future internet

MDPI

Article

Enhanced Multi-Task Traffic Forecasting in Beyond 5G Networks: Leveraging Transformer Technology and Multi-Source Data Fusion

Ibrahim Althamary , Rubbens Boisguene and Chih-Wei Huang *

Department of Communication Engineering, National Central University, Taoyuan City 320317, Taiwan; 106583603@cc.ncu.edu.tw (I.A.); rubbensonly@gmail.com (R.B.)
* Correspondence: cwhuang@ce.ncu.edu.tw

Abstract: Managing cellular networks in the Beyond 5G (B5G) era is a complex and challenging task requiring advanced deep learning approaches. Traditional models focusing on internet traffic (INT) analysis often fail to capture the rich temporal and spatial contexts essential for accurate INT predictions. Furthermore, these models do not account for the influence of external factors such as weather, news, and social trends. This study proposes a multi-source CNN-RNN (MSCR) model that leverages a rich dataset, including periodic, weather, news, and social data to address these limitations. This model enables the capture and fusion of diverse data sources for improved INT prediction accuracy. An advanced deep learning model, the transformer-enhanced CNN-RNN (TE-CNN-RNN), has been introduced. This model is specifically designed to predict INT data only. This model demonstrates the effectiveness of transformers in extracting detailed temporal-spatial features, outperforming conventional CNN-RNN models. The experimental results demonstrate that the proposed MSCR and TE-CNN-RNN models outperform existing state-of-the-art models for traffic forecasting. These findings underscore the transformative power of transformers for capturing intricate temporal-spatial features and the importance of multi-source data and deep learning techniques for optimizing cell site management in the B5G era.

Keywords: deep learning; data fusion; transformer models; mobile traffic forecasting; recurrent neural networks; convolution neural networks; self-organizing networks

check for updates

Citation: Althamary, I.; Boisguene, R.; Huang, C.-W. Enhanced Multi-Task Traffic Forecasting in Beyond 5G Networks: Leveraging Transformer Technology and Multi-Source Data Fusion. *Future Internet* **2024**, *16*, 159. https://doi.org/10.3390/fi16050159

Received: 18 March 2024
Revised: 19 April 2024
Accepted: 2 May 2024
Published: 5 May 2024

1. Introduction

The exponential growth in data traffic in Beyond 5G (B5G) cellular networks necessitates intelligent infrastructure management solutions, as evidenced by Cisco's observation of a nearly threefold traffic increase for 5G connections compared to 4G [1]. Traditional network expansions face cost barriers and complex management issues, including inter-cell interference [2,3]. Innovations are essential to enhance network performance without incurring disproportionate expenses [2,4–6]. To address this, traffic offloading to mixed networks, supported by B5G technology, offers a promising solution for enhancing coverage and efficiency. Self-organizing network (SON) technology is crucial for the intelligent management of these networks, and its effectiveness depends on accurate network traffic prediction [7–10]. The precise forecasting of internet traffic is crucial for optimizing real-time mobile network performance. This practice can prevent network congestion, manage traffic, and enhance offloading processes while also facilitating the prediction of traffic patterns for future network designs.

Analyzing and predicting mobile internet traffic using big data are crucial to implementing SON technology effectively. This is essential for proactively managing traffic offload in mixed networks. By leveraging historical data from Milan Telecom, we can predict future network traffic and determine the number of small base stations (BSs) that need activation. We propose using a deep learning model to tackle the challenges of mobile internet traffic, which varies across space and time. Deep learning methods have shown promise in uncovering hidden patterns in vehicle traffic flows and traffic management where they have already been used to predict internet traffic flow [7,11,12]. With the ongoing advancements in deep learning, it is possible to solve various problems [13].

Deep learning models have successfully analyzed complex data, including traffic flows. This paper introduces advanced deep learning models designed to address the complexities of predicting and managing network traffic in B5G networks. We investigate how these models can improve network traffic prediction by integrating multiple data sources, aiming to contribute significantly to cellular network management through advanced computational techniques. These techniques employ a data fusion approach to overcome the limitations of single-source analysis. However, integrating diverse data sources can present significant challenges. Recent advancements in transformer-based architectures offer powerful tools for understanding intricate data dependencies, making them ideal for this task [10,14–16].

Building upon our previous research [13], this study introduces a transformer-enhanced CNN-RNN (TE-CNN-RNN) that focuses on INT data analysis and surpasses previous methods by harnessing the power of transformers for refined feature extraction. Also, it introduces a multi-source CNN-RNN (MSCR) that incorporates diverse data sources to provide a more comprehensive analysis of network traffic patterns. Our models leverage the latest transformer advancements, convolutional and recurrent layers, and multi-source data integration, reflecting recent trends in deep learning. This approach positions our work at the forefront of B5G network traffic management and demonstrates the transformative potential of integrating transformer networks and data fusion for improved predictive accuracy.

The contributions of this paper are as follows:

- We introduce two novel deep learning models specifically tailored for B5G cell site management:
  - Transformer-enhanced CNN-RNN (TE-CNN-RNN), which demonstrates the significant impact of transformers, surpassing conventional CNN-RNNs as well as recent transformer-based methods like TSMixer [17] and SAMformer [18] using INT data alone.
  - Multi-source CNN-RNN (MSCR) leverages diverse data (periodic, weather, news, social) and a unique architecture for comprehensive traffic analysis.
- We develop an adaptive weighting mechanism for the advanced MSCR model. This mechanism intelligently prioritizes data sources based on their current relevance, ensuring optimal prediction accuracy
- We comprehensively analyze the impact of different data inputs on network traffic prediction. This analysis offers valuable insights for optimizing B5G cell site management strategies.
- We demonstrate the superior performance of both models through rigorous comparison with established baselines: ARIMA, ST-ResNet [19], T-DenseNet [20], and CNN-RNN [13] and recent transformer-based methods TSMixer [17] and SAMformer [18].

This paper includes a review of related work in Section 2, the optimization of hyperparameters and manuscript preparation in Section 3, an overview of the dataset in Section 4, a detailed explanation of the TE-CNN-RNN and MSCR models along with data preprocessing in Section 5, and the experimental setup and results in Section 6, and concludes with our findings in Section 7.

2. Related Work

Integrating diverse data sources through data fusion plays a crucial role in decision-making across domains, including healthcare, image analysis, traffic management, and satellite observation [11,21–23]. Traditional fusion methods typically focus on early (pre-

prediction) or late (post-prediction) integration. Due to the non-linear nature of many real-world data, late fusion approaches often yield superior results [13,24,25].

The increase in traffic forecasting research highlights the importance of accurate prediction methods given the complex spatio-temporal relationships within traffic systems [21,26–28]. These studies highlight the challenges and future directions and advocate for advanced models capable of deciphering complex traffic flow patterns. Moreover, [29] surveys current traffic classification methods, which can be used for data fusion. The methods are classified into five categories: statistics-based, correlation-based, behavior-based, signature-based, and hybrid-based methods.

Recent advancements in deep learning have enabled more sophisticated multi-source fusion techniques, particularly for addressing time series data. Early work in this area often focused on combining images and text [30–33]. To analyze temporal characteristics, the ST-ResNet model [19] combines inputs and external factors to address proximity, periodicity, and trends within crowd traffic data. However, relying on the residual network limits its full exploration of time characteristics. Additionally, it primarily focuses on the correlation between weather and population traffic, potentially missing insights from other influential data sources. Our work extends this concept by incorporating a more comprehensive range of data sources and advanced techniques to enhance the analysis of temporal characteristics in network traffic.

Several papers leverage the Telecom Italia Dataset for traffic prediction [20,34,35]. The Spatial–Temporal Cross-domain Neural Network (STCNet) employs deep learning and transfer learning to address traffic pattern similarities across different city areas [34]. Zeng et al. [35] expand on this approach by introducing a fusion transfer learning method (STC-N) for multiple cross-domain datasets. To further enhance the prediction accuracy, Nie et al. [36] combines reinforcement learning and a Kullback–Leibler (KL) divergence-based Monte-Carlo Q-learning method. Research also focuses on integrating traffic prediction with energy-aware network management strategies. For example, Wu et al. [37] propose the DeepBSC model, which utilizes a geographical and semantic spatial–temporal network (GS-STN) to forecast traffic demand. This prediction informs dynamic base station sleep control, optimizing energy consumption. Similarly, Huang et al. [38] employ a deep Q network (DQN) alongside traffic forecasting to enhance energy efficiency.

Transformer-based methods offer a powerful approach to capturing long-range dependencies in time-series data, but various deep learning techniques can further enhance traffic prediction. Several recent studies showcase these advancements. For instance, paper [4] proposes a cognitive controller framework that utilizes artificial intelligence (AI) and machine learning (ML) for intelligent network management. The paper showcases the potential of AI for broader network optimization tasks. However, its focus on a specific use case limits its generalizability.

The study by Gu et al. [10] proposes the GLSTTN model for cellular traffic prediction, effectively combining global and local spatial–temporal features using a transformer-based architecture. Ma et al. [39] introduce the CCSANet model, which innovatively integrates a correlation layer in ConvLSTM with a self-attention mechanism, demonstrating improved accuracy over existing methods for extracting spatial features in cellular network traffic prediction. Ferreira et al. [28] provide a comprehensive review and tutorial on network traffic forecasting, examining various models' mathematical foundations and presenting numerical experiments to compare different forecasting methods directly. Finally, Nyalapelli et al. [5] discuss integrating AI and ML technologies in 5G networks, presenting a broad perspective on ML-assisted solutions for various network-level challenges in 5G technology.

Recent advancements in transformer-based architectures further enhance time-series forecasting. TSMixer [17] innovates by combining operations along both time and feature dimensions, achieving success on benchmarks like M5. SAMformer [18] addresses limitations in multivariate long-term forecasting with a sharpness-aware optimized model, significantly outperforming TSMixer using fewer parameters. The iTransformer [40] offers a state-of-the-art solution for capturing complex dependencies, demonstrating robust generalization across datasets. Despite these advancements, challenges remain in handling series with extensive lookback windows, particularly in multi-task scenarios, highlighting the need for continuous innovation.

In advancing network traffic management for 5G SONs, this paper introduces the TE-CNN-RNN model. Building upon recent transformer-based advancements in time-series forecasting, our TE-CNN-RNN incorporates a multi-task learning framework to address network traffic data's intricate spatial and temporal complexities. Integrating a 3D convolutional neural network (CNN), transformer block, and gated recurrent unit (GRU) layer allows our model to capture diverse patterns and dependencies within the data effectively. We further propose the MSCR model, equipped with data fusion mechanisms designed to overcome the limitations of previous approaches that often struggled with extensive lookback windows. These innovations address existing gaps and set a new performance benchmark for traffic management solutions, demonstrating the transformative power of deep learning for optimizing traffic systems and beyond.

3. Methodology

3.1. Optimizing Hyperparameters

Our models were developed using TensorFlow Keras. To ensure a fair evaluation process, we optimized the hyperparameters using Optuna [41], a reliable framework for hyperparameter optimization that helped us identify the optimal parameter configurations for all transformer models, ensuring fair and robust evaluations. We conducted five independent runs with seeds set to [2222, 42, 7891, 123, 1].

3.2. Preparing the Manuscript

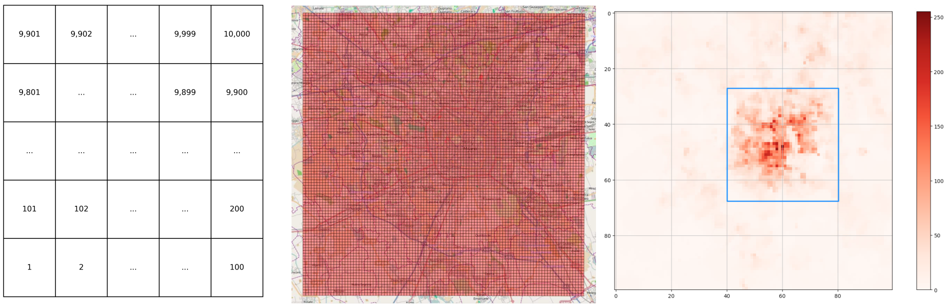
We employed AI tools such as ChatGPT and Grammarly to maintain grammatical accuracy. The authors reviewed the suggestions provided by these tools, and corrections were made based on technical accuracy and contextual relevance.

4. Dataset Description

Using deep learning to analyze large, high-quality datasets is crucial and often difficult to obtain. In this study, the data source for the dataset is a significant challenge. This dataset was organized by the Italian telecom operator TIM in 2015 [42]. This dataset is one of the most comprehensive collections from operators and is publicly available. It mainly includes telecommunications, weather, news, and social networks for Milan and Trentino from November to December 2013. We focus on the telecom record in Milan city for mobile internet forecasts. First, we define a geographic grid for data recording, and the city is divided into 100 × 100 areas. The Milan grid's aggregated call data record (CDR) is shown in Figure 1. Each grid has a unique square ID covering an area of 235 × 235. The data are in the standard WGS84 (EPSG: 4326) GeoJson format, as shown in Figure 1a. Figure 1b overlays this grid on a map of Milan to visually represent how the grid corresponds to the city's layout. Figure 1c is a heat map that displays network traffic intensity across Milan. The axes of the heat map correspond to grid cell IDs along the horizontal and vertical directions. At the same time, the color intensity represents the volume of network traffic, with red indicating higher traffic areas. The blue square highlights a zone of particular interest, which is discussed further in our analysis. Additionally, the dataset contains the following information used in this project:

Internet traffic: internet traffic (INT) in Milan.

- Square ID: the identifier of the square in the Milan grid.
- Time interval: The beginning of the time interval of the record. The end interval time can be obtained by adding 10 min to this value.
- Internet traffic activity: The number of internet activities generated during the time interval in this square id.



9,901	9,902	...	9,999	10,000
9,801	...	...	9,899	9,900
...	...	...	...	...
101	102	...	...	200
1	2	...	...	100

(**a**) (**b**) (**c**)

Figure 1. Milan grid map. (**a**) Milan grid, (**b**) Milan grid overlaid on Milan map, (**c**) Milan network traffic heat map.

Milano Today: The news in Milan.

- Data: the content of the news articles available online (http://www.milanotoday.it, accessed on 6 August 2015).
- Date: the time at which the news event occurred.
- Geometry: the coordinates where the news event happened.

Social Pulse: Tweet records in Milan.

- Data: the locations of tweets sent within the Milan and Trentino area.
- Date: the time at which the tweet was sent.
- Geometry: the coordinates from where the tweet was sent.

The weather data describe the type and intensity of meteorological phenomena in Milan city. The data for Milan were collected by the Agenzia Regionale per la Protezione dell'Ambiente (ARPA). Weather data are divided into two datasets: legend datasets and weather phenomena. In this project, we mainly use the latest dataset. The sensors can measure the following meteorological phenomena: wind direction, wind speed, temperature, relative humidity, precipitation, global radiation, atmospheric pressure, and net radiation.

Weather Phenomena:

- *Sensor ID*: identification string of the sensor.
- *Time Instant*: the time instant of the measurement, which is expressed as a date/time with the following format: YYYY/MM/DD HH24:MI (e.g., 2013/07/22 15:30)
- *Measurement*: The value of meteorological phenomenon intensity measured at the time instant by the sensor. The unit of measurement (UOM) of the value recorded by the given sensor is specified in the legend dataset.

In this paper, we select four types of weather data: temperature, rainfall, wind speed, and air pollution.

5. Proposed Models

Having discussed the Telecom Italia dataset, we now introduce our models designed to address the complexities of network traffic prediction.

5.1. Transformer-Enhanced CNN-RNN (TE-CNN-RNN)

Efficient network traffic prediction is essential for optimizing resource allocation, preventing congestion, and improving the quality of service (QoS) in 5G and future networks. In order to take advantage of the benefits of sophisticated modeling approaches, it's important to understand the complex dynamics of network traffic calls. We introduce an innovative architecture termed the transformer-enhanced CNN-RNN (TE-CNN-RNN) depicted in Figure 2. This architecture ingeniously amalgamates CNNs, transformer blocks, and GRUs via an intricate fusion mechanism, facilitating the adept extraction and interpretation of complex data patterns. This hybrid model is exceptionally versatile, proving effective across many machine learning tasks [7,14,16,43].

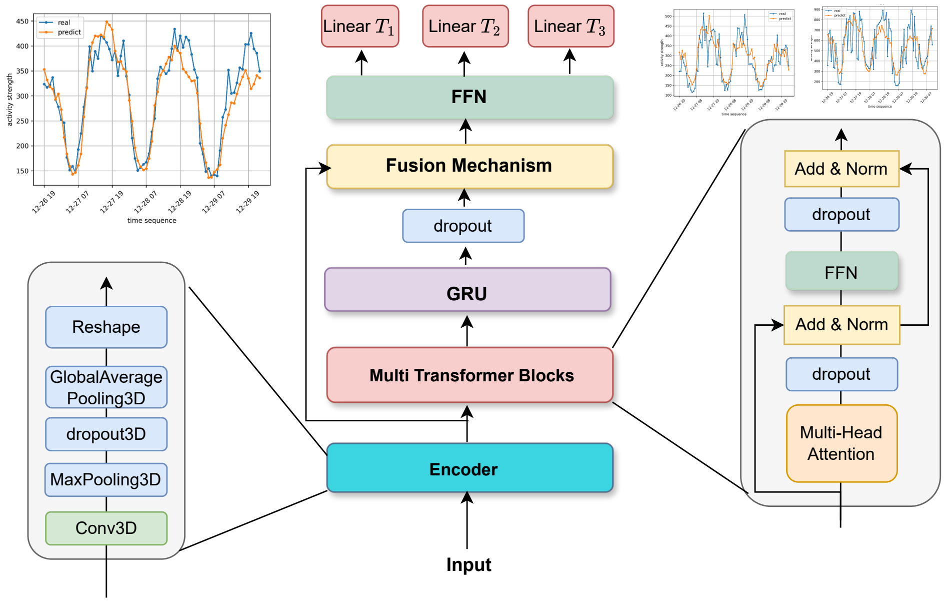


Figure 2. Transformer-enhanced CNN-RNN (TE-CNN-RNN).

5.1.1. Model Architecture Overview

The TE-CNN-RNN architecture begins with a 3D convolutional layer (Conv3D) that acts as the encoder. Its purpose is to extract spatial features from the input data and encapsulate the dimensional intricacies of network traffic. After this spatial feature extraction, the Multi Transformer Blocks process these features, considering their sequential relationships and enriching the model's understanding of temporal patterns. The GRU layer sequentially explores the dynamics over time, providing a profound temporal analysis. This processing sequence concludes with a fusion mechanism that integrates the derived spatial and temporal features to output a refined prediction ready for effective network management.

5.1.2. Multi Transformer Blocks

At the heart of our TE-CNN-RNN framework lies the Multi Transformer Blocks, which utilize a Multi-Head Attention (MHA) mechanism for parallel processing of input data. This architecture is crucial in capturing diverse dependencies, significantly enriching the model's ability to interpret complex data streams.

To overcome the limitations of traditional CNN-RNN architectures, we integrated Multi Transformer Blocks into our TE-CNN-RNN model. These architectures often struggle with temporal dynamics and long-range dependencies in sequential data. The Multi-Head Attention mechanism within these blocks allows our model to focus dynamically on various segments of the input sequence. This feature is particularly crucial in applications like traffic forecasting, where diverse and rapidly changing factors can significantly influence patterns. The MHA mechanism enhances the model's feature representation and decision-making capabilities by simultaneously processing different data aspects, providing a more accurate and robust framework for understanding and predicting complex dynamic systems. The Multi Transformer Blocks encompass the following components:

Multi-Head Attention (MHA)

Multiple attention heads are utilized to process distinct segments of the input sequence in parallel, enhancing the model's ability to recognize network traffic data patterns. Given an input sequence INT $X = (X_0, X_1, \ldots, X_n)$, the Multi-Head Attention mechanism is applied:

$$\mathrm{MHA}(X) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O \tag{1}$$

where

$$\text{head}_i = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V) \tag{2}$$

W^Q, W^K, W^V, and W^O are the parameter matrices for query, key, value, and output, respectively. The INT referred to earlier is the original data, i.e., the regular data before feature extraction, as explained in Section 5.3.1.

Dropout and Residual Connections

Dropout and residual connections are integrated within our architecture to prevent overfitting and ensure that important information is retained throughout the network, thus promoting smoother gradient flow and enhancing generalization. These connections combine the MHA's output with the input sequence, preserving vital data and enabling a smoother gradient flow.

$$X' = \text{Dropout}(\text{MHA}(X)) + X \tag{3}$$

Layer Normalization

Layer normalization standardizes the data across features within each sample, stabilizing training dynamics and accelerating convergence. Applied before and after the feed-forward network, it normalizes the data across features, ensuring a consistent mean and variance. This stabilizes gradient propagation during training and prevents any single feature from dominating the learning process, leading to more stable updates.

$$X_{\text{norm}} = \text{LayerNorm}(X') \tag{4}$$

Feed-Forward Network (FFN)

The feed-forward network (FFN) deepens the architecture, enabling the model to interpret the refined data from the MHA and other preprocessing steps further. This is essential for making accurate final predictions or classifications. It consists of two dense layers with ReLU activation in between, aimed at processing the sequence further to uncover complex data relationships.

$$\text{FFN}(X_{\text{norm}}) = \text{ReLU}(X_{\text{norm}}W_1 + b_1)W_2 + b_2 \tag{5}$$

with W_1, W_2, b_1, and b_2 being the weights and biases of the FFN layers.

The Output

Finally, the architecture applies another layer normalization step to the FFN's output, yielding the finalized feature set for subsequent fusion.

$$X_{\text{final}} = \text{LayerNorm}(\text{FFN}(X_{\text{norm}}) + X_{\text{norm}}) \tag{6}$$

5.1.3. Fusion Mechanism

The fusion mechanism is a critical component of our model, integrating features extracted by the CNN and GRU layers. It utilizes a gating strategy to modulate the flow of information, selectively emphasizing the most relevant features for precise prediction. The gated fusion is computed by concatenating the outputs from the CNN and GRU layers, followed by a sigmoid activation that forms a gate-controlling feature significance.

$$G = \sigma(\mathbf{W}_g \cdot [A; B] + \mathbf{b}_g) \tag{7}$$

where A and B are the flattened outputs from the CNN and GRU layers, respectively, and $\mathbf{W}_g$ and $\mathbf{b}_g$ are learnable gate parameters. The resultant gated features are then combined element-wise with the concatenated features and further processed through a dense layer, culminating in the final fused output that embodies the spatial–temporal intricacies required for accurate network traffic forecasting.

$$F = \mathrm{ReLU}(\mathbf{W}_f \cdot (G \odot [A; B]) + \mathbf{b}_f) \quad (8)$$

where $\odot$ represents element-wise multiplication.

5.2. Multi-Source CNN-RNN (MSCR) Data Fusion

To address the complexities of multi-source data, we introduce MSCR data fusion.

5.3. Architecture Overview

This section introduces our proposed framework for integrating diverse data categories to enhance traffic flow prediction. Figure 3 outlines the architecture designed to handle input data with spatial and temporal characteristics. We incorporate CNNs and long short-term memory (LSTM) models into our framework to capture these features.

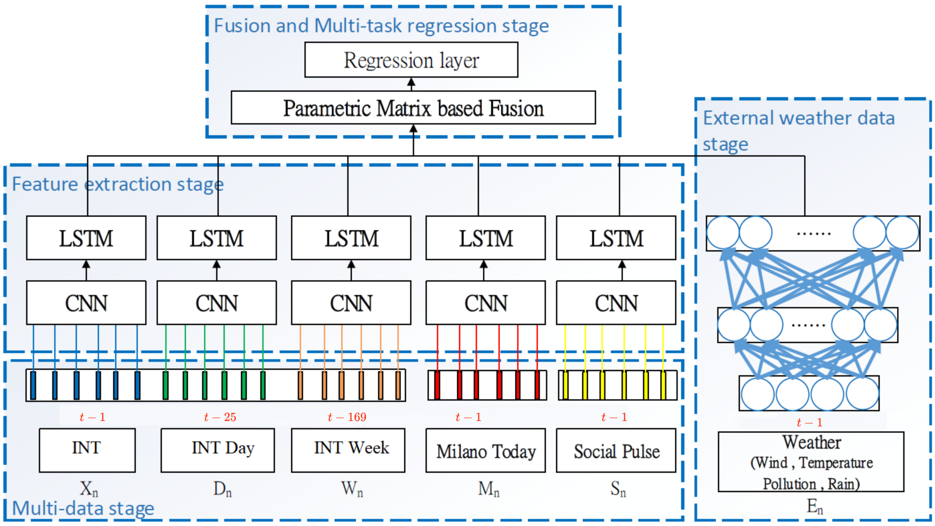


Figure 3. Multi-Source CNN-RNN (MSCR).

Successfully handling diverse data types depends on using a model that optimally processes these sources. Simply combining them arbitrarily may not yield the best results. We leverage deep learning techniques to discover underlying patterns and relationships between data types.

Our previous research [13] focused on INT data in Milan. In this work, we expand our analysis with weather, news, and social media data to improve our model's predictive power. We recognize that INT flow varies from hour to hour but shows similarities within the same day or week. Our fusion model extracts these periodic features using INT data collected at different times. We also include correlated INT data based on a correlation coefficient map. Additionally, we incorporate Milano Today and Social Pulse data to enhance our predictions.

Our architectural diagram consists of four stages: grid data processing, large-area dataset processing, feature extraction, and fusion and multi-task regression:

Grid Data Processing: Data are consolidated into a uniform grid format for consistency.

Large-Area Dataset Processing: To ease computation, the resolution of large-area datasets is reduced.

Feature Extraction: CNNs and LSTM models are used to extract spatial and temporal features, respectively, from the data.

Fusion and Multi-Task Regression: A weighted fusion model integrates diverse data types. Multi-task regression models are then trained to generate the final traffic flow predictions.

5.3.1. Processing Grid Data

The structure illustrated in Figure 3 accepts six data types as input. Five of these are directed into the neural network, preserving a consistent architecture. One of the input data types is internet traffic (INT), which is part of the call detail records (CDRs) and is represented by X_n: $\{X_0, \ldots, X_{t-1}\}$. Each data point in INT corresponds to a 100 × 100 grid map, with each interval spanning 10 min. However, our study only considers a 15 × 15 section of the 100 × 100 grid map as the INT input. Moreover, since we focus on forecasting for Milan's city center, we treat the central 3 × 3 area as the target label.

Furthermore, we analyze the correlation coefficient between the internet traffic and the two data sources, aiming to quantify:

- The magnitude and direction of the impact of the two datasets on each other.
- The correlation direction indicates whether they co-occur or have an inverse relationship.

We extract periodic features from INT by selecting different timestamps, as INT exhibits both temporal and periodic characteristics. As depicted in Figure 4, the network traffic consistently peaks simultaneously every day. This repetitive pattern reflects the daily routine of people using the network, with peak usage typically occurring between 11:00–13:00 and 17:00–19:00, represented in hours. Additionally, the weekly pattern affects network traffic, varying significantly between weekends and weekdays, with generally lower and more variable traffic on weekends.

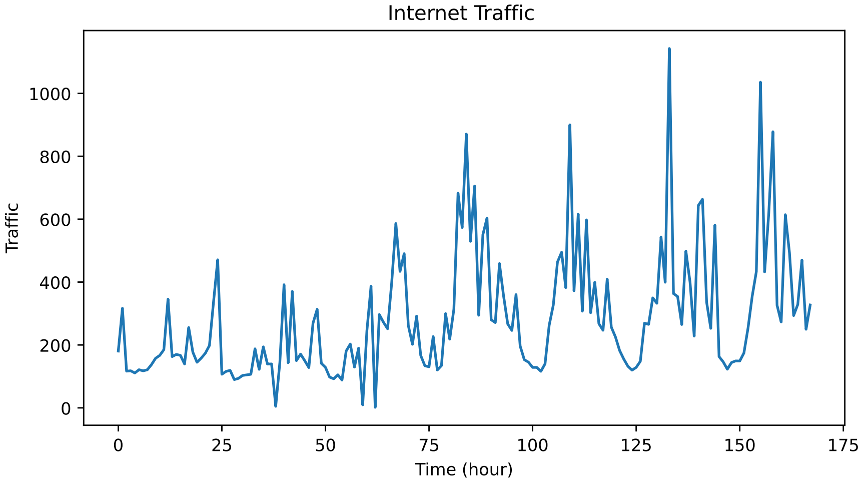
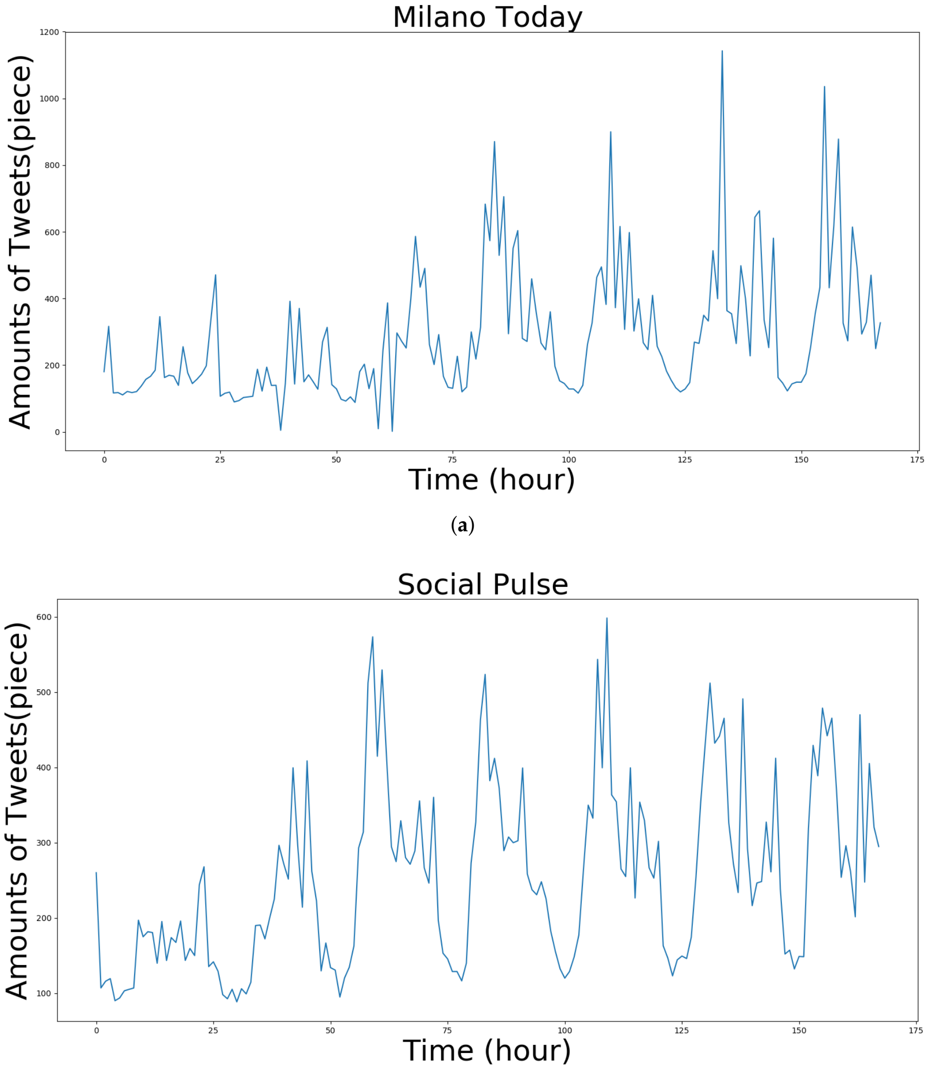


Figure 4. Internet traffic over an hour.

We extract features over daily and weekly time spans to capture these patterns, denoted by l_d and l_w, respectively. Specifically, we collect data points from one-day and one-week periods before the prediction time t, where the number of data points d and w is 24 and 168, corresponding to 10-minute intervals in one day and one week, respectively. Thus, the periods for INT_{day} and INT_{week} are marked as $t - 25$ and $t - 169$ in Figure 3, respectively. These extracted features are labeled as D_n: $\{D_0, \ldots, D_{t-(l_d-1)\times d}\}$ and W_n: $\{W_0, \ldots, W_{t-(l_w-1)\times w}\}$.

The INT referred to earlier is the original data, i.e., the regular data before feature extraction. In summary, the periodic features extracted from INT capture the daily and weekly traffic patterns, subsequently utilized as inputs to the neural network for predicting traffic flow in the Milan city center.

The Milano Today and Social Pulse datasets include news articles, tweet records, and the corresponding coordinates where the events occurred. The data trends for Milano Today (MT) and Social Pulse (SP) are illustrated in Figure 5a,b, showing similarities to the trends observed in INT. These figures indicate that the length of news articles and the volume of tweets increase concurrently with network traffic, suggesting a linkage between these datasets and the daily and weekly patterns of INT traffic.



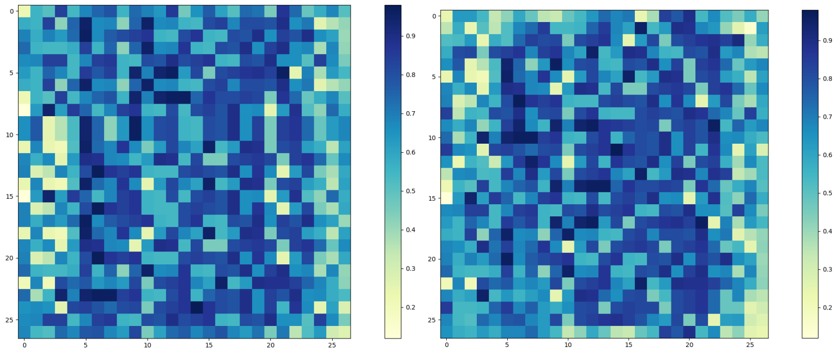
(**a**)

(**b**)

Figure 5. Two types of traffic: Social Pulse and Milano Today. (**a**) Milano Today traffic over an hour, (**b**) Social Pulse traffic over an hour.

According to Figure 6, most grid correlations exceed 0.7, with a correlation coefficient of 0.756 between internet traffic and Social Pulse and 0.831 between internet traffic and Milano Today. A correlation coefficient above 0.7 denotes a strong correlation, with darker colors indicating more vital relevance. Therefore, we infer that these two data sources are highly relevant to internet traffic, justifying their use as inputs in our model to enhance network traffic predictions.

For Social Pulse and Milano Today, the time of occurrence is accurately recorded. However, we align these data with the internet traffic intervals, counting the number of events every 10 min and converting the coordinates of each data point into a 7 × 7 matrix. This adjustment accounts for the data's spatial resolution, given that the error range of Milano Today and Social Pulse is approximately 600 m, and each grid in our dataset represents an area of approximately 200 × 200 meters. We then overlay these matrices onto a 100 × 100 grid, analogous to INT, and focus on the central 15 × 15 area. These data are denoted as M_n: $\{M_0, \ldots, M_{t-1}\}$ and S_n: $\{S_0, \ldots, S_{t-1}\}$, respectively.



(**a**)

(**b**)

Figure 6. Correlation coefficient between Social Pulse and Milano Today. (**a**) Correlation of INT and Social Pulse, (**b**) Correlation of INT and Milano Today.

5.3.2. Processing the Large Area Datasets

Various external factors, weather conditions, and events can significantly impact INT flows. In our weather data category, we integrate four types of data: rainfall, pollution, temperature, and wind speed. These data sources are amalgamated and represented as E_n: $\{E_0, \ldots, E_{t-1}\}$, with each data point collected at an hourly interval.

The dataset includes sensor data for weather conditions, specifically capturing rainfall, pollution, temperature, and wind speed. However, these data are tied to specific sensors in various areas, and grid maps for these data points are not provided. Since our predictive analysis focuses on the downtown area, we select the sensor closest to this central region. To standardize the data, we apply Min-Max normalization to rescale the values within a range of [0, 1]. Subsequently, the four types of weather data are concatenated into a single vector comprising four elements, denoted as E_n. Instead of utilizing CNN and LSTM models, this vector is introduced into a fully connected (FC) layer, effectively encapsulating the features of weather conditions.

The time interval for the weather data deviates from that of INT, Milano Today, and Social Pulse, which are recorded at 10-minute intervals. This discrepancy necessitates a distinct structure to process the weather data. The four-element weather vector, prepared during preprocessing, is fed into two FC layers. The initial layer functions as an embedding layer, capturing the intricacies of weather features. We choose an embedding size of 100 for this purpose. The subsequent layer is responsible for reshaping the data to align with the neural network's output dimensions for data fusion. This reshaping layer is configured to output data with dimensions of 512, ensuring compatibility with the overall neural network architecture.

5.3.3. Feature Extraction Stage

The feature extraction stage of our model emulates the CNN-RNN structure from our prior research. Initially, input data undergoes processing via a CNN to extract spatial features. Given the grid-like structure of digital images, the CNN proves adept at identifying geospatial-based features, as demonstrated in the data illustrated in Figure 1. We employ a CNN with multiple layers to account for spatial dependencies within each region. A single convolution operation addresses only adjacent grid dependencies, which might be insufficient for comprehensive traffic forecasting. While temporal features are paramount in traffic forecasting, standard 2D CNN models typically do not capture them. Conversely, LSTM excellently identifies relationships between temporally contiguous data points. In Figure 3, multiple INT inputs (specifically X_{t-6} to X_{t-1}) are processed through the CNN stage before being sequenced into the recurrent neural network (RNN) stage. The CNN output is then fed into the LSTM, which captures temporal features among the six data points within an hour.

This structure is a deep hierarchical feature extractor, employing a CNN for spatial domain learning and an RNN for temporal domain learning. The approach has succeeded in activity recognition and video description tasks [7,13,44]. Notably, our method distinguishes itself from that described in [45] by incorporating a more comprehensive array of data, including Milano Today, Social Pulse, and various weather data, thus enhancing the LSTM model's comprehensiveness and adaptability.

5.3.4. Data Fusion and Multi-Task Regression Stage

Figure 3 showcases the MSCR model, comprising six components: INT, daily INT, weekly INT, Milano Today, Social Pulse, and external weather data. While data fusion effectively enhances the deep learning model's performance, it introduces challenges regarding the fusion methodology and timing.

To address these challenges, we adopt a **parametric matrix-based fusion** strategy. This strategy entails the integration of a learnable weight fusion layer following the LSTM modules, allowing the model to determine the data fusion method autonomously. The fusion equation is delineated as follows:

$$y = W_1 \circ F_C + W_2 \circ F_D + W_3 \circ F_W + W_4 \circ F_M + W_5 \circ F_S + W_6 \circ F_E \tag{9}$$

where $\circ$ denotes the Hadamard product, W_1 through W_6 represent the learnable weights, and F_C through F_E symbolize the outputs of each LSTM module. This fusion technique proficiently amalgamates features from varied data sources.

For fusion timing, we employ late fusion, which combines features extracted from different networks using distinct learnable weights after the CNN and LSTM modules. This typically results in superior performance compared to early fusion methods.

The multi-task regression layer's structure mirrors that of our previous work [13], comprising two stages: feature extraction and multi-task regression. The lower stage conducts feature extraction, conveying values to the multi-task regression layer. Inputs $X_1, X_2, \ldots, X_n$ undergo several layers of nonlinear feature transformations via diverse deep learning architectures.

The upper stage consists of flattened and regression layers, producing outputs for multiple tasks, denoted as $Y_1, Y_2, \ldots, Y_n$. Multi-task learning (MTL) enhances predictions by joint training on related tasks and sharing parameters across different tasks, facilitating more efficient learning [14,46]. This study aims to predict maximum, average, and minimum INT values within an hour, offering insights into the best, worst, and average network conditions at the forecasted time t and aiding SON management.

6. Experimental Setup and Results

6.1. Experimental Settings

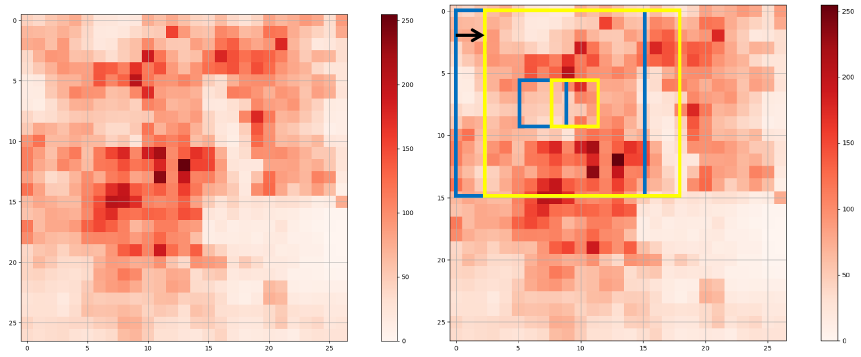
We conducted a comparative study on mobile traffic prediction across various models using the Telecom Italia dataset for Milan City. The models, including ARIMA, ST-ResNet, T-DenseNet, and CNN-RNN [13,19,20], encompass deep learning and traditional forecasting techniques.

For this study, we reformatted INT, Milano Today, Social Pulse, and periodic data into four-dimensional arrays as described in Section 5, with the array parameters (T, H, V, F) set to $(6, 15, 15, 1)$. This configuration spans a 15-by-15 grid area, covering 225 grids in the Milan Grid, each representing six temporal data points within one hour.

The output array parameters were defined as $(T, H, V, F) = (1, 3, 3, 1)$, focusing on predicting network traffic within a central 3×3 grid area from the original 15×15 grid, on emphasizing city center traffic forecasting.

To reflect the broader city-wide traffic, we expanded the prediction scope to approximately a 27×27 grid area, covering the entire city center of Milan. Figure 7a displays the network traffic for this expanded area. Our objective was to predict network traffic over this extensive area.

Different ranges for prediction were selected, as shown in Figure 7b, with the blue and yellow boxes indicating the chosen prediction ranges. The feature extraction was initially performed within the blue box area, followed by predictions within this range before proceeding to the yellow box area. This iterative process, repeated 81 times, covered the entire city center, culminating in an aggregated average to formulate the final prediction.



(**a**) (**b**)

Figure 7. Grid heat map and selection feature. (**a**) 27 × 27 grid heat map, (**b**) Method of selecting features.

In our deep learning model, we optimized its performance by adjusting several parameters. The following are the parameters we set for the model:

- For the abbreviations of each data type, please refer to Table 1.
- CNN-RNN: The overall structure of the CNN-RNN model remains similar to our previous work [13].
- Table 2 summarizes the parameters optimized.
- Training data: 90% of the data from all but the last two weeks for training, and 10% for validation
- Testing data: The second-to-last week.

Table 1. Full names and abbreviations.

Abbreviation	Full Name
INT	Internet traffic
D	Daily features
W	Weekly features
E	External weather data
M	Milano Today
S	Social Pulse
all	Internet+D+W+E+MT+SP

Table 1 shows the full names and abbreviations used for the different data types.

To evaluate the performance of our model, we employed two commonly used metrics: the mean absolute percentage error ($MAPE$) and the root mean square error ($RMSE$). These metrics allow us to assess the accuracy of our predictions. Additionally, we calculated the mean accuracy (MA), which provides a more intuitive measure of accuracy between the real and predicted values. The $MAPE$ and $RMSE$ are defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i|)^2} \tag{10}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{11}$$

The MA is calculated as:

$$MA = (1 - MAPE) \times 100 \tag{12}$$

These metrics provide insights into the accuracy and performance of our deep learning model in predicting mobile traffic. To address the challenges of mobile internet traffic forecasting, we compare our proposed models against a diverse set of baselines. These baselines include statistical models, feature-focused deep learning methods, and recent transformer-based architectures. This multifaceted comparison will illuminate the strengths and potential shortcomings of our model, guiding future refinements in mobile internet traffic forecasting:

- **ARIMA**: ARIMA is always used in statistics and econometrics, particularly in time series analysis.
- **ST-ResNet** [19]: a deep learning model considering periodic features and external weather data.
- **T-DenseNet** [20]: a deep learning model for spatial data and temporal features with DenseNet.
- **CNN-RNN**[13]: a model with CNN and LSTM, but without multi-data.
- **TSMixer** [17] excels by blending operations across time and feature dimensions, achieving success on benchmarks such as M5.
- **SAMformer** [18] extends TSMixer by integrating sharpness-aware optimization and channel-wise attention, enhancing the performance and efficiency in multivariate long-term forecasting.

6.2. Benchmarking with TE-CNN-RNN

Our experiment used advanced hardware and software for reliable performance benchmarks. We used an Intel® Core™ i7-14700 CPU with 28 threads, an NVIDIA GeForce GTX 1080 Ti graphics card, and 31.1 GiB of RAM. The software environment was Ubuntu 20.04.6 LTS with a 64-bit OS type. We also tailored the loss function selection for each model, with the TE-CNN-RNN using the mean absolute error (MAE) and Huber loss according to the prediction context. This approach aimed to enhance the model's predictive accuracy across various scenarios.

Table 2 summarizes the parameters optimized for each model, including those specific to our TE-CNN-RNN architecture. For TE-CNN-RNN, TSMixer and SAMformer, the following parameters were identified through Optuna,

Table 2. Optimization parameters for CNN-RNN (INT), TSMixer, SAMformer, and TE-CNN-RNN models.

<table>
<tr><th>Parameter</th><th>CNN-RNN (INT)</th><th>TSMixer</th><th>SAMformer</th><th colspan="2">TE-CNN-RNN</th></tr>
<tr><td>Learning Rate</td><td>0.00002</td><td>0.009</td><td>0.009</td><td colspan="2">0.009</td></tr>
<tr><td>Batch Size</td><td>96</td><td>58</td><td>58</td><td colspan="2">58</td></tr>
<tr><td>Optimizer</td><td>SGD</td><td>Adam</td><td>Adam</td><td colspan="2">adagrad</td></tr>
<tr><td>Clipnorm</td><td>-</td><td>9.98</td><td>9.98</td><td colspan="2">1.2</td></tr>
<tr><td>L2 norm</td><td>0.001</td><td>1×10^{-5}</td><td>1×10^{-5}</td><td colspan="2">0.0002</td></tr>
<tr><td>Embedding nodes</td><td>100</td><td>-</td><td>-</td><td colspan="2">-</td></tr>
<tr><th colspan="6">Exclusive Parameters for TE-CNN-RNN</th></tr>
<tr><td>Num Heads</td><td>3</td><td>Coefficient</td><td>30</td><td>CNN Filter</td><td>153</td></tr>
<tr><td>Fusion FFN</td><td>246</td><td>Kernel Size</td><td>5</td><td>Time Steps</td><td>5</td></tr>
<tr><td colspan="2">D Model</td><td>90</td><td colspan="2">Dropout Rate</td><td>0.23</td></tr>
</table>

Table 3 presents a comprehensive performance comparison of our CNN-RNN (INT) baseline, TSMixer, SAMformer, and the proposed TE-CNN-RNN model, focusing on their ability to forecast network traffic. The TE-CNN-RNN model is superior across all tasks (Max, Avg, Min), consistently achieving the lowest RMSE scores and highest MA percentages. This validates its accuracy and robustness for time-series analysis.

Table 3. Comparative metrics for CNN-RNN (INT), TSMixer [17], SAMformer [18], and CNN-RNN transformer models.

		CNN-RNN (INT)	TSMixer	SAMformer	TE-CNN-RNN
Task Max	MA (%)	73.6	74.60	74.69	**75.57**
	RMSE	206.28	126.51	126.05	**120.90**
Task Avg	MA (%)	78.9	81.53	82.64	**82.70**
	RMSE	116.45	70.18	68.14	**65.35**
Task Min	MA (%)	72.8	79.97	79.05	**81.55**
	RMSE	133.56	63.75	61.32	**58.94**

TSMixer and SAMformer represent advanced time-series forecasting methods. TSMixer excels in blending operations across time and feature dimensions, while SAMformer enhances efficiency through sharpness-aware optimization and channel-wise attention. However, our TE-CNN-RNN's advantage likely stems from its comprehensive approach. The integration of CNNs, transformer blocks, and GRUs, combined with our fusion mechanism, enables the model to effectively capture complex spatial–temporal patterns and long-range dependencies inherent in network traffic data.

Table 4 complements the findings from the figures, showing that the TE-CNN-RNN achieved the highest MA and the lowest RMSE consistently across all tasks, further confirming its superior performance. The confidence intervals included in the table reflect the robustness and reliability of these results over five runs with different random seeds.

Table 4. Model performance comparison across different tasks multi-run with 95% confidence intervals.

		TSMixer	SAMformer	TE-CNN-RNN
Max Task	MA (%)	73.31 (±2.26)	73.53 (±1.71)	**74.04 (±2.74)**
	RMSE	130.67 (±15.96)	134.61 (±18.39)	**124.23 (±14.75)**
Avg Task	MA (%)	80.50 (±0.88)	80.87 (±0.95)	**81.90 (±0.97)**
	RMSE	71.29 (±10.96)	71.11 (±11.30)	**66.42 (±10.50)**
Min Task	MA (%)	79.50 (±2.26)	78.27 (±2.67)	**81.07 (±2.60)**
	RMSE	64.11 (±7.56)	64.78 (±7.64)	**59.56 (±5.53)**

An analysis of variance (ANOVA) was utilized to assess the prediction accuracy among various time groups and model types, yielding F-statistics of 33.713 ($p < 0.001$) for time groups and 51.838 ($p < 0.001$) for models. These results indicate a highly significant discrepancy in performance. Notably, the TE-CNN-RNN model emerged as the most accurate, consistently outperforming TSMixer and SAMformer over multiple time intervals and across differing levels of internet traffic demand.

The statistical tests support the robustness of the TE-CNN-RNN, demonstrating its predictive reliability that is not merely a consequence of sample variation. This tailored analysis method aligns with our goal of creating a forecasting model adept at navigating the complexities of real-world internet traffic characterized by dynamic temporal shifts and diverse volume changes.

Furthermore, the enduring superiority of the TE-CNN-RNN across a range of prediction tasks affirms its utility and sets a precedent in network traffic forecasting endeavors. This model's performance establishes a new standard in the domain, highlighting the efficacy of the carefully engineered architecture that leverages data fusion. The results also propel the development of our MSCR model, designed to harness multi-faceted data sources' potential.

6.3. Cross-Source Performance

This section presents the experimental results derived from various input data fusion models compared with our study. We evaluated downtown Milan, an area of approximately 750 grids, as illustrated in Figure 1c. The evaluation method uses CDR and other relevant datasets to measure network traffic activity.

Table 5 provides a summarized overview of the performance of the various data input fusion models, considering their unique characteristics. This table illustrates the MA and RMSE values across the 750 grids. The mean accuracy metric represents the average accuracy of the model predictions, while the RMSE quantifies the overall discrepancy between predicted values and ground truth.

Table 5. Compare input data for different characteristics.

		INT	**INT+S**	**INT+M**	**INT+S+M**
Task Max	MA (%)	73.6	74.8	75.1	**75.9**
	RMSE	206.28	202.24	200.11	**197.54**
Task Avg	MA (%)	78.9	79.8	79.7	**80.9**
	RMSE	116.45	115.58	115.97	**111.54**
Task Min	MA (%)	72.8	77.8	78.1	**79.3**
	RMSE	133.56	106.57	104.54	**100.45**

As seen in Table 5, the performance of the INT data in isolation mirrors the results we procured in our previous paper for the Maximum (Max), Average (Avg), and Minimum (Min) tasks [13]. This implies that our architecture effectively predicts network traffic, even in suburban locales characterized by sparse network traffic, as it sustains a high accuracy under such circumstances.

Assessing the data fusion models, we discern that integrating Social Pulse and Milano Today data with INT augments the performance compared to employing INT exclusively. This signifies that these supplementary data sources benefit the network traffic prediction task. Furthermore, upon contrasting the performance of various prediction methodologies, it is consistently observed that the task average MA is superior to the minimum and maximum. This suggests that our model excels at average predictions and exhibits less sensitivity towards extreme values.

In summary, the accuracy attained by incorporating pertinent data sources surpasses the accuracy of the single INT forecasts by approximately 2% to 3%. This underscores the effectiveness of introducing additional data sources to enhance the accuracy of network traffic prediction.

Our approach involves utilizing the temporal characteristics of the INT data by processing them simultaneously in two distinct dimensions: the preceding day and the previous week. We have noted that deep learning models are especially skilled at identifying patterns in the data when both these temporal variables are considered. By considering daily and weekly data simultaneously, we can make better predictions than if we relied only on the forecast of the fusion day.

Table 6 illustrates the prediction performance of integrating daily and weekly temporal characteristics. The table shows combining data from these time scales leads to better predictions. By considering the characteristics of both temporal dimensions together, we observe an improvement in prediction accuracy, with an increase ranging from 3% to 6%, compared to using only the single INT data.

Table 6. Compare the impact of the input period.

		INT	INT+D	INT+W	INT+D+W
Task Max	MA (%)	73.6	75.1	77.1	**79.1**
	RMSE	206.28	199.96	154.36	**78.98**
Task Avg	MA (%)	78.9	79.2	83.8	**84.2**
	RMSE	116.45	116.35	71.24	**65.84**
Task Min	MA (%)	72.8	73.5	78.4	**80.5**
	RMSE	133.56	128.35	101.35	**74.48**

This insight underscores the importance of acknowledging short-term (daily) and long-term (weekly) patterns in network traffic prediction. By simultaneously incorporating daily and weekly temporal characteristics, our model leads to significantly more accurate predictions and a holistic understanding of temporal dynamics.

We have incorporated weather data into our previous analysis to measure the influence of weather conditions on data fusion and investigate their role in enhancing accuracy. According to the findings outlined in Table 7, relying solely on INT or a combination of INT with Social Pulse or Milano Today does not yield the most favorable predictions. However, an improvement in accuracy ranging from 0.5% to 1% is noticeable when combining periodic data with weather conditions, as demonstrated in Table 6. This suggests that incorporating weather data can be a significant factor in enhancing the accuracy of predictions.

Table 7. Comparison of the weather effect.

		INT	INT+E	INT+S+E	INT+M+E	INT+S+M+E	INT+D+E	INT+W+E	INT+D+W+E
Task Max	MA (%)	73.6	72.9	74.5	74.8	75.3	75.5	77.6	**80.9**
	RMSE	206.28	209.35	204.32	203.95	199.63	198.67	**151.24**	173.25
Task Avg	MA (%)	78.9	78.5	79.9	80.1	81.2	79.6	83.6	**85.6**
	RMSE	116.45	118.36	114.69	113.63	109.54	115.32	69.32	**64.37**
Task Min	MA (%)	72.8	72.5	77.5	77.8	78.6	74.1	79.2	**81.9**
	RMSE	133.56	135.24	107.35	106.32	99.79	124.35	92.35	**67.12**

In Table 8, we compare the performances of five distinct scenarios: solo INT, INT paired with news, INT integrated with periodic data, INT combined with news and periodic data, and fusion of all available data. It becomes evident from the table that including data linked to INT, such as news and periodic data, boosts the prediction accuracy. Furthermore, the results suggest that periodic data appear more beneficial than Milano Today, Social Pulse, and weather data, as the periodic patterns are naturally embedded within the INT data. It is important to highlight that integrating more relevant data does not trigger interference among the different data sources. Applying deep learning to identify correlations between these data sources enhances the model's overall performance.

Table 8 compares the impact of different fusion strategies on the data, namely early and late fusion. Early fusion involves preprocessing the data into 15 × 15 grids and merging same-grid data before feeding them into the CNN + LSTM architecture. This process aims to capture the data's temporal and spatial characteristics. It is important to note that weather data were not included in the early fusion approach. The results show that early fusion, which integrates the data without extracting features, reduces the prediction accuracy, even when the data sources have some relation. This is likely due to the architecture's inability to capture all the data's attributes when combined without applying suitable feature extraction techniques. On the other hand, the late fusion approach improves the accuracy of the prediction. In this approach, the neural network performs feature extraction and identifies correlations within the data. The results show that late fusion outperforms early fusion in terms of prediction accuracy. This highlights the importance of neural-network-based feature extraction in enhancing fusion results, as it provides a deeper understanding of the data's characteristics and correlations.

Table 8. Fusion comparison.

		Early Fusion				**Late Fusion**			
		INT	**INT+S+M**	**INT+D+W**	**INT+S+M+D+W**	**INT+S+M**	**INT+D+W**	**INT+S+M+D+W**	**CNN-RNN (fusion)**
Task Max	MA (%)	73.6	72.1	68.9	71.5	75.9	79.1	80.6	**81.7**
	RMSE	206.28	215.62	234.62	227.96	197.54	78.98	76.34	**75.52**
Task Avg	MA (%)	78.9	76.8	72.4	75.9	80.9	84.2	86.9	**87.4**
	RMSE	116.45	118.65	134.51	127.26	111.54	65.84	64.01	**59.34**
Task Min	MA (%)	72.8	71.8	67.5	69.9	79.3	80.5	83.4	**84.1**
	RMSE	133.56	135.36	158.63	147.68	100.45	74.48	66.31	**66.32**

6.4. Overall Performance

In this sub-section, we will discuss the overall performance of the MSCR model in managing and analyzing diverse data sources. The MSCR model is designed to handle multi-source data fusion effectively.

Table 9 summarizes various model performances—including ST-ResNet, T-DenseNet, ARIMA, and our MSCR framework—in predicting network traffic. The table depicts the MA and RMSE across the 750 grids.

Table 9. Comparison of different works.

		ARIMA	**CNN-RNN (INT)**	**ST-ResNet**	**T-DenseNet**	**INT+D+W**	**CNN-RNN (fusion)**
Task Max	MA (%)	64.1	73.6	71.5	79.5	79.1	**81.7**
	RMSE	239.35	206.28	224.36	78.83	78.98	**75.52**
Task Avg	MA (%)	76.3	78.9	78.6	82.9	84.2	**87.4**
	RMSE	134.65	116.45	118.36	68.35	65.84	**59.34**
Task Min	MA (%)	68.3	72.8	73.6	80.5	80.5	**84.1**
	RMSE	154.35	133.56	124.36	76.36	74.48	**66.32**

ST-ResNet leverages a deep residual network (ResNet) to forecast crowd traffic. It integrates periodic data and weather information to fuse the raw data. However, its reliance solely on a CNN, without including LSTM, might limit its effectiveness in capturing temporal features. Also, it does not consider other data sources like Milano Today. Despite these constraints, it achieves a superior accuracy of 73.6% in the minimum task, marking a 1% increase compared to previous work.

T-DenseNet employs a dense convolutional network (DenseNet) architecture to forecast call data. This model interconnects each neural network layer, enabling efficient feature extraction and reducing the parameter count. However, T-DenseNet does not employ LSTM, thus limiting its ability to capture time-related characteristics fully. Despite this, it achieves reasonable accuracy levels, improving the prediction accuracy of the predecessor model by approximately 5% across the task maximum, task average, and task minimum.

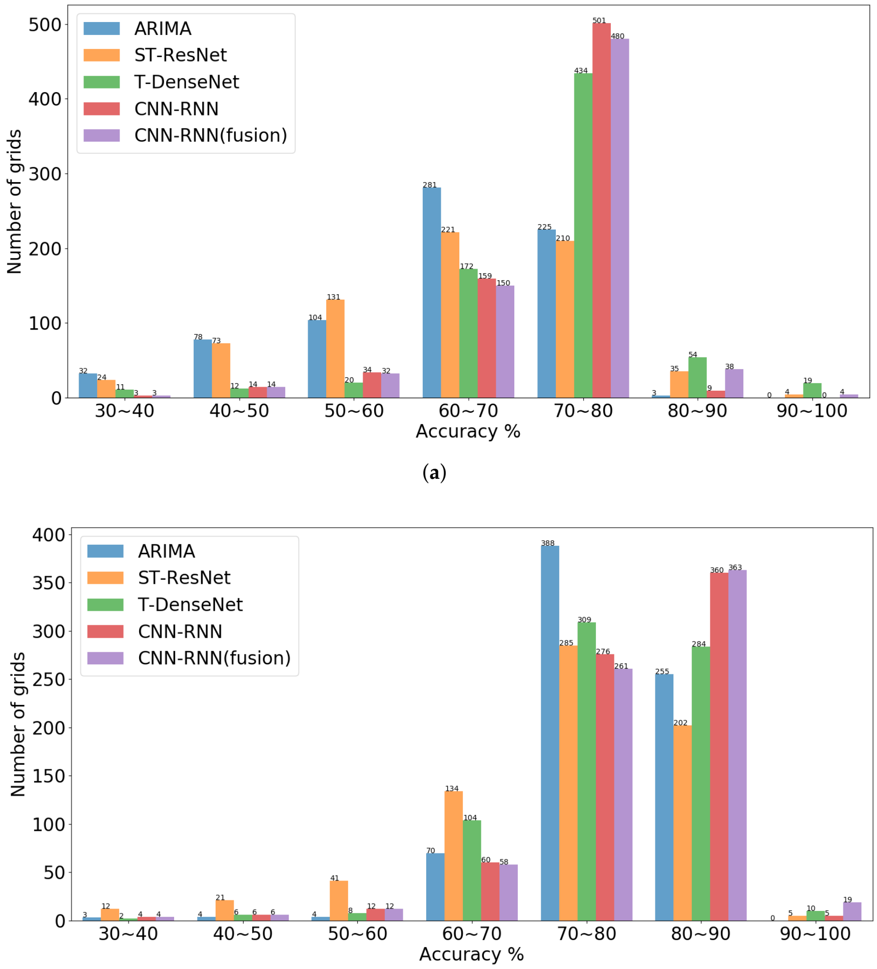
ARIMA, a conventional time-series model, achieves approximately 76.5% accuracy in the task average. However, it falls short in capturing uncertainties, such as the maximum and minimum task values, where it achieves only 64% and 68% accuracy, respectively.

This illustrates the limitations of traditional time-series models in accurately predicting network traffic.

Conversely, our MSCR framework integrates LSTM to capture time dependencies effectively. By merging various data sources and utilizing LSTM, our framework secures higher accuracy rates across the task maximum, task average, and task minimum compared to ST-ResNet and T-DenseNet. Furthermore, our model surpasses the prior work by approximately 7% in prediction accuracy. In the task maximum, task average, and task minimum, our model attains accuracy levels exceeding 80%.

Our MSCR framework, incorporating LSTM and diverse data sources, outperforms other models in predicting network traffic. Moreover, to advance the field of intelligent information systems, we have developed two distinct models: the transformer and the CNN-RNN (fusion). Both models represent our innovative approach to data integration and prediction tasks. Below, we present a comparative analysis of their performance. The transformer model is designed for high precision in scenarios with well-defined patterns, while the all-fusion model is tailored to deliver consistent accuracy across diverse and complex datasets.

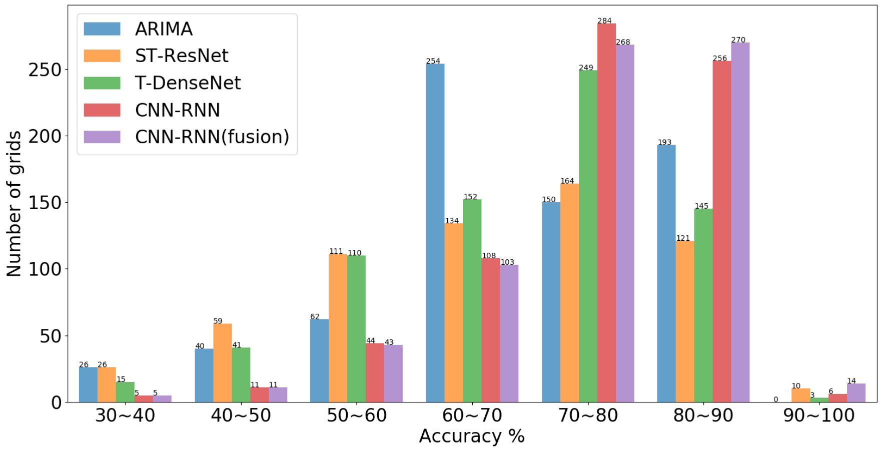
Figure 8 offers a detailed visual representation of the forecasting accuracy distribution for each method. This focuses on the expansive 27 × 27 grid area and displays the accuracy distribution for the minimum, average, and maximum tasks.



(**a**)

(**b**)

Figure 8. *Cont.*



(**c**)

Figure 8. Performance comparison of different methods for network traffic prediction. (**a**) Task Max: accuracy distribution, (**b**) Task Avg: accuracy distribution (**c**) Task Min: accuracy distribution.

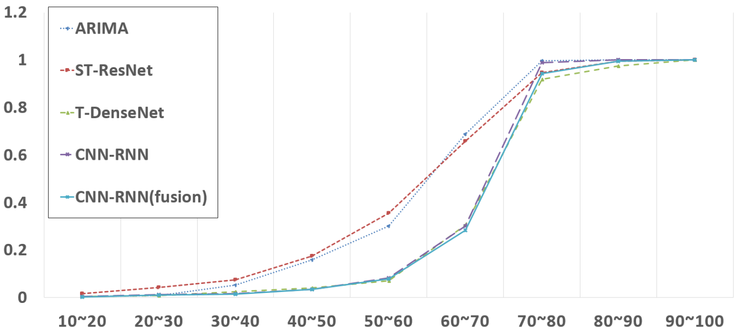
In the context of the task minimum, both CNN-RNN and CNN-RNN (fusion) outshine other methods with accuracy levels ranging from 70% to 90%. This performance could be due to the inclusion of LSTM, which effectively captures temporal features. Notably, around 550 of the total 750 grids attain a prediction accuracy exceeding 70%.

Regarding the task average, the top three prediction methods CNN-RNN, CNN-RNN(fusion), and T-DenseNet achieve accuracy in the 70% to 90% range. Interestingly, ARIMA also fares well within this range due to its lower variance at the task average.

For the task maximum, which is substantially relevant for traffic offloading applications, CNN-RNN (fusion), T-DenseNet, and CNN-RNN demonstrate superior performance, with accuracy ranging between 70% and 90%. Nonetheless, it is worth highlighting that T-DenseNet exhibits m accurate grids within the 80% to 90% and 60% to 70% accuracy ranges.

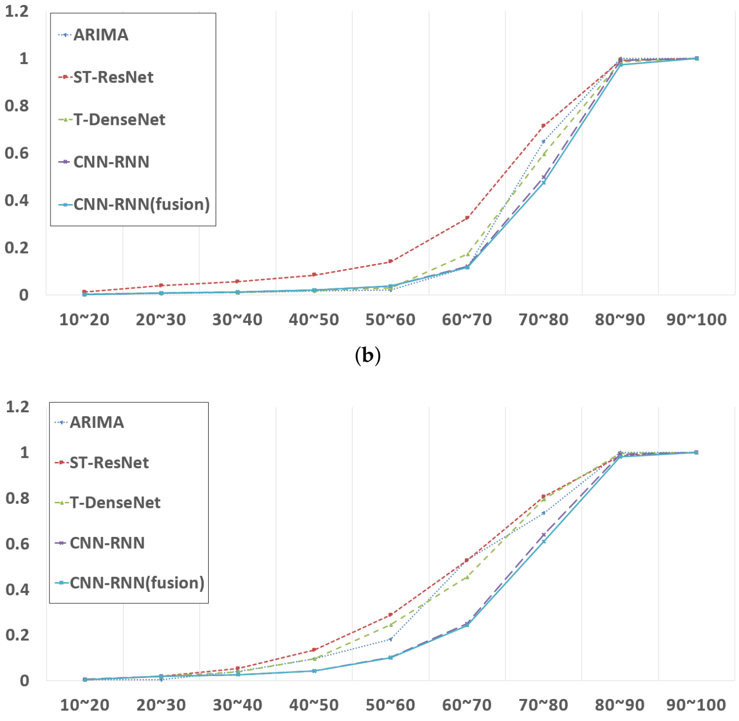
Our results demonstrate that both models significantly outperform existing methods, thus providing practical tools for optimizing B5G network management. This study seeks to answer how these advanced models can predict and manage network traffic more effectively in B5G networks. Also, the original CNN-RNN model displays promising results when addressing the network traffic problem across different tasks. Moreover, integrating diverse data types enhances the prediction accuracy across various grids.

As illustrated in Figure 8, the accuracy rate is mainly within the 60% to 80% range. To delve deeper into the accuracy distribution, we employed the cumulative distribution function (CDF) depicted in Figure 9.



(**a**)

Figure 9. *Cont.*



(**b**)

(**c**)

Figure 9. Cumulative comparison of different methods of network traffic prediction. (**a**) Task Max: Accurate cumulative distribution, (**b**) Task Avg: Accurate cumulative distribution, (**c**) Task Min: Accurate cumulative distribution.

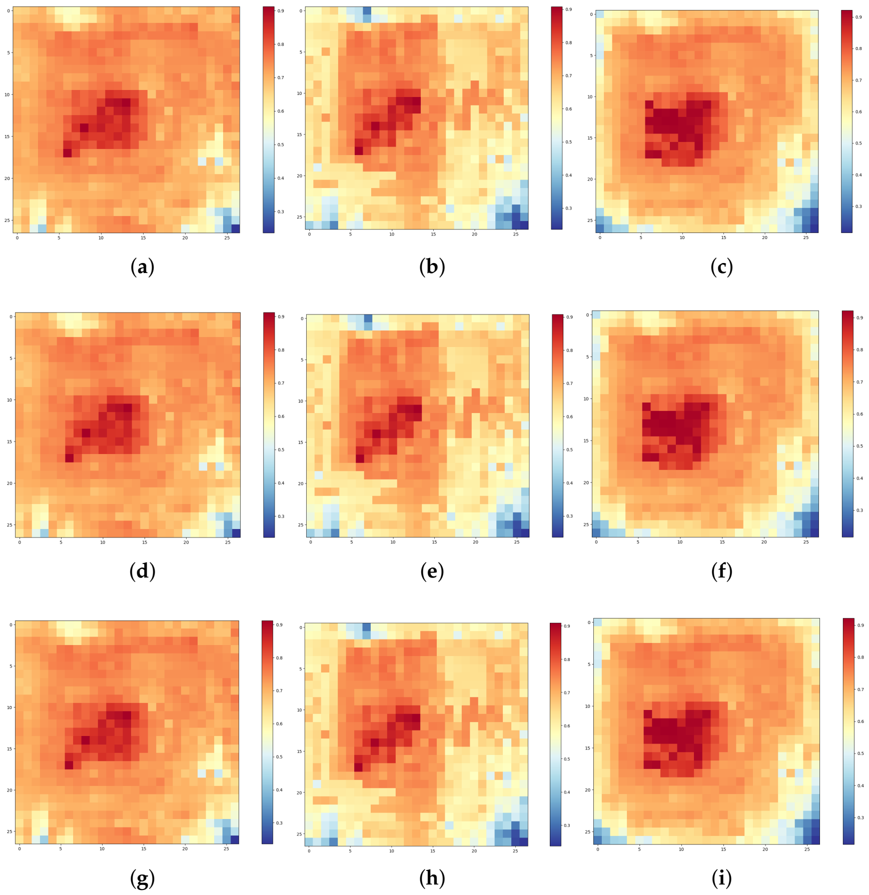
Regarding the maximum task, our approach (CNN-RNN (fusion)), as noted in Table 9, achieves an accuracy of 81.7%, marginally higher than the T-DenseNet model. However, Figure 9a reveals that the T-DenseNet model's cumulative accuracy surpasses 70% in a more significant number of grids. This implies that the T-DenseNet model has more grids with high or low accuracies. In contrast, our method demonstrates a more consistent performance, yielding an average accuracy within the 60% to 80% range.

Our method outperforms other techniques for average and minimum tasks, as reflected in Table 9. The CDF chart shows a more substantial accumulation of grids with over an 80% accuracy. Most of our method's grids achieve an accuracy above 70%, and the average accuracy significantly surpasses other methods.

In conclusion, our method exhibits a higher average accuracy for each grid, especially in the average and minimum tasks. The CDF analysis shows that our method accumulates more grids with an accuracy above 80%.

As evident in Figure 9, both the ARIMA and T-DenseNet methods tend to accumulate more grids with high and low accuracy, signaling a broad distribution of prediction performance. We introduce color maps to investigate the predicted values on each grid further, emphasizing the accuracy levels for different tasks, including task max, task avg, and task min.

In Figure 10, we observe that the T-DenseNet and ARIMA methods exhibit lower predicted values in the low-flow regions of the 27 × 27 grid. In contrast, the central region with a higher traffic flow displays higher predicted values than our method. This pattern implies that these methods might capture high-traffic areas more accurately but fail to predict low-flow regions precisely. On the contrary, our method predicts the average value across the entire grid, resulting in a predominantly reddish color map. This signifies that our method achieves an accuracy of over 70% in a significant portion of the grids, delivering a more balanced and, on the whole, superior accuracy prediction.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 10. Accuracy map of Max/Avg/Min value. (**a**) INT(Max), (**b**) ST-ResNet(Max), (**c**) T-DenseNet(Max), (**d**) INT(Avg), (**e**) ST-ResNet(Avg), (**f**) T-DenseNet(Avg), (**g**) INT(Min), (**h**) ST-ResNet(Min), (**i**) T-DenseNet(Min).

In conclusion, the color maps provide visual insights into the predicted values, underscoring the superior performance of our method in securing a higher accuracy across the entire grid, particularly in the task average.

7. Conclusions

This study demonstrates the potential of combining multiple data sources to predict network traffic. It highlights the effectiveness of two deep learning models, multi-source CNN-RNN (MSCR) and transformer-enhanced CNN-RNN (TE-CNN-RNN), in achieving this task. The MSCR model integrates various data types, such as weather, social trends, and network traffic patterns. It significantly improves the accuracy and adaptability as it scales from a limited 3 × 3 grid to an extensive 27 × 27 grid. The TE-CNN-RNN model performs exceptionally well in precision and shows enhanced feature extraction capabilities. Both models outperform established benchmarks such as TSMixer, SAMformer, and classic ARIMA, setting a new standard for network traffic forecasting. The study highlights the importance of tailoring deep learning techniques to handle complex network traffic data effectively. Future research efforts will enhance these models to process a broader range of data inputs, identify subtle distinctions in low-flow areas, and incorporate real-time and multi-source data from various network scenarios. These advancements will substantially improve network management technologies and pave the way for more advanced network management systems.

Author Contributions: Conceptualization, I.A., R.B. and C.-W.H.; methodology, I.A. and C.-W.H.; software, I.A. and C.-W.H.; validation, I.A., R.B. and C.-W.H.; formal analysis, I.A. and C.-W.H; investigation, I.A. and C.-W.H.; resources, I.A. and C.-W.H.; data curation, I.A. and C.-W.H.; writing—original draft preparation, I.A. and C.-W.H; writing—review and editing, I.A., R.B. and C.-W.H.; visualization, I.A.; supervision, C.-W.H.; project administration, I.A. and C.-W.H.; funding acquisition, C.-W.H. All authors have read and agreed to the published version of the manuscript.

Funding: The research is based on work supported by the National Science and Technology Council, Taiwan, under Grant number NSTC 112-2221-E-008-059-MY2.

Data Availability Statement: The datasets used in this study are openly accessible under the Harvard Dataverse. They include telecommunications, weather, news, and social networks data for Milan and Trentino from November to December 2013, organized by TIM in 2015. For more details and access, please refer to the Harvard Dataverse at DOI: 10.7910/DVN/UTLAHU and the related publication in Scientific Data, 2015 (DOI: 10.1038/sdata.2015.55).

Acknowledgments: While preparing this work, the author(s) used ChatGPT and Grammarly to check grammar and enhance English language skills. After using these services, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

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

References

1. Cisco, U. Cisco annual internet report (2018–2023) white paper. 2020. *Acessado* **2021**, *10*, 1–35.
2. Borralho, R.; Mohamed, A.; Quddus, A.U.; Vieira, P.; Tafazolli, R. A survey on coverage enhancement in cellular networks: Challenges and solutions for future deployments. *IEEE Commun. Surv. Tutorials* **2021**, *23*, 1302–1341. [CrossRef]
3. Kavehmadavani, F.; Nguyen, V.D.; Vu, T.X.; Chatzinotas, S. Intelligent Traffic Steering in beyond 5G Open RAN based on LSTM Traffic Prediction. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 7727–7742. [CrossRef]
4. Patro, S.; Rath, H.K.; Nadaf, S.M.; Mishra, G. A Cognitive Approach for Management and Orchestration of Heterogeneous Networks. In Proceedings of the 2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 8–9 April 2023; pp. 748–753.
5. Nyalapelli, A.; Sharma, S.; Phadnis, P.; Patil, M.; Tandle, A. Recent Advancements in Applications of Artificial Intelligence and Machine Learning for 5G Technology: A Review. In Proceedings of the 2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS), Nagpur, India, 5–6 April 2023; pp. 1–8.
6. Nan, H.; Li, R.; Zhu, X.; Ma, J.; Niyato, D. An Efficient Data-driven Traffic Prediction Framework for Network Digital Twin. *IEEE Netw.* **2023**, *38*, 22–29. [CrossRef]
7. Zhang, Y.; Liu, S.; Zhang, P.; Li, B. GRU-and Transformer-Based Periodicity Fusion Network for Traffic Forecasting. *Electronics* **2023**, *12*, 4988. [CrossRef]
8. Goudarzi, S.; Kama, M.N.; Anisi, M.H.; Soleymani, S.A.; Doctor, F. Self-organizing traffic flow prediction with an optimized deep belief network for internet of vehicles. *Sensors* **2018**, *18*, 3459. [CrossRef] [PubMed]
9. Ayub, A.; Jangsher, S.; Butt, M.M.; Maud, A.R.; Bhatti, F.A. A comparative analysis of Wi-Fi offloading and cooperation in small-cell network. *Electronics* **2021**, *10*, 1493. [CrossRef]
10. 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]
11. Almukhalfi, H.; Noor, A.; Noor, T.H. Traffic management approaches using machine learning and deep learning techniques: A survey. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108147. [CrossRef]
12. Li, M.; Wang, Y.; Wang, Z.; Zheng, H. A deep learning method based on an attention mechanism for wireless network traffic prediction. *Ad Hoc Netw.* **2020**, *107*, 102258. [CrossRef]
13. Huang, C.; Chiang, C.; Li, Q. A study of deep learning networks on mobile traffic forecasting. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–6. [CrossRef]
14. Chen, Z.; Jiaze, E.; Zhang, X.; Sheng, H.; Cheng, X. Multi-task time series forecasting with shared attention. In Proceedings of the 2020 International Conference on Data Mining Workshops (ICDMW), Sorrento, Italy, 17–20 November 2020; pp. 917–925.
15. Qiang, H.; Guo, Z.; Xie, S.; Peng, X. MSTFormer: Motion Inspired Spatial–temporal Transformer with Dynamic-aware Attention for long-term Vessel Trajectory Prediction. *arXiv* **2023**, arXiv:2303.11540.
16. Zhou, X.; Bilal, M.; Dou, R.; Rodrigues, J.J.; Zhao, Q.; Dai, J.; Xu, X. Edge Computation Offloading with Content Caching in 6G-Enabled IoV. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 2733–2747. [CrossRef]
17. Chen, S.A.; Li, C.L.; Yoder, N.; Arik, S.O.; Pfister, T. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv* **2023**, arXiv:2303.06053.
18. Ilbert, R.; Odonnat, A.; Feofanov, V.; Virmaux, A.; Paolo, G.; Palpanas, T.; Redko, I. Unlocking the Potential of Transformers in Time Series Forecasting with Sharpness-Aware Minimization and Channel-Wise Attention. *arXiv* **2024**, arXiv:2402.10198.

19. Zhang, J.; Zheng, Y.; Qi, D. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, USA, 4–9 February 2017; pp. 1655–1661.
20. 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]
21. Ounoughi, C.; Yahia, S.B. Data fusion for ITS: A systematic literature review. *Inf. Fusion* **2023**, *89*, 267–291. [CrossRef]
22. Lau, B.P.L.; Marakkalage, S.H.; Zhou, Y.; Hassan, N.U.; Yuen, C.; Zhang, M.; Tan, U.X. A survey of data fusion in smart city applications. *Inf. Fusion* **2019**, *52*, 357–374. [CrossRef]
23. Xiao, Y.; Li, X.; Yin, J.; Liang, W.; Hu, Y. Adaptive multi-source data fusion vessel trajectory prediction model for intelligent maritime traffic. *Knowl. Based Syst.* **2023**, *277*, 110799. [CrossRef]
24. Pereira, L.M.; Salazar, A.; Vergara, L. A Comparative Study on Recent Automatic Data Fusion Methods. *Computers* **2023**, *13*, 13. [CrossRef]
25. Cunha, N.; Barros, T.; Reis, M.; Marta, T.; Premebida, C.; Nunes, U.J. Multispectral Image Segmentation in Agriculture: A Comprehensive Study on Fusion Approaches. *arXiv* **2023**, arXiv:2308.00159.
26. Tedjopurnomo, D.A.; Bao, Z.; Zheng, B.; Choudhury, F.M.; Qin, A. A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 1544–1561. [CrossRef]
27. Yin, X.; Wu, G.; Wei, J.; Shen, Y.; Qi, H.; Yin, B. Deep learning on traffic prediction: Methods, analysis, and future directions. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4927–4943. [CrossRef]
28. Ferreira, G.O.; Ravazzi, C.; Dabbene, F.; Calafiore, G.C.; Fiore, M. Forecasting Network Traffic: A Survey and Tutorial with Open-Source Comparative Evaluation. *IEEE Access* **2023**, *11*, 6018–6044. [CrossRef]
29. Zhao, J.; Jing, X.; Yan, Z.; Pedrycz, W. Network traffic classification for data fusion: A survey. *Inf. Fusion* **2021**, *72*, 22–47. [CrossRef]
30. Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; Ng, A.Y. Multimodal deep learning. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011; pp. 689–696.
31. Zhao, X.; Jia, Y.; Li, A.; Jiang, R.; Song, Y. Multi-source knowledge fusion: A survey. *World Wide Web* **2020**, *23*, 2567–2592. [CrossRef]
32. Zhang, X.; Jiang, H.; Xu, N.; Ni, L.; Huo, C.; Pan, C. MsIFT: Multi-source image fusion transformer. *Remote Sens.* **2022**, *14*, 4062. [CrossRef]
33. Zou, X.; Yan, Y.; Hao, X.; Hu, Y.; Wen, H.; Liu, E.; Zhang, J.; Li, Y.; Li, T.; Zheng, Y.; et al. Deep Learning for Cross-Domain Data Fusion in Urban Computing: Taxonomy, Advances, and Outlook. *arXiv* **2024**, arXiv:2402.19348.
34. 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]
35. Zeng, Q.; Sun, Q.; Chen, G.; Duan, H.; Li, C.; Song, G. Traffic Prediction of Wireless Cellular Networks Based on Deep Transfer Learning and Cross-Domain Data. *IEEE Access* **2020**, *8*, 172387–172397. [CrossRef]
36. Nie, L.; Ning, Z.; Obaidat, M.; Sadoun, B.; Wang, H.; Li, S.; Guo, L.; Wang, G. A Reinforcement Learning-based Network Traffic Prediction Mechanism in Intelligent Internet of Things. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2169–2180. [CrossRef]
37. Wu, Q.; Chen, X.; Zhou, Z.; Chen, L.; Zhang, J. Deep Reinforcement Learning With Spatio-Temporal Traffic Forecasting for Data-Driven Base Station Sleep Control. *IEEE/ACM Trans. Netw.* **2021**, *29*, 935–948. [CrossRef]
38. Huang, C.W.; Chen, P.C. Joint demand forecasting and DQN-based control for energy-aware mobile traffic offloading. *IEEE Access* **2020**, *8*, 66588–66597. [CrossRef]
39. Ma, X.; Zheng, B.; Jiang, G.; Liu, L. Cellular Network Traffic Prediction Based on Correlation ConvLSTM and Self-Attention Network. *IEEE Commun. Lett.* **2023**, *27*, 1909–1912. [CrossRef]
40. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. itransformer: Inverted transformers are effective for time series forecasting. *arXiv* **2023**, arXiv:2310.06625.
41. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 4–8 August 2019; pp. 2623–2631.
42. Barlacchi, G.; De Nadai, M.; Larcher, R.; Casella, A.; Chitic, C.; Torrisi, G.; Antonelli, F.; Vespignani, A.; Pentland, A.; Lepri, B. A multi-source dataset of urban life in the city of Milan and the Province of Trentino. *Sci. Data* **2015**, *2*, 1–15. [CrossRef] [PubMed]
43. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are transformers effective for time series forecasting? In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 11121–11128.
44. Serpush, F.; Rezaei, M. Complex human action recognition using a hierarchical feature reduction and deep learning-based method. *SN Comput. Sci.* **2021**, *2*, 94. [CrossRef]
45. Jin, W.; Lin, Y.; Wu, Z.; Wan, H. Spatio-Temporal Recurrent Convolutional Networks for Citywide Short-term Crowd Flows Prediction. In Proceedings of the ICCDA 2018, 2nd International Conference on Compute and Data Analysis, New York, NY, USA, 23–25 March 2018; pp. 28–35. [CrossRef]
46. Crawshaw, M. Multi-task learning with deep neural networks: A survey. *arXiv* **2020**, arXiv:2009.09796.