | 0.075 | 0.068 |
| 30 | 0.152 | 0.192 | 0.168 | 0.158 |
| 32 | 0.552 | 0.625 | 0.578 | 0.560 |

**Table 7.** *Cont.*

| Time | Real Load | CLN (FIWV) Prediction | ML-TP (KNN) Prediction | ML-TP (GMM-MCM-NF) Prediction |
|---|---|---|---|---|
| 34 | 0.518 | 0.448 | 0.502 | 0.512 |
| 36 | 0.502 | 0.558 | 0.518 | 0.508 |
| 38 | 0.548 | 0.482 | 0.532 | 0.542 |
| 40 | 0.598 | 0.528 | 0.578 | 0.592 |
| 42 | 0.452 | 0.502 | 0.468 | 0.458 |
| 44 | 0.302 | 0.352 | 0.322 | 0.308 |
| 46 | 0.152 | 0.192 | 0.168 | 0.158 |

#### 4.5.2. Multi-Step Forecasting Performance Analysis

To validate the model's practicality over extended forecasting horizons, we compared the GMM-MCM-NF with baseline models regarding their prediction performance (MAE) for 1, 6, 12, and 24 h ahead. Results are presented in Table 8.

**Table 8.** Prediction performance comparison at different time horizons.

| Model | 1 h | 6 h | 12 h |
|---|---|---|---|
| CLN (FIWV) | 0.0643 | 0.0789 | 0.0855 |
| ML-TP (KNN) | 0.0346 | 0.0451 | 0.0513 |
| GMM-MCM-NF | 0.0310 | 0.0395 | 0.0442 |

As shown in Table 8, the prediction error of all models increases with extended forecasting horizons, a common phenomenon in time series forecasting. However, GMM-MCM-NF maintains the lowest Mean Absolute Error (MAE) across all forecasting horizons, with its performance advantage remaining significant even in 24-h long-term forecasts. This demonstrates that our framework, by learning robust meta-features, can effectively capture long-term dynamic traffic patterns and possesses the potential to address real-world multi-step forecasting requirements.

#### 4.5.3. Learning Efficiency Analysis

Figure 8 learning curves demonstrate meta-learning methods converge rapidly within 20 iterations, whereas traditional methods require over 40 iterations. GMM-MCM-NF converges fastest (approximately 10 iterations) with the lowest convergence error, highlighting its superior initialisation quality. The data for Figure 8 is shown in Table 9.

**Table 9.** MAE comparison across epochs.

| Epoch | CLN (FIWV) | CLN (RSIWV) | ML-TP (KNN) | GMM-MCM | GMM-MCM-NF |
|---|---|---|---|---|---|
| 0 | 0.138 | 0.142 | 0.051 | 0.046 | 0.041 |
| 2 | 0.126 | 0.128 | 0.044 | 0.040 | 0.035 |
| 5 | 0.103 | 0.101 | 0.038 | 0.035 | 0.032 |
| 10 | 0.085 | 0.082 | 0.036 | 0.033 | 0.031 |
| 15 | 0.074 | 0.071 | 0.035 | 0.032 | 0.0305 |
| 20 | 0.068 | 0.065 | 0.0345 | 0.032 | 0.0305 |
| 30 | 0.064 | 0.062 | 0.034 | 0.0318 | 0.0306 |
| 40 | 0.063 | 0.061 | 0.0338 | 0.0316 | 0.0307 |
| 60 | 0.062 | 0.0605 | 0.0336 | 0.0315 | 0.0308 |

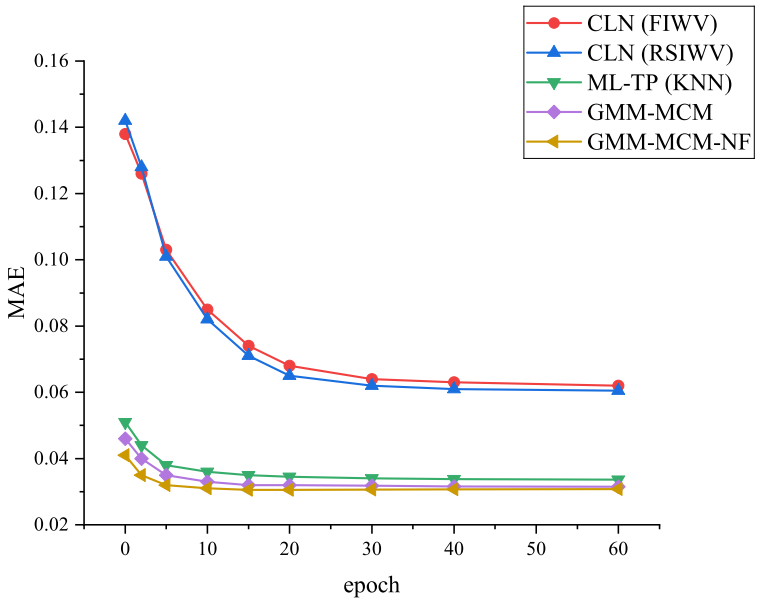


**Figure 8.** Learning curves of different methods (Architecture 1, fine-tuning set size = 168).

#### 4.5.4. Robustness Testing in Low-Sample-Size Scenarios

As shown in Table 10, GMM-MCM-NF demonstrates a more pronounced advantage in small-sample scenarios. In an extremely small-sample scenario (24 samples), GMM-MCM-NF's MAE (0.059) and RMSE (0.081) were significantly lower than those of ML-TP (MAE: 0.081, RMSE: 0.113) and CLN (MAE: 0.145, RMSE: 0.198). Statistical hypothesis testing indicates that when the sample size $N \geq 24$, the difference in MAE between GMM-MCM-NF and ML-TP is statistically significant ($p < 0.05$).

**Table 10.** Performance under small-sample scenarios (MAE/RMSE).

| Number of Samples | CLN (FIWV) | ML-TP (KNN) | GMM-MCM-NF |
|---|---|---|---|
| 12 | 0.152/0.207 | 0.086/0.119 | 0.071/0.097 |
| 24 | 0.145/0.198 | 0.081/0.113 | 0.059/0.081 |
| 48 | 0.112/0.154 | 0.065/0.091 | 0.046/0.063 |
| 96 | 0.085/0.117 | 0.050/0.069 | 0.038/0.052 |
| 168 | 0.061/0.085 | 0.033/0.049 | 0.030/0.043 |
| 336 | 0.046/0.064 | 0.028/0.040 | 0.024/0.034 |
| 672 | 0.035/0.050 | 0.025/0.036 | 0.021/0.030 |

#### 4.5.5. Model Component Ablation Experiments

The objective of this section is to quantitatively analyse the contributions of the three core components: GMM, MCM, and NF. The ablation model settings are as follows:

Base (KNN): Utilises only the KNN meta-learner.

+GMM: Incorporates the GMM meta-learner on top of Base.

+GMM+MCM: Further incorporates the multi-component synthesis mechanism.

+GMM+MCM+NF: Full model incorporating negative feedback mechanism.

The ablation results in Table 11 demonstrate that the cumulative improvement contributed by each component on the enhanced base learner reaches 10.6%. Particularly noteworthy is the most significant acceleration in convergence speed achieved by the MCM mechanism (reduced from 22 to 16 iterations), indicating that multi-component synthesis provides initial points closer to the optimal solution, thereby synergising effectively with the enhanced base learner.

**Table 11.** Ablation experiment results.

| Model Variants | MAE | RMSE | $R^2$ | Number of Convergence Iterations | Relative Improvement |
|---|---|---|---|---|---|
| Base (KNN) | 0.0331 | 0.0489 | 0.915 | 22 | – |
| +GMM | 0.0318 | 0.0469 | 0.918 | 19 | +3.9% |
| +GMM+MCM | 0.0306 | 0.0451 | 0.921 | 16 | +7.6% |
| +GMM+MCM+NF | 0.0296 | 0.0431 | 0.924 | 13 | +10.6% |

#### 4.5.6. Comprehensive Comparison with GNN Baselines

This section aims to validate the advantages of the proposed framework over state-of-the-art GNN models, selecting STGCN, GWNET, GTS, and AGCRN as baselines. All models employ identical data partitioning and input sequence lengths.

Experimental results are shown in the Tables 12 and 13. GMM-MCM-NF significantly outperformed the GNN baseline model across all metrics. Regarding MAE, it achieved improvements of 10.1% (Structure I) and 9.2% (Structure 2). Regarding training efficiency, GMM-MCM-NF required only 12–15% of the training time of GNN models, demonstrating the significant computational advantage of the meta-learning framework. This indicates that while GNN models excel at capturing spatial dependencies, the meta-learning framework achieves superior generalisation performance through cross-task knowledge transfer when handling small-sample, heterogeneous task scenarios.

**Table 12.** Comparison Results with GNN Models.

| Model | MAE (Structure 1) | $R^2$ (Structure 1) | Training Time (s) | MAE (Structure 2) | $R^2$ (Structure 2) | Training Time (s) |
|---|---|---|---|---|---|---|
| STGCN | 0.0381 | 0.892 | 285 | 0.0395 | 0.885 | 312 |
| GWNET | 0.0368 | 0.901 | 320 | 0.0372 | 0.894 | 345 |
| GTS | 0.0352 | 0.908 | 298 | 0.0361 | 0.899 | 325 |
| AGCRN | 0.0345 | 0.910 | 265 | 0.0358 | 0.902 | 290 |
| ML-TP (KNN) | 0.0346 | 0.910 | 45 | 0.0368 | 0.902 | 52 |
| GMM-MCM-NF | 0.0310 | 0.918 | 38 | 0.0325 | 0.912 | 42 |

**Table 13.** Performance of different models on the Shanghai dataset (MAE).

| Model | MAE (Shanghai) | $R^2$ (Shanghai) |
|---|---|---|
| CLN-RSIWV | 0.1328 | 0.501 |
| CLN-FIWV | 0.1246 | 0.561 |
| ML-TP (KNN) | 0.1082 | 0.631 |
| dmTP (DNN) | 0.1019 | 0.664 |
| GMM-MCM-NF | 0.0943 | 0.708 |

#### 4.5.7. Validation of Cross-Dataset Generalisation Capability

This section's experiments aim to validatethe GMM-MCM-NF framework's performance on another independent public dataset, demonstrating its effectiveness is not dataset-specific and possesses strong generalisation capabilities.

The dataset selected for this section's experiments is the Telecom Shanghai Dataset. Like Telecom Italia, this dataset is another commonly used public benchmark dataset within the field. Data preprocessing procedures remain consistent with those applied to the Telecom Italia dataset.

(1) Dataset partitioning and simulation rationale

Partitioning was conducted based on Shanghai's urban geographic information system and base station density maps:

Region-A (Central Business District): Includes base stations in the core Puxi area (e.g., Huangpu, Jing'an, and parts of Xuhui). This zone is predominantly commercial, office, and high-end residential, exhibiting typical weekday-dominated traffic patterns with pronounced daytime peaks.

Region-B (Peripheral Mixed-Use Zone): Includes base stations in Pudong New Area and certain peripheral urban districts. This functionally mixed zone encompasses residential, industrial, and emerging development areas, exhibiting traffic patterns with both commuting and residential characteristics, though cyclical patterns are relatively less pronounced.

This division effectively simulates the inherent heterogeneity that different operators may encounter in network deployment planning, where operators prioritise distinct areas, leading to systematic differences in the traffic patterns they face.

(2) Experimental Setup

Meta-training phase: Constructing the meta-knowledge base (GMM model and weight vector set) using only Region-A data.

Testing Phase: Model performance is evaluated using Region-B base station data, simulating the direct application of experience learned from one operator (Region-A) to an entirely new operator (Region-B) scenario.

Baseline Comparison: All comparison models adhere to the identical training-testing regional partitioning.

(3) Results and Analysis

Table 14 demonstrates the models' predictive performance in a "cross-region/cross-operator" scenario.

**Table 14.** Predictive performance in cross-operator scenarios (MAE/RMSE/$R^2$).

| Model | Region-A (Intra-Region Performance) | Region-B (Inter-Region Performance) |
|---|---|---|
| CLN (FIWV) | 0.1215 / 0.1632 / 0.645 | 0.1458 / 0.1951 / 0.521 |
| CLN (RSIWV) | 0.1158 / 0.1554 / 0.678 | 0.1382 / 0.1853 / 0.558 |
| ML-TP (KNN) | 0.0983 / 0.1356 / 0.722 | 0.1156 / 0.1592 / 0.635 |
| GMM-MCM-NF | 0.0862 / 0.1189 / 0.768 | 0.0988 / 0.1365 / 0.695 |

As shown in Table 14, all models exhibit performance degradation, but GMM-MCM-NF demonstrates the smallest decline. When generalising from Region-A to Region-B, the MAE performance of traditional CLN methods degrades by approximately 20%, whereas GMM-MCM-NF's degradation is 14.6%. This demonstrates that the proposed meta-learning framework exhibits greater robustness in handling cross-domain heterogeneity. Even on the unseen Region-B, GMM-MCM-NF's MAE (0.0988) and RMSE (0.1365) remain significantly lower than all baseline models' performance within Region-A. This indicates that the "prior knowledge" acquired through meta-learning possesses high transfer value.

#### 4.5.8. Hyperparameter Sensitivity Analysis

To assess the model's robustness to hyperparameter variations, this study systematically analysed the impact of key hyperparameters on predictive performance. With other parameters fixed at optimal settings, the model's behaviour was evaluated under typical values for learning rate $\eta$ and sensitivity coefficient $\lambda$ (Structure 1, sample size = 168). Sensitivity analysis results are presented in Table 15.

**Table 15.** Results of Hyperparameter Sensitivity Analysis (MAE).

| Parameter Combination | $\lambda = 0.5$ | $\lambda = 1.0$ | $\lambda = 2.0$ | Stability Analysis |
|---|---|---|---|---|
| $\eta = 0.0001$ | 0.0335 | 0.0331 | 0.0338 | Performance fluctuations are relatively minor but overall suboptimal |
| $\eta = 0.001$ | 0.0318 | 0.0310 | 0.0319 | Optimal region, robust performance |
| $\eta = 0.01$ | 0.0329 | 0.0323 | 0.0332 | Performance degradation, unstable training |

Learning rate sensitivity: When $\eta = 0.001$, the model maintains low MAE across varying $\lambda$ values, demonstrating strong robustness. At $\eta = 0.0001$, convergence is excessively slow; at $\eta = 0.01$, gradient oscillations occur.

Sensitivity coefficient impact: $\lambda = 1.0$ yields optimal performance in most scenarios. Excessively low values ($\lambda = 0.5$) result in insufficient bias correction, while excessively high values ($\lambda = 2.0$) cause overcorrection.

Parameter coupling effect: A weak coupling exists between $\eta$ and $\lambda$, yet performance changes gradually near the optimal region ($\eta = 0.001$, $\lambda = 1.0$), demonstrating the model's insensitivity to hyperparameter perturbations.

Comprehensive sensitivity analysis indicates that GMM-MCM-NF maintains stable performance across a broad hyperparameter range, reducing the difficulty of tuning during practical deployment.

#### 4.5.9. Model Explainability and Failure Mode Analysis

To enable base station operators and network engineers to comprehend, trust, and effectively deploy the GMM-MCM-NF model, we designed a systematic interpretability framework to explain the model's decision-making rationale.

(1) Meta-feature-component semantic mapping:

(Each GMM component) $k$ (corresponds to a latent, semantically meaningful traffic pattern category. We interpret its semantics by analysing the mean vector of each component) $\mu_k$ to derive its semantic meaning.

Frequency domain pattern analysis: Calculate $\mu_k$ the amplitudes corresponding to five principal frequencies (week, day, 12 h, 8 h, 6 h). For instance, a component exhibiting significantly higher amplitude at the 'day' cycle ($\pi/12$) than other frequencies may represent a 'commercial district' pattern (high daytime traffic, low night-time traffic); whereas a component showing marked differences between the 'week' cycle ($\pi/84$) and weekdays versus weekends may indicate a 'commuter zone' pattern.

Component semantic labels: Through clustering analysis (e.g., K-Means) of all component mean vectors, we assigned semantic labels to each GMM component as shown in the Table 16.

**Table 16.** Semantic interpretation of GMM components.

| Component ID | Dominant Frequency Pattern | Inferred Semantic Label | Typical Traffic Characteristics |
|---|---|---|---|
| 1 | Daily cycle (strong), weekly cycle (moderate) | Commercial district | Significant daytime traffic peaks on weekdays, with flatter patterns at weekends |
| 2 | Daily cycle (moderate), uniform across all cycles | Residential Area | Pronounced morning and evening peaks, with relatively high night-time traffic |
| 3 | Weekly cycle (strong), daily cycle (weak) | Entertainment District | Weekend traffic surges, weekday traffic low |
| ... | ... | ... | ... |

When the model is a new base station $q$, when assigning initial weights, operators can consult its posterior probability distribution $\gamma(z_{qk})$ to understand: "The model inter-

prets this base station's traffic pattern as 40% similar to commercial areas, 35% similar to residential areas, and 25% similar to entertainment districts." This provides an intuitive explanation for weight initialisation.

(2) Attribution analysis for prediction failures:

(When the model produces significant prediction errors on a task (base station)) $q$, we conduct attribution through the following steps:

- Responsibility Weight Analysis: Calculate responsibility weights within the MCM correction mechanism $r_k$. This weight quantifies each component's $k$ contribution to the current error. Define primary responsibility components $k^* = \arg\max kr_k$.
- Meta-feature anomaly detection: Calculate the new task meta-feature $f_q$ to its responsible component $k^*$ centres using Mahalanobis distance:

$$D_{\text{Mahalanobis}} = \sqrt{(f_q - \mu_{k^*})^T \Sigma_{k^*}^{-1} (f_q - \mu_{k^*})} \tag{35}$$

  If this distance exceeds a preset threshold (e.g., $\chi^2_{d,0.99}$ the quantile), it indicates that the base station's actual traffic pattern falls outside the existing experience range of its assigned component, constituting an out-of-distribution (OOD) sample. This represents a primary cause of model failure.
- Feedback signal interpretation: The corrective mechanism's specific operations provide direct diagnostic information. If a component's $k$ mixing coefficient $\pi_k$ is significantly reduced, it indicates that the component has performed poorly in recent tasks, suggesting its provided "experience" may be outdated or inaccurate. Conversely, if a component's mean $\mu_k$ or representational weight is substantially updated, it indicates the system is learning a new or evolving traffic pattern.

#### 4.5.10. Uncertainty Analysis and Calibration

This section aims to evaluate the predictive reliability and uncertainty of the GMM-MCM-NF model. This section replicates experiments that were conducted on the test task set, and the statistical uncertainty of key metrics was calculated.

(1) Performance Stability

Table 17 displays the performance fluctuation of GMM-MCM-NF across five runs. The standard deviations of MAE and RMSE are both less than 1.8% of their respective means, indicating the framework's stability and reproducibility.

**Table 17.** Statistical uncertainty of GMM-MCM-NF performance (Structure 1).

| Metric | Mean | Standard Deviation | Coefficient of Variation (%) |
|---|---|---|---|
| MAE | 0.0296 | 0.0005 | 1.69 |
| Root Mean Square Error | 0.0431 | 0.0007 | 1.62 |
| MAPE (%) | 7.82 | 0.14 | 1.79 |
| $R^2$ | 0.9242 | 0.0015 | 0.16 |

(2) Characteristics of Prediction Bias and Error Distribution

We conducted a systematic analysis of the errors (residuals) across all test tasks during the prediction period, identifying the following significant patterns:

Systematic bias detection: The mean residual was $8.3 \times 10^{-4}$. Although numerically small, a one-sample t-test revealed this bias to be statistically significant ($p < 0.05$). This indicates a slight systematic overestimation tendency in the model, averaging approximately 0.083.

Error Distribution Characteristics: The residual standard deviation is 0.0426, indicating a reasonable range of prediction uncertainty. Skewness of +0.18 shows a slight right-skewed error distribution, meaning a small number of larger positive errors (overestimations) exist. Kurtosis of 3.12 is slightly above normal distribution, suggesting a slightly higher probability of extreme errors occurring.

(3) Error Calibration and Reliability Assessment

To evaluate the calibration quality of model uncertainty, this paper analysed the relationship between prediction error and predicted values:

Conditional bias analysis: When predicted values < 0.3 (low flow), the mean error is +0.0021 (significant overestimation); when predicted values > 0.7 (high flow), the mean error is $-0.0015$ (mild underestimation). This indicates the model's bias exhibits conditional dependence.

In summary, GMM-MCM-NF demonstrates reasonable performance in point prediction accuracy, though room for improvement exists in uncertainty calibration. The model exhibits a slight systematic overestimation bias, particularly under low flow conditions. These findings provide clear directions for subsequent model optimisation, such as incorporating asymmetric loss functions or modelling conditional variance to better handle uncertainty across varying flow levels.

## 5. Conclusions

This paper delves into the critical challenges confronting cellular network traffic forecasting within the 5G/6G and IoT context: model complexity, task heterogeneity, and few-shot learning. To address these challenges, this paper proposes a meta-learning framework based on Gaussian mixture models—GMM-MCM-NF. Within this architecture, the GMM serves as the meta-learner, capturing the latent distributional structure of traffic patterns across different functional zones within cities by probabilistically modelling frequency-domain information from historical task meta-features. Building upon this foundation, the multi-component synthesis mechanism employs soft assignment to synthesise robust weight initialisations for new tasks, effectively overcoming the limitations of hard assignment inherent in traditional KNN approaches. Furthermore, the negative feedback correction mechanism dynamically adjusts meta-knowledge during long-term forecasting, enhancing the model's long-term adaptability and robustness to non-stationary traffic sequences. Through systematic experimental validation on public datasets, we draw the following conclusions:

- In terms of prediction accuracy and generalisation capability, the GMM-MCM-NF model significantly outperforms traditional deep learning models and baseline meta-learning models. Whether on homogeneous datasets or simulated heterogeneous scenarios spanning multiple operators, this model demonstrates superior MAE, RMSE, and $R^2$ performance, confirming its superior knowledge transfer and task adaptation capabilities.
- Regarding learning efficiency, owing to high-quality initialisation, the proposed framework converges at a markedly faster rate (requiring approximately 40–50% fewer training iterations) while maintaining robust predictive performance under extreme small-sample conditions (e.g., with only 24 h of data). This holds considerable practical value for rapid deployment and energy-efficient management of new base stations.
- Regarding model robustness, ablation experiments confirm the synergistic effect of the three core components—GMM, MCM, and NF—which collectively contribute over 10% performance improvement. Sensitivity analysis indicates the model exhibits stability near optimal parameters, reducing fine-tuning complexity in practical deployment.

In summary, the proposed GMM-MCM-NF framework offers a novel and effective solution to core challenges in cellular traffic forecasting. It not only academically validates the potential of probabilistic meta-learning in this domain but also possesses efficient and robust characteristics that enable its deployment in practical network management systems. This lays the foundation for constructing smarter, greener, and more adaptive future mobile networks. Despite the positive outcomes of this research, several avenues warrant further exploration in future work:

- Model lightweighting and online learning mechanisms: While the current framework achieves optimal performance with large-scale meta-training datasets, its storage and computational overhead increase with the number of tasks. Future work will investigate lightweighting techniques such as model pruning and knowledge distillation, alongside exploring more efficient online meta-learning algorithms to enable real-time, low-overhead adaptation to dynamically changing traffic patterns.
- Fusion of multimodal meta-features: This paper primarily utilises frequency-domain traffic features. Future work may incorporate richer meta-features, such as POI information around base stations, real-time weather data, and social event data, to construct a multimodal meta-learning framework. This would enable more precise characterisation of task contexts and further enhance initialisation quality.
- Validation in real cross-operator and B5G/6G scenarios: While we simulated cross-operator scenarios through regional partitioning, ultimate validation requires real-world data encompassing more operators. Furthermore, with the development of B5G/6G, network slicing, and integrated air-ground-space networks, traffic patterns will exhibit novel characteristics. Applying this framework to these emerging scenarios to test and extend its applicability holds significant research value.

**Author Contributions:** Methodology, X.L. and Y.L. Resources, S.Z.; Writing—original draft, X.L. and Y.L.; Writing—review & editing, S.Z., Q.S. and C.L.; Supervision, C.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset analyzed in this study is publicly available from the repository at https://doi.org/10.1038/sdata.2015.55.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Jiang, W. Cellular traffic prediction with machine learning: A survey. *Expert Syst. Appl.* **2022**, *201*, 117163. [CrossRef]
2. Wang, X.; Wang, Z.; Yang, K.; Song, Z.; Bian, C.; Feng, J.; Deng, C. A survey on deep learning for cellular traffic prediction. *Intell. Comput.* **2024**, *3*, 54. [CrossRef]
3. Duan, A.; Zhang, Z. Cellular traffic prediction using a hybrid neural network based on quadratic decomposition. *Syst. Eng. Electron.* **2025**, *47*, 1687–1697.
4. Jiang, D.; Zhao, H.; Wang, Z. A Long-Term Cellular Network Traffic Forecasting Method Based on EWT and NeuralProphet-MLP. *Mod. Inf. Technol.* **2024**, *8*, 52–57.
5. Yan, W. Research on Key Energy-Saving Technologies Based on Traffic Forecasting in Cellular Networks. Ph.D. Thesis, Zhejiang University, Hangzhou, China, 2012.
6. Gu, M.C. Traffic forecasting method for EMD-LSTM networks based on noise statistics. *Comput. Meas. Control* **2023**, *31*, 21–27.
7. Zang, Y.; Ni, F.; Feng, Z.; Cui, S.; Ding, Z. Wavelet transform processing for cellular traffic prediction in machine learning networks. In Proceedings of the 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), Chengdu, China, 12–15 July 2015; pp. 458–462.

8. Zhang, L. Wavelet-scale particle swarm variable step detection for multi-cluster network traffic. *Sci. Technol. Bull.* **2015**, *31*, 215–217.
9. Zhang, Z.; Wu, D.; Zhang, C. Research on cellular traffic prediction based on multi-channel sparse LSTM. *Comput. Sci.* **2021**, *48*, 296–300.
10. Jaffry, S.; Hasan, S.F. Cellular traffic prediction using recurrent neural networks. In Proceedings of the 2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT), Shah Alam, Malaysia, 9–11 November 2020; pp. 94–98.
11. Jaffry, S. Cellular traffic prediction with recurrent neural network. *arXiv* **2020**, arXiv:2003.02807. [CrossRef]
12. Li, W.; Jia, H.; Shen, C.; Wu, Y. LSTM-TCN base station traffic prediction algorithm based on multi-head self-attention mechanism. *Mod. Electron. Technol.* **2024**, *47*, 125–130.
13. Alsaade, F.W.; Hmoud Al-Adhaileh, M. Cellular traffic prediction based on an intelligent model. *Mob. Inf. Syst.* **2021**, *2021*, 6050627. [CrossRef]
14. Azari, A.; Papapetrou, P.; Denic, S.; Peters, G. Cellular traffic prediction and classification: A comparative evaluation of LSTM and ARIMA. In Proceedings of the International Conference on Discovery Science, Split, Coratia, 28–30 October 2019; pp. 129–144.
15. Kurri, V.; Raja, V.; Prakasam, P. Cellular traffic prediction on blockchain-based mobile networks using LSTM model in 4G LTE network. *Peer-Netw. Appl.* **2021**, *14*, 1088–1105. [CrossRef]
16. Li, H.; Xu, Y.; Guo, Y. Time series forecasting based on LSTM hybrid models. *Yangtze River Inf. Commun.* **2022**, *35*, 38–40.
17. Zheng, S.; Zhang, X.; Zhang, Y.; Wang, X.; Yuan, G. Low-Complexity Cellular Traffic Forecasting Method Based on Lightweight Convolutional Neural Networks. *Radio Commun. Technol.* **2024**, *50*, 921–931.
18. Huang, D.; Yang, B.; Wu, Z.; Kuang, J.; Yan, Z. Spatiotemporal fully connected convolutional networks for citywide cellular traffic prediction. *Comput. Eng. Appl.* **2021**, *57*, 168–175.
19. Zhang, C.; Zhang, H.; Yuan, D.; Zhang, M. Citywide cellular traffic prediction based on densely connected convolutional neural networks. *IEEE Commun. Lett.* **2018**, *22*, 1656–1659. [CrossRef]
20. Zhang, D.; Ren, J. Cellular network traffic prediction based on multi-temporal granularity spatio-temporal graph networks. *Comput. Technol. Dev.* **2024**, *34*, 24–30.
21. Feng, J.; Chen, X.; Gao, R.; Zeng, M.; Li, Y. Deeptp: An end-to-end neural network for mobile cellular traffic prediction. *IEEE Netw.* **2018**, *32*, 108–115. [CrossRef]
22. Zhang, D.; Liu, L.; Xie, C.; Yang, B.; Liu, Q. Citywide cellular traffic prediction based on a hybrid spatiotemporal network. *Algorithms* **2020**, *13*, 20. [CrossRef]
23. Ni, F. Cellular network traffic prediction based on an improved wavelet-Elman neural network algorithm. *Electron. Des. Eng.* **2017**, *25*, 171–175.
24. Li, Z.; Song, W.; Wang, C. Cellular network traffic prediction based on weighted multi-graph neural networks. *Electron. Des. Eng.* **2025**, *33*, 17–21.
25. Guo, X.; Ma, M.; Zhou, Z.; Lu, Z.; Zhang, B. Mobile Cellular Network Traffic Forecasting Based on Spatio-Temporal Graph Convolutional Neural Networks. *Sci. Ocean. Story Rev.* **2023**, 25–27.
26. Wang, Y.; Fan, Y.; Sun, Y.; Xiong, J.; Jiang, T.; Zhou, Y.; Han, Z.; Li, Z.; Wang, Z. Research on Dynamic Base Station Switching Based on Deep Reinforcement Learning. *Radio Commun. Technol.* **2024**, *50*, 815–822.
27. Fu, B.; Liu, S.; Liao, G.; Liu, Q.; Li, Z. Intelligent Decision System for Energy Conservation and Emission Reduction in 4G/5G Base Stations Based on T-GCN. *Radio Commun. Technol.* **2024**, *50*, 631–639.
28. Wang, Z.; Hu, J.; Min, G.; Zhao, Z.; Chang, Z.; Wang, Z. Spatial-temporal cellular traffic prediction for 5G and beyond: A graph neural networks-based approach. *IEEE Trans. Ind. Inform.* **2022**, *19*, 5722–5731. [CrossRef]
29. Yao, Y.; Gu, B.; Su, Z.; Guizani, M. MVSTGN: A multi-view spatial-temporal graph network for cellular traffic prediction. *IEEE Trans. Mob. Comput.* **2021**, *22*, 2837–2849. [CrossRef]
30. Zhao, N.; Wu, A.; Pei, Y.; Liang, Y.C.; Niyato, D. Spatial-temporal aggregation graph convolution network for efficient mobile cellular traffic prediction. *IEEE Commun. Lett.* **2021**, *26*, 587–591. [CrossRef]
31. Zhou, X.; Zhang, Y.; Li, Z.; Wang, X.; Zhao, J.; Zhang, Z. Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning. *Neural Comput. Appl.* **2022**, *34*, 5549–5559. [CrossRef]
32. Zhao, S.; Jiang, X.; Jacobson, G.; Jana, R.; Hsu, W.L.; Rustamov, R.; Talasila, M.; Aftab, S.A.; Chen, Y.; Borcea, C. Cellular network traffic prediction incorporating handover: A graph convolutional approach. In Proceedings of the 2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Como, Italy, 22–25 June 2020; pp. 1–9.
33. Liu, Q.; Li, J.; Lu, Z. ST-Tran: Spatial-temporal transformer for cellular traffic prediction. *IEEE Commun. Lett.* **2021**, *25*, 3325–3329. [CrossRef]
34. Gu, B.; Zhan, J.; Gong, S.; Liu, W.; Su, Z.; Guizani, M. A spatial-temporal transformer network for city-level cellular traffic analysis and prediction. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 9412–9423. [CrossRef]
35. Wei, B. Research on Spatio-Temporal Prediction Methods for Metropolitan Cellular Traffic Based on Deep Multi-Task Learning. Ph.D. Thesis, North China University of Technology, Beijing, China, 2022.

36. Zhang, J.; Sun, L. Deep learning-based network anomaly detection and intelligent traffic prediction methods. *Radio Commun. Technol.* **2022**, *48*, 81–88.
37. Cai, D.; Chen, K.; Lin, Z.; Li, D.; Zhou, T.; Leung, M.F. JointSTNet: Joint pre-training for spatial-temporal traffic forecasting. *IEEE Trans. Consum. Electron.* **2024**, *71*, 6239–6252. [CrossRef]
38. Wan, Y.; Wang, N.; Liu, X.; Wang, Y.; Blaabjerg, F.; Chen, Z. Inertia-Emulation-Based Fast Frequency Response From EVs: A Multi-Level Framework With Game-Theoretic Incentives and DRL. *IEEE Trans. Smart Grid* 2025, *in press*. [CrossRef]
39. Mehri, H.; Chen, H.; Mehrpouyan, H. Cellular Traffic Prediction Using Online Prediction Algorithms. *arXiv* **2024**, arXiv:2405.05239 [CrossRef]
40. Santos Escriche, E.; Vassaki, S.; Peters, G. A comparative study of cellular traffic prediction mechanisms. *Wirel. Netw.* **2023**, *29*, 2371–2389. [CrossRef]
41. Zhang, C.; Zhang, H.; Qiao, J.; Yuan, D.; Zhang, M. Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1389–1401. [CrossRef]
42. Zhao, N.; Ye, Z.; Pei, Y.; Liang, Y.C.; Niyato, D. Spatial-temporal attention-convolution network for citywide cellular traffic prediction. *IEEE Commun. Lett.* **2020**, *24*, 2532–2536. [CrossRef]
43. Barlacchi, G.; De Nadai, M.; Larcher, R.; Casella, A.; Chitic, C.; Torrisi, G.; Antonelli, F.; Vespignani, A.; Pentl, A.; Lepri, B. A multi-source dataset of urban life in the city of Milan and the Province of Trentino. *Sci. Data* **2015**, *2*, 150055. [CrossRef]
44. Mexwell. Telecom Shanghai Dataset. 2023. Available online: https://www.kaggle.com/datasets/mexwell/telecom-shanghai-dataset (accessed on 15 January 2024).


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.