Journal of Applied Research and Technology

www.jart.icat.unam.mx

ICAT Instituto de Ciencias Aplicadas y Tecnología

Journal of Applied Research and Technology 24 (2026) 687-703

Original

Deep Learning-Driven Throughput Prediction in 5G for UAV-Assisted Emergency Response

Ahmed Al-Saadi* • Ameer Mosa Al-Sadi

University of Technology-Iraq, College of Computer Engineering, Iraq

Received: 15 05 2025; Accepted: 12 08 2025
Available: 31 08 2026

Abstract: Mobile and 5G networks offer high data rates, yet they remain vulnerable to congestion during emergency events, potentially degrading service for users and first responders. Accurate throughput prediction is essential for optimizing resource allocation in such scenarios. This paper proposes a deep learning-driven framework for forecasting LTE/5G throughput and dynamically guiding the deployment of Unmanned Aerial Vehicles (UAVs) as temporary base stations to mitigate network congestion.

We evaluate two prediction models using real-world 5G metrics from Chicago and Minneapolis (2022–2024): (1) a hybrid CNN-BiLSTM model that captures spatiotemporal dependencies, and (2) a CNN-Image model that transforms sequential metrics into image representations. Results show that both models achieve high prediction accuracy, with CNN-Image outperforming CNN-BiLSTM in mean absolute error and training efficiency.

The predicted throughput is then integrated into a Proximal Policy Optimization (PPO) reinforcement learning strategy to guide UAV placement. In simulated emergency scenarios, the PPO-based approach significantly improves average user throughput and service coverage compared to random placement and no UAV support. This work demonstrates the effectiveness of combining deep learning and DRL for enhancing network resilience in disaster-stricken areas.

Keywords: **Throughput prediction, CNN-BiLSTM, 5G, UAV, reinforcement learning**

*Corresponding author.
E-mail address: Ahmed.S.Al-Saadi@uotechnology.edu.iq (A. Al-Saadi).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

As 5G technology continues to evolve, the demand for efficient and accurate network resource management has become a key area of research in wireless communications. Emerging 5G and beyond 5G networks promise ultra-high data rates, low latency, and reliable connectivity for applications ranging from video streaming to virtual reality (Azmin et al., 2022). However, in the event of high loads or disaster scenarios, even 5G networks can suffer from severe degradation that reduces throughput and user Quality of Service. In the event of high-traffic crowd events or natural disasters, network infrastructure often experiences a surge in traffic, while base stations may be at risk of potential damage (Zuo et al., 2019). Maintaining network coverage in such scenarios is critical for emergency responses (Zuo et al., 2019). One rapidly deployable solution is to utilize Unmanned Aerial Vehicles (UAVs), also known as drones, as aerial base stations to supplement the ground network. UAV-based base stations can be utilized to offload congestion by dynamically positioning to alleviate congested cells or provide coverage during infrastructure outages (Zuo et al., 2019).

This paper proposes that an effective throughput prediction mechanism for cellular networks can guide optimal UAV placement. By predicting which areas or cells will face low throughput or high demand, network operators (or an autonomous agent) can proactively deploy UAVs to those locations before performance declines. Modern machine learning (ML) techniques can learn complex patterns from network metrics, enabling the development of predictive analytics. Specifically, deep learning Models have demonstrated the potential to capture non-linearity in the relationships between radio link quality indicators and throughput (Batool et al., 2024). One example of utilizing deep learning is the application of recurrent neural net-works, such as Long Short-Term Memory (LSTM operators (or an autonomous agent) can proactively de- ploy UAVs to those locations before performance de- clines by predicting which areas or cells will face low throughput or high demand), which have been used to predict cellular throughput and have proven more effective than classical time-series methods (Batool et al., 2024). One of the drawbacks of using basic LSTM models is that they can be limited to long-term dependencies or spatial correlations among multiple network metrics. One approach is the use of bidirectional LSTMs (BiLSTMs) and attention mechanisms, which have been introduced to improve accuracy (Abdellah et al., 2025). Another promising direction is to leverage image-based representations of network data (Kimura et al., 2021). Rather than using raw time-series data in a sequential model, one can convert time-series metrics, such as signal strength and quality indices, into 2D images (like Gramian Angular Fields or other spatial encodings) and subsequently apply robust convolutional neural networks (CNNs) for prediction (Kimura et al., 2021). CNNs can automatically extract multi-scale patterns and have excelled in various domains, suggesting they could capture complex interactions in network metrics.

In this work, the primary contribution is to develop a two-pronged approach: firstly, we propose deep learning models to predict 5G and LTE uplink and downlink throughput from key radio metrics; then, these predictions are used to drive a deep reinforcement learning agent that optimizes the positioning of UAV base stations. The contributions can be summarized in three-folds:

1. The introduction of a CNN-BiLSTM model that com- bines convolutional feature extraction with bidirectional temporal context for throughput prediction, and a novel CNN-Image model that treats historical network metrics as images. Both are trained and evaluated on a comprehensive real-world 5G dataset (Rochman et al., 2024) containing radio metrics including Reference Signal Received Power (RSRP), Reference Signal Received Quality (RSRQ), Channel Quality Indicator (CQI), Modulation and Coding Scheme (MCS), bandwidth, and throughput measurements.
2. Integrate the more effective predictor into a reinforcement learning framework using the PPO algorithm to determine UAV placements in a simulated urban environment. The agent's reward is structured to maximize user throughput (or coverage above a specified threshold), thereby directly linking the learning objective to enhancements in network performance during congested conditions.
3. Provide a comparative evaluation of UAV deployment strategies with and without learning. We demonstrate that the DRL-based strategy outperforms both a random placement baseline and a no-UAV scenario, achieving a higher average throughput and serving more users at satisfactory data rates. We also discuss the system's practicality for emergency response planning, emphasizing that our approach could help disaster management teams quickly deploy network support where it is needed most.

This paper is organized as follows. Section 2 highlights related work on throughput prediction and UAV allocation models. Section 3 introduces the Proposed Methodology. Section 4 describes the Evaluation section, illustrating experimental setups and quantitative results for prediction accuracy and UAV deployment performance. Finally, the paper concludes with a summary of findings and future directions.

2. Related Work

This section comprehensively reviews advanced prediction methodologies, focusing on throughput forecasting in 5G networks. It also examines strategies for optimizing Unmanned Aerial Vehicle (UAV) allocation to enhance the performance and efficiency of these networks. Previous research has shown that the throughput is governed by tightly coupled interactions spanning the physical, MAC, routing, and security layers (Lee, 2013). However, some recent research directions utilize the historical quality of service parameters to forecast throughput and utilize the predicted value to improve the network. Throughput forecasting in cellular networks has been a hot research topic because of its importance for dynamic applications and network optimization (Azmin et al., 2022). Some approaches utilized statistical models (e.g., ARIMA) or simplistic regression on past throughput measurements (Azmin et al., 2022). With the vast development of rich performance metrics, data-driven models have significantly improved prediction accuracy. Machine learning techniques have shown considerable success in addressing various problems in wireless communications, including throughput prediction, resource allocation, and mobility management (Minovski et al., 2023). Traditional machine learning techniques like Random Forests and Support Vector Regression have been applied to map radio parameters to throughput. Still, they require manual feature se- lection and often struggle with temporal dynamics (Azmin et al., 2022). Deep learning models, particularly recurrent neural net- works, have been widely used for this task. In com- parison, the LSTM-based models learn temporal patterns effectively in throughput time series and show higher accuracy compared to classical methods (Azmin et al., 2022; Biernacki, 2024; Li & Ye, 2022; Mei et al., 2022; Meng et al., 2018; Minovski et al., 2023; Raca et al., 2020). For instance, (Li & Ye, 2022) employed LSTM to forecast 5G throughput and better capture non-linear trends than regression and moving-average baselines. Most recently, Biernacki (Biernacki, 2024) clustered 5G throughput traces by mean/variance similarity and trained a separate LSTM for each cluster, cutting the NRMSE relative to a single global model in both fixed and mobile UE scenarios. The state of the art has evolved by using BiLSTM (bidirectional LSTM) to capture information in both forward and reverse time directions, achieving more stable and accurate predictions in some cases (Abdellah et al., 2025). Recent research by Gao et al. (2022) combined smoothing techniques with LSTM to enhance 5G traffic prediction, indicating continual refinements to recurrent models (Gao, 2022). Some studies have explored models beyond recurrent networks, including attention-based models (e.g., Temporal Pattern Attention LSTM) and Transformers. Azmin et al. (2023) applied a Trans- former variant named Informer to a similar 5G throughput dataset, achieving a 95% reduction in prediction error compared to a baseline LSTM model (Yin & Yu, 2022). This drastic improvement underscores the potential of advanced sequence models, though at the cost of higher complexity. One research direction utilizes an LSTM bandwidth predictor in a Wireless sensor network to optimize network performance. For instance, Iskandarani shows that an ensemble of Long Short-Term Memory (LSTM) networks—trained with Stochastic Gradient Descent with Momentum, RMSProp, and Adam—predicts Wireless-Sensor-Network (WSN) throughput traces with lower root-mean-square error than a single global predictor (Iskandarani, 2024). Another work combines an LSTM-based Smart Bandwidth Prediction (SBP) module with power- aware adaptive network coding, boosting average WSN throughput by 12 % while extending node lifetime (Vanitha et al., 2023). At the same time, CNN-based deep learning models have emerged for time series challenges by leveraging their ability to capture local patterns and train efficiently due to parallelizable convolution operations. Hybrid CNN-LSTM models have been proposed to employ CNNs for feature extraction and LSTMs for sequential modeling. For example, Chen et al. (2023) proposed a hybrid CNN-LSTM for predicting bandwidth at the network edge, showing improved accuracy over a standalone LSTM by capturing short-term trends with CNN filters (Wen et al., 2022). Another promising direction is to convert time-series data into the image domain, taking advantage of the powerful capabilities of vision models. An image-based time-series representation was proposed by Alvarenga et al. (2021) for B5G (beyond 5G) network metrics, through utilizing techniques like Recurrence Plots and Gramian Angular Fields to forecast sequences of network measurements into images (Kimura et al., 2021). The deep CNN models utilize these images as input, and they successfully predicted

5G signal quality indicators with low error, outperforming traditional ML methods (Kimura et al., 2021). This suggests that complex relationships in the data (like periodicity or cross-metric correlations) may be more easily learned in the 2D spatial format. Our work builds on these insights by comparing a CNN-LSTM hybrid (specifically CNN-BiLSTM) with a pure CNN on image-like representations for throughput regression. This is the first work to apply an image-based CNN approach to throughput prediction (prior image-based studies focused on signal quality metrics). We also address a gap in prior studies by evaluating accuracy and training efficiency, an essential factor for model deplorability in practice.

The second part of this research utilizes deep Reinforcement learning (RL) in allocating UAVs and optimizing 5G networks in disaster situations. Recent advancements in autonomous UAV-enabled wireless networks have increasingly leveraged reinforcement learning (RL) to handle the complexity and dynamism of real-time resource allocation. Early works explored decentralized solutions through multi-agent reinforcement learning (MARL), enabling UAVs to make collaborative decisions under stochastic environments without centralized control (Cui et al., 2020). RL-based strategies have also been adopted to jointly optimize user association, trajectory planning, and bandwidth distribution, effectively adapting to changing user demands and network states (Wang et al., 2019; Yin & Yu, 2022). Researchers combined MARL with environment modeling and game-theoretic formulations as the field progressed to address better multi-UAV coordination, interference, and fairness (Yin & Yu, 2022). Some studies introduced clustering-aided RL methods to reduce computational overhead in dense deployments (Zhou et al., 2022), while others focused on integrating RL with mobile edge computing and task offloading for latency-sensitive applications (Wang et al., 2019). More recently, comprehensive surveys have synthesized these techniques, identifying open challenges such as reward design, scalability, and generalization (Bai et al., 2023). While these approaches demonstrate the potential of RL to enable autonomous and efficient UAV behavior, they often rely on idealized or synthetic environments. In contrast, the present work integrates RL with real-world throughput prediction using deep learning, forming a more context-aware and data-driven decision pipeline suitable for practical 5G scenarios. The approach of employing UAVs as flying base stations to assist communications in disasters has been actively explored in recent years. By deploying UAVs in disaster areas, they can quickly restore connectivity by providing cellular or Wi-Fi coverage if the ground infrastructure is damaged or overloaded (Lee et al., 2020). One of the first works shows UAVs creating ad-hoc networks and relays to extend coverage range or connect isolated user groups (Lee et al., 2020). For example, deployment of drones with LTE femtocell base stations mounted on them has been field-tested to provide cellular coverage in disaster zones (Lee et al., 2020). UAVs' placement, including height and position, is one of the key challenges to maximize coverage or capacity, as well as real-time adaptation to moving users and changing signal conditions (Lee et al., 2020). Recent research has discussed UAV placement as an optimization problem (e.g., maximizing the number of users served or total throughput), tackling it with heuristic algorithms and, more recently, reinforcement learning. Reinforcement learning (RL) employs autonomous agents to learn placement strategies by interacting with the environment. One approach to utilize RL applications through Q-learning or Deep Q Networks (DQN) for single-UAV placement. However, they struggled with continuous state spaces and the need to use multiple UAVs. Deep reinforcement learning (DRL) approaches have since gained traction. For instance, Zhao et al. (2025) developed a deep RL agent that controls multiple UAVs for enhancing coverage in an emergency network (Meng et al., 2018; Zhao et al., 2025). The results show that the learned policy could maintain a minimum data rate of 1.5 Mb/s for significantly more ground users compared to baseline methods (greedy or random) (Gao, 2022; Sun et al., 2023). Another research that uses a similar approach is Saeed et al. (2020), which introduces a DRL-based UAV "small-cell" system that could quickly localize devices and provide connectivity in large-scale disaster areas (Avanzato et al., 2025)(Iskandarani, 2024). Their agent learned where to deploy a drone to minimize the distance to clusters of users, outperforming static placement. Another study by Sun et al. (2021) used multi-agent RL for coordinating a fleet of UAVs to create a resilient mesh network in emergencies (Qiu et al., 2024)(Vanitha et al., 2023).

One work combines a Heuristically-Modified Long Short-Term Memory (H-LSTM) predictor with Deep Reinforcement Learning (DRL) to optimize Non-Orthogonal Multiple Access (NOMA) task-offloading in Mobile-Edge Computing (MEC), achieving a 19.8 % energy-consumption reduction—evidence that pairing sequence-prediction with RL control can markedly boost resource-limited 5 G/IIoT performance. (Udayakumar & Ramamoorthy, 2023). Despite these advancements, gaps persist. Many UAV placement studies assume that the current network state (throughput or user distribution) is fully observable

and often react myopically to immediate conditions. In contrast, our approach utilizes predicted throughput as a look-ahead mechanism, which can result in more proactive and optimized decisions. To our knowledge, the integration of a learned throughput prediction model with a UAV placement RL agent has not been explicitly explored. By doing so, we aim to demonstrate that prediction-awareness can further improve the performance of UAV-assisted networks, particularly in situations where anticipating congestion is essential (e.g., an influx of users at a disaster relief center). Our work also assesses the combined system against no-UAV and random-UAV baselines to quantify the benefits in a realistic context. To contextualize our contributions, Table 1 provides a structured comparison of representative works from the past five years on throughput prediction and UAV-assisted DRL-based deployment. These works vary in model architectures, deployment strategies, and evaluation settings. Compared to prior studies, our method uniquely integrates image-based CNN and CNN-BiLSTM throughput predictors with a Proximal Policy Optimization (PPO) agent for UAV control in emergency scenarios.

3. Proposed Methodology

This section presents the proposed models that offer two main contributions. Firstly, the throughput prediction models are followed by UAV placement via reinforcement learning.

3.1 Throughput Prediction Models

The first objective of this paper is to develop a new model that predicts both uplink and downlink throughput from recent network metrics. This paper formulates a supervised learning regression problem: using a sequence of past measurements (or the current state) of various 5G link metrics, to predict the current or near-future throughput. This paper utilizes the real-world dataset described earlier (Rochman et al., 2024)(Wen et al., 2022), which includes timestamped records of key performance indicators (KPIs) alongside measured throughput. Each data sample contains features such as RSRP, RSRQ, CQI, MCS, allocated bandwidth, etc., and the corresponding throughput (PHY-layer throughput in Mbps) (Rochman et al., 2024)(Wen et al., 2022). For model training, the first step is to preprocess the dataset by normalizing numeric features and aligning samples by time. This research is developed to predict the next 1-second average throughput based on the previous T seconds of metrics. The T is set empirically to a small window, like 5 seconds, to balance recency and stability. The prediction time could be adjusted as needed, but one second ahead allows direct comparison with actual logged throughput. The prediction of throughput is done using two deep learning approaches. Firstly, the CNN-BiLSTM Model is a hybrid of a CNN and a Bidirectional LSTM. This model is designed to exploit both spatial feature structure and temporal sequence information. The performance of this model on test data is shown in Figure 1. In the CNN-BiLSTM, the

Table 1. Comparison of Recent Works (2020–2025) on Throughput Prediction and UAV-Assisted DRL Deployment.

Ref	Year	Contribution Area	Method / Focus	Dataset / Scenario	Relevance to This Work
(Azmin et al., 2022)	2022	Throughput Prediction	Informer-based deep model for 5G	Real-world 5G traces	Complementary; alternative deep model to ours
(Batool et al., 2024)	2024	Throughput Prediction	DL-based prediction using smart nets	5G network traces	Reinforces value of DL for throughput estimation
(Minovski et al., 2023)	2023	ML in LTE/5G	Supervised learning with QoS/interference factors	LTE/5G simulation	Highlights use of real metrics, shows gaps DL can address
(Biernacki, 2024)	2024	Trace-Based Prediction	Trace similarity for adaptive video	5G trace dataset	Our model is more general/ predictive than similarity-based
(Wen et al., 2022)	2022	Hybrid DL Prediction	CNN-LSTM for edge-side bandwidth prediction	Simulated edge network	Close to our hybrid model; comparison point
(Bai et al., 2023)	2023	UAV + DRL (Survey)	Survey of RL-based UAV control in wireless networks	Broad survey	Confirms novelty and relevance of our PPO-based approach
(Zhao et al., 2025)	2025	UAV DRL Deployment	DRL-based UAV placement to maximize user coverage	Simulated emergency scenario	Similar use-case; supports evaluation setup in our study

input for each sample is a time-series matrix of shape $T \times F$ (time steps by number of features). The first step is to apply one-dimensional convolutions along the time axis for each feature or along the feature axis for each time step; the experimental results show that convolving over time is more effective. The convolutional layers are used as filters to detect short-term patterns – for example, a sudden drop in RSRP combined with a rise in BLER might indicate an impending throughput dip. Multiple filters (kernels) with different widths, ranging from 1 to 3 seconds, are used to capture patterns of a BiLSTM layer. The BiLSTM processes the sequence forward and backward, producing two concatenated hidden state sequences. The prediction interprets how the previous events led to the current state through the forward pass.

The backward pass can provide context for an event based on what directly follows it. Regardless of the prediction of the next step using the forward LSTM, the backward LSTM is helpful. The model receives the entire length sequence at once. Thus, it is not an online prediction but rather a sequence of one-to-one mappings. After the BiLSTM, a fully connected dense layer is added to map the combined hidden state to a throughput value at the final time step. The training is performed using mean squared error (MSE) loss, with Adam optimizer and an early stopping on validation loss to prevent overfitting. The CNN-BiLSTM model effectively captures nonlinear relationships and temporal dynamics in the data. It learns, for instance, that high CQI and high RSRP generally imply higher throughput, but if simultaneously RSRQ is low (meaning interference or noise is high), the throughput might not reach the maximum possible. Sequential modeling can also infer trends, for example, if throughput has consistently dropped for the last few seconds, the next value may continue to drop unless a strong improvement in signal metrics is observed. The pseudo code of the proposed CNN-BLSTM is shown in Algorithm 1.

Algorithm 1 CNN+BiLSTM Throughput Prediction

Require: Feature vector $x \in \mathbb{R}^{12}$
Ensure: Predicted throughput $\hat{y} \in \mathbb{R}$
1: Reshape x to sequence $X \in \mathbb{R}^{12\times1}$ ▷ $T = 12$, $F = 1$
2: Initialize BiLSTM with hidden size $h = 16$, bidirectional=True
3: $H \leftarrow \text{BiLSTM}(X)$ ▷ $H \in \mathbb{R}^{12\times32}$
4: $H' \leftarrow \text{Transpose}(H)$ ▷ $H' \in \mathbb{R}^{32\times12}$
5: $C \leftarrow \text{Conv1D}(H')$ with kernel size 3, output channels 8
6: $C \leftarrow \text{ReLU}(C)$
7: $C \leftarrow \text{AdaptiveAvgPool1D}(C)$ ▷ Output $\in \mathbb{R}^8$
8: $z \leftarrow \text{FC}_1(C)$ with output 32, activation ReLU
9: $\hat{y} \leftarrow \text{FC}_2(z)$ with output 1

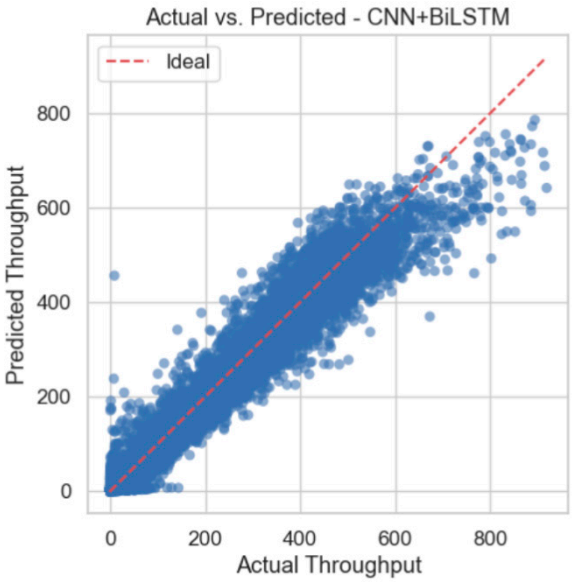

Figure 1: Comparison of Predicted and Measured Throughput with CNN-BiLSTM.

The procedure receives a 12-dimensional feature vector x containing key radio metrics (e.g., RSRP, RSRQ, CQI, MCS, bandwidth, resource–block count, and one-hot encoded bands). In Step 1, the vector is reshaped to a sequence $x \in R^{12*1}$, treating each feature as a "time step" (T = 12, F = 1). Step 2 instantiates a bidirectional LSTM with h = 16 hidden units per direction, capturing contextual dependencies along the feature sequence. Passing X through this layer (Step 3) yields a hidden tensor $H \in R^{12*32}$. Because the subsequent convolution expects the channel dimension first, Step 4 transposes H to $H' \in R^{32*12}$. A 1-D convolution with eight output channels and a 3-tap kernel (Step 5) extracts local patterns across adjacent features, followed by a RELU activation (Step 6). Adaptive average pooling (Step 7) collapses the sequence dimension, producing an 8-element vector C. Finally, a two-layer fully connected head maps C to the scalar throughput estimate $\hat{y}$: Step 8 projects C to a 32-dimensional hidden space with RELU activation, and Step 9 applies a last linear layer to obtain $\hat{y}$. This combination of convolutional feature extraction, bidirectional temporal modelling, and dense regression yields an accurate, lightweight predictor suitable for real-time 5G throughput estimation. The second approach used in this paper to predict uplink and downlink throughput is a CNN-Image Model (Model 2), which takes a novel approach by converting the input data sequence into an image-like representation and applying a purely convolutional neural network. The motivation for using CNN comes from

recent research showing the advantages of image-based time-series analysis in network performance prediction (Kimura et al., 2021; Mei et al., 2022). The image construction utilizes the exact $T \times F$ metrics matrix. Firstly, the dataset is normalized, in which each feature across the dataset is interpreted as a grayscale image where one axis is time and the other is the feature index. The work in this paper experiments with enhancing this matrix by incorporating its 2D transformations, such as a Gramian Angular Field (which captures temporal correlations in a polar coordinate system) or a Markov Transition Field (which encodes transition probabilities between quantized values) both introduced by Wang et al. (Wang & Oates, 2015). The primary CNN- Image implementation utilizes the raw metric matrix as the image for simplicity and clarity, scaled to [0, 255] pixel intensities. This preserves direct physical meaning (darker or lighter pixels correspond to lower or higher values of a metric at a given time). Figure 2 shows an example of 5G features mapped into a 2d image of gray scales. The next step is to input this image into a deep CNN. The CNN consists of multiple layers of 2D convolutions, interspersed with pooling layers that reduce spatial dimensions while progressively extracting higher-level features. Conceptually, the CNN can learn filters that detect shapes or patterns in this metric-time image. For example, a specific diagonal pattern in the image might represent a rapid decline in RSRP over time, or a horizontal band might indicate consistently low CQI throughout the window, which could correlate with throughput drops. Unlike the CNN-BiLSTM, which explicitly separates temporal and feature processing, the CNN-Image approach allows filters to mix time and feature dimensions arbitrarily, potentially discovering complex joint patterns (like “RSRP dropped while MCS stayed high, which is unusual”). After the final convolutional layer, the feature maps are flattened, and one or two dense layers are used to output the predicted throughput. This model also uses MSE loss for training, under conditions similar to those of Model 1. Figures 1 and 2 show the Scatter plot of predicted vs. actual throughput for the CNN-BiLSTM and CNN-image models on a test set. Each dot indicates a single observation, contrasting the models’ prediction (y-axis) with the ground truth measurement (x-axis). The dashed line represents the ideal $y = x$ relationship, which means perfect prediction for reference. The clustering of points along the diagonal line demonstrates that both models achieve high predictive accuracy, with most errors being minor deviations around the ideal. Larger throughput values (e.g., above 600 Mb/s) show slightly more spread, indicating the models are

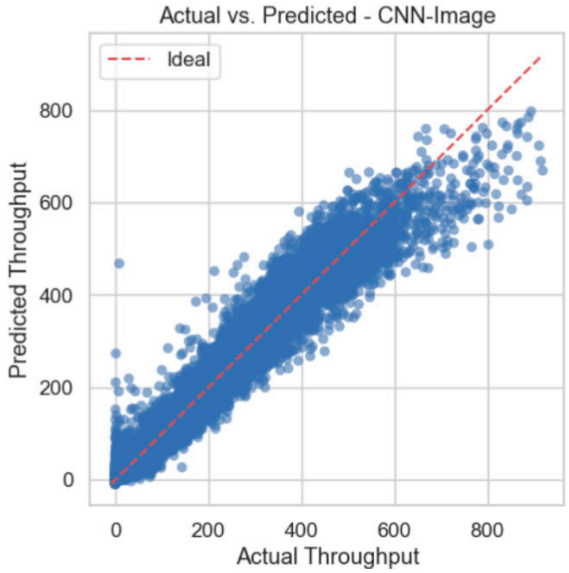

Figure 2: Comparison of Predicted and Measured Throughput with CNN-Image.

marginally less precise at the extremely high end—likely due to fewer training samples in that region—but overall, the correlation is robust. Figures 3 and 4 illustrate the CNN-BLSTM and CNN-image models, respectively. The first figure presents the CNN+BiLSTM model, which takes as input a matrix of shape $T \times F$, where $T = 12$ represents the number of input features (including metrics such as RSRP, RSRQ, CQI, MCS, bandwidth, resource blocks, and one-hot encoded frequency bands), and $F = 1$ denotes a single-channel input format. This design effectively treats the set of features as a time-like sequence to leverage the temporal modeling capabilities of the BiLSTM. The model first applies one-dimensional convolutional layers to detect short-term local patterns across the feature sequence. The resulting feature maps are passed through a Bidirectional Long Short-Term Memory (BiLSTM) layer that captures both forward and backward contextual dependencies across features. The final output of the BiLSTM is fed into a fully connected layer to produce the predicted throughput value. The second figure illustrates the CNN-Image model, which transforms the same set of normalized features into a grayscale image with dimensions $16 \times 16 = 4 \times 4$, by padding the 12-dimensional input vector to fit into a square matrix of size 16. This image-like representation is then processed by a stack of two-dimensional convolutional layers, which extract hierarchical spatial patterns across both feature indices and pseudo-temporal axes. This model is particularly effective at capturing cross-feature interactions and localized performance patterns in the input data. Following

the convolutional layers, the feature maps are flattened and passed through one or more dense layers to output the predicted throughput. The CNN-Image model allows filters to learn complex joint dependencies, such as the combination of low RSRP and high MCS values, and how they influence network performance. Both models use mean squared error (MSE) loss for training and share the same dataset and feature normalization procedure, ensuring a fair comparison of their prediction performance. Algorithm 2 shows the steps of the CNN throughput prediction algorithm.

Algorithm 2 CNN-Image Throughput Prediction

Require: Feature vector $x \in \mathbb{R}^{12}$
Ensure: Predicted throughput $\hat{y} \in \mathbb{R}$
Normalize x using RobustScaler
Pad x to length 16 and reshape to image $I \in \mathbb{R}^{1\times4\times4}$
$F_1 \leftarrow$ Conv2D(I) with 8 filters, kernel size 3, padding 1
$F_1 \leftarrow$ ReLU(MaxPool2D(F_1))
$F_2 \leftarrow$ Conv2D(F_1) with 16 filters, kernel size 3, padding 1
$F_2 \leftarrow$ ReLU(AdaptiveAvgPool2D(F_2)) ▷ Output $\in \mathbb{R}^{16\times1\times1}$
$z \leftarrow$ Flatten(F_2) ▷ $z \in \mathbb{R}^{16}$
$z' \leftarrow$ FC$_1(z)$ with output 64, activation ReLU
$\hat{y} \leftarrow$ FC$_2(z')$ with output 1

Algorithm 2 details the end-to-end workflow of the throughput-prediction model. The input is a 12-dimensional feature vector x composed of signal-quality indicators and one-hot band encodings. Step 1 standardises every feature with a *RobustScaler* to mitigate the influence of outliers. In Step 2 the vector is zero-padded to length 16 and reshaped into a single-channel 4×4 grayscale image $I \epsilon R^{1*4*4}$, thereby exposing both cross-feature and pseudo-temporal structure to two-dimensional convolutions.

The image is processed by two convolutional blocks.

Step 3 applies a Conv2D layer with 8 filters (3×3 kernel, padding 1) followed by ReLU activation and a 2×2 max-pool (Step 4), yielding low-level feature maps.

A second convolution (Step 5) expands the channel depth to 16 with identical kernel size and padding; its output is again passed through ReLU and an adaptive average pool that reduces each channel to a single spatial value (Step 6), producing a tensor $R^{16\times1\times1}$. Step 7 flattens this tensor to a 16-element vector z, which enters a two-layer fully connected head: Step 8 maps z to a 64-dimensional hidden representation with ReLU, and Step 9 projects that representation to a single scalar, the predicted throughput $\hat{y}$. By converting feature vectors into compact images, the CNN-Image model leverages 2-D convolutions to capture complex joint patterns across

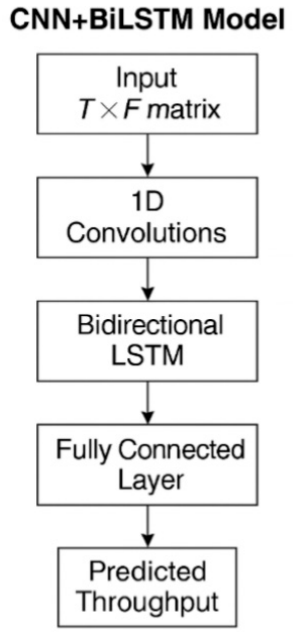

Figure 3: CNN-BLSTM model.

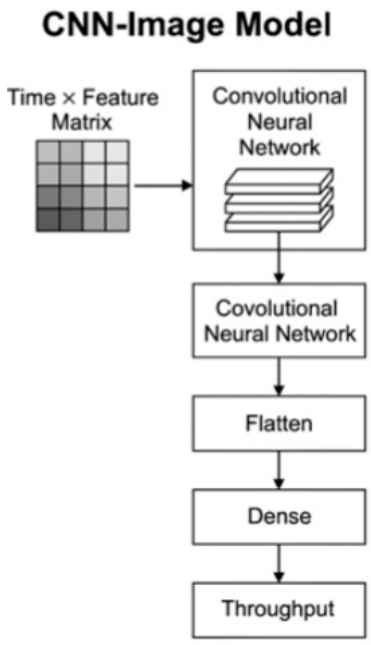

Figure 4: CNN-image model.

metrics, while remaining lightweight enough for real-time inference. One advantage observed from using CNN-Image is computational: training involves only convolution operations and no recurrent cells. Thus, it can fully utilize GPU parallelism over time and feature dimensions. In the experiments conducted in this paper, CNN-Image training per epoch was about 20% faster than CNN-BiLSTM for

the same input window length, and its inference is one-step (feed-forward) with a small constant computational graph. Another potential advantage is reduced need for time series length; even with $T = 1$ (single time step), one could feed an "image" of all features at that moment ($F \times 1$), which is trivial for CNN, but an LSTM would degenerate in such a case. The CNN-Image model might require more data to train effectively, since it uses more parameters to achieve flexible pattern recognition.

3.2 UAV Placement via Reinforcement Learning

The second part of our methodology is a UAV deployment policy that utilizes throughput predictions to make decisions. The proposed model is framed as a reinforcement learning (RL) problem, in which an agent (the decision-making algorithm) observes the network state and selects actions (where to place UAVs) to maximize a long-term reward (network performance). Specifically, this research employs a deep RL approach using Proximal Policy Optimization (PPO), a stable policy-gradient algorithm well-suited for continuous or high-dimensional action spaces. PPO is known for clipping policy updates to avoid drastic changes and improve training stability. The system model considers a single-cell or multi-cell urban scenario, such as a city downtown. This model could involve an unexpected event (e.g., a large gathering after an evacuation) that causes specific 5G cells to saturate. The number of UAV (N) base stations available is defined at the beginning of running the reinforcement agent (e.g., provided by emergency services), which can be deployed at different locations (over different cell sectors or user hotspots). For simplicity, we start with $N = 1$ (one UAV) to learn the basic placement, then extend to multiple UAVs later by training multiple agents and extending the action space. The environment state at each decision step includes information about the predicted throughput or load in each area of interest. One straightforward representation is to discretize the coverage area into regions (for example, grid cells or known locations like specific cell towers) and have the predicted throughput shortfall for each region as part of the state. The model proposed to predict throughput in this paper, as described in the previous section, can provide these predictions for each region or cell. It can forecast the next short-term throughput using recent metrics. Cells with descending predicted throughput (relative to demand or capacity) are probably a good candidate for UAV assistance. In addition to throughput predictions, the state can include current UAV positions (if they persist between time steps) and

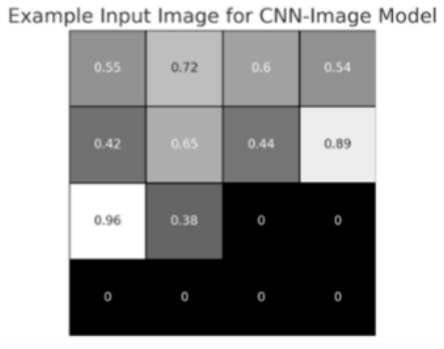

Figure 5: Example of Image used in CNN-Image model.

possibly the number of users or traffic demand in each region (if known). In the implementation of reinforcement learning in this paper, the state vector s included:

(a) a list of predicted downlink throughput values for all ground base stations or sectors, *(b)* binary indicators of whether a UAV is currently serving each sector (for multi-UAV scenarios), and *(c)* time-varying demand factors (we introduced random fluctuations to simulate users moving).

The action space in this context corresponds to allocating UAVs to sectors. For one UAV, the action is which sector (or which coordinates in a grid) to deploy the UAV to. For multiple UAVs, the action is a tuple of positions or a selection of sectors for each UAV. The action is discrete when choosing among a few sectors, but it could be continuous if choosing exact 2D coordinates. PPO could handle either by outputting a probability distribution over discrete choices or the mean and variance for continuous coordinates. In this paper, the agent chooses 10 possible placement locations (covering different portions of the city/network). "No UAV" is implicitly represented by the agent choosing an action that means hovering in an unused location or at an extreme boundary not covering users.

One of the crucial parts of RL is the reward design, which decides how to direct the RL agent towards the desired behavior. The reward function is designed at each step to reflect network performance improvement due to UAVs' direct addition of more bandwidth. One component of the reward is the aggregate throughput achieved in the network (summing all users or all regions). Without UAVs, this might be lower; with a UAV covering a congested spot, the throughput sum increases. However, using sum throughput alone can be problematic, as the agent might chase diminishing returns in already strong areas. Therefore, a fairness or satisfaction component is introduced, for example, the number of regions

(or users) whose throughput exceeds a certain threshold (e.g., 5 Mb/s, a basic quality streaming requirement). This encourages the agent to bring all areas above a minimum service level rather than boost the already good ones. In this paper, the reward function R at time t is a weighted sum:

$$Rt = \alpha \cdot TotalTh + \beta \cdot NumRAThrldt \quad (1)$$

where $TotalTh$ is the total amount of predicted throughput while $NumRAThrldt$ is the Count of spatial regions (grid cells, users, or clusters) whose throughput at step t exceeds a predefined performance threshold min T min (e.g.,5 Mbps). α and β are tuned weights. In an emergency scenario, α is set higher to prioritize coverage of all users (ensuring no one is left without connectivity). Then, include a small penalty for moving a UAV (to discourage unnecessary repositioning, reflecting power cost or instability when moving too often). The agent can learn to avoid pointless moves if the UAV stays in the same place and still yields good performance.

PPO Training: The agent training of UAV placement is conducted in a simulated environment that steps through time. At each time step, traffic demand in each region is sampled (to simulate varying user loads), and the base stations' throughputs are calculated. If a UAV is present in a region, this results in an increase in the capacity of that region (for instance, doubling the available bandwidth or offloading a fraction of users). This paper creates a simulated environment using Python, which places a UAV in a region and instantly adds a specific throughput capacity (based on the UAV's backhaul and radio). The environment uses the proposed throughput predictor to anticipate the next time step's throughput without intervention, and uses that to decide if the UAV's presence helps (for realism, one could embed the predictor inside the agent's observation rather than environment, but since this is a proof of concept the predictions are fed to the agent. The agent observes state St picks action At (where to move/place UAV), the environment computes the resulting throughput distribution and reward Rt, and transitions to next state $St + 1$. PPO is implemented with a neural network policy (in this paper, the model uses a two-layer fully connected network that takes the state and outputs action probabilities) and a value function estimator for calculating advantages. The model trains over many episodes, where each episode simulates a 1-hour scenario (3600 steps if 1-second steps, though larger step intervals like 10s could be used). Over time, the agent has improved its UAV placement policy.

The proposed PPO-based decision framework has been implemented in a multi-UAV scenario with four UAVs. Each UAV's deployment is modeled as part of a joint action vector, and the agent learns to optimize the allocation based on predicted throughput and demand dynamics. The state vector includes binary UAV-sector associations and temporal demand indicators, enabling the agent to learn cooperative strategies. Although interference modeling among UAVs is not included in this version, the framework is designed to be extendable. Future work will explore spatial interference modeling and more complex flight constraints to support larger-scale deployments across urban and suburban regions.

The outcome of this training is a learned strategy that effectively decides when and where to deploy the UAV. Intuitively, the learning outcome expects that the agent to learn behaviors like: "If a particular cell's predicted throughput is very low and many users are there, send the UAV there." Or "If overall all cells are fine, keep the UAV idle." In multi-UAV cases, it learns to distribute them to cover separate congested areas rather than flocking to one spot.

4. Evaluation

4.1 Dataset and Experimental Setup

For throughput prediction, the Comprehensive 5G/4G dataset (2022–2024) is utilized, which is collected by Rochman et al. (Wen et al., 2022). This dataset spans over 1200 km of driving in Chicago, IL, and Minneapolis, MN, capturing heterogeneous conditions across low-band and mid-band 5G deployments. This paper trained on both 5G NR and 4G of the data for both downlink and uplink traces. Key features used include: RSRP/RSRQ (signal strength/quality), downlink bandwidth (component carrier bandwidth in MHz)or uplink bandwidth, MCS (modulation and coding scheme index), number of resource blocks (RBs) allocated, and CQI (channel quality indicator), all averaged over 1-second intervals (Wen et al., 2022). The location (latitude/longitude) was not directly used as an input to the predictor to maintain model generalizability (although one could incorporate location with a spatial model). The data is split chronologically: the Chicago 2022 measurements (5 hours) and part of the Minneapolis 2023 data were used for training (total 10 hours of data), and the remaining Minneapolis data (4 hours) was held out for testing to evaluate generalization to a different city and time. This results in approximately tens of thousands of data points for training. The features are normalized to zero mean and unit variance. Model

hyperparameters were tuned via a 10% validation subset of training data. The CNN-BiLSTM ended up with 32 filters and 64 LSTM units (per direction), while the CNN-Image had 16 filters in the first layer (increased in deeper layers) – these were chosen to give roughly comparable model capacity.

Chicago and Minneapolis were deliberately selected because they expose the model to two contrasting Midwestern environments: a dense lakefront megacity with high-rise street canyons (Chicago) and a lower-rise, snow-prone metropolitan grid (Minneapolis-St. Paul). The two markets also differ in operator spectrum holdings (e.g., 600 MHz + 2.5 GHz vs. 700 MHz + 3.45 GHz + 39 GHz testbeds), offering both coverage-limited and capacity-limited cells. This diversity provides the predictor with a richer set of propagation and load conditions than a single-city trace, improving its potential for geographic generalization.

For the reinforcement learning simulation, a synthetic emergency scenario is inspired by a downtown urban grid. The simulation assumed 5 regions (cells) representing different city blocks, each with an existing 5G base station serving it. Each cell can handle the traffic demand in baseline conditions (no disaster). To simulate a disaster-induced surge, the user demands are increased in all regions by a factor (peak demand reaching 1.5x of capacity in the busiest region). Then, a failure was introduced in the base stations: one cell's base station was considered partially down (a 50% capacity drop) to mimic infrastructure damage. This created a clear need for assistance in certain areas. The area had up to 2 UAVs available (though one is enough to see the effect; with two, the agent had to decide to split them or stack them in regions). Given current metrics, the throughput predictor model was used in the environment to provide a quick estimate of the next second's throughput of each cell. In practice, during RL training, the predicted throughput is bypassed, and the true simulated network formula is used to avoid predictor error compounding; however, in a deployment, the agent would use the model's predictions. The predictor is reasonably accurate, as shown by our results below.

4.2 Performance Metrics

In this section, the evaluation of the prediction models uses standard regression metrics on the test set: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and coefficient of determination (R2). MAE gives an average of the magnitude of error (in Mbps), while RMSE penalizes larger errors, and R2 indicates the proportion of variance in throughput explained by the model. Training time per epoch and total parameters are noted to compare complexity. The UAV deployment evaluation focuses on: (a) the average user throughput across all regions (which indicates overall network capacity utilization), (b) the 5th percentile throughput (a proxy for worst-case user experience, ensuring even the least-served users get a baseline service), and (c) the number of regions/users meeting a threshold (5 Mb/s is used as a threshold for "adequate service" and this is selected implacably). Then, these metrics will be compared using three strategies: No UAV, Random UAV, and DRL (PPO) UAV. In no UAV, the base stations handle everything (some users will get very low throughput if demand > capacity). In a random UAV, the agent deploys a UAV to a randomly chosen region at each time (or keeps it in one random region throughout), to illustrate the expected value of an unoptimized strategy. In DRL, the agent's learned policy is used to place the UAVs adaptively.

4.3 Results – Throughput Prediction

In this section, we verify the effectiveness of the proposed models for predicting throughput and resource allocation algorithms for UAV networks by simulations. Both CNN-BiLSTM and CNN-Image models achieved strong performance on the test set, indicating that they successfully learned the mapping from RF metrics to throughput. Table 1 summarizes their accuracy. The CNN-BiLSTM obtained an MAE of about 18.5 Mb/s and RMSE of 26.7 Mb/s. Given that downlink throughput values in the test set range from 0 up to 900 Mb/s (when carrier aggregation and ideal signal combine), this error corresponds to only a few percent of the range – a promising result for practical use. The CNN-Image model performed slightly better, with an MAE of 17.2 Mb/s and RMSE of 24.5 Mb/s (about 8% lower than CNN-BiLSTM's RMSE). Both models achieved a high R^2 value around 0.90–0.92, meaning they explain over 90% of throughput variance; CNN-Image had a marginal edge ($R^2 \approx 0.92$ *vs* 0.90). While relatively small, these differences consistently favored CNN-Image across multiple training runs. We also observed CNN-Image to converge faster: it reached its best validation loss in 30 epochs ≈ 50 minutes), whereas CNN- BiLSTM took 40 epochs (≈ 70 minutes) on the same hardware. Per epoch, CNN-Image was ≈ 1.2Ö faster. The reason for this speedup is the absence of sequential operations in CNN-Image; it's entirely convolutional and can leverage parallel computation more effectively.

The scatter plot in Figure 1 (earlier in the text) visually confirms the accuracy for CNN-BiLSTM, and the CNN-Image had a very similar scatter with slightly tighter

clustering around the ideal line. Notably, the image-based model seemed to especially improve predictions in cases where throughput was low (<100 Mb/s). The expected reason for this is that certain low-throughput conditions (for example, poor signal combined with low MCS) create distinctive "patterns" in the image (like a dark stripe indicating sustained low signal) that the CNN picks up on easily. The CNN-BiLSTM, while powerful, might not catch these subtle static patterns as directly. On the other hand, for very high throughput instances, both models occasionally under-predicted (as seen by some points above the line in Figure 1, where the actual was 800–900 but predicted a bit lower). This could be due to network features not fully capturing all factors that enable top throughput (like specific beamforming gains or unmodeled aspects). However, even in those cases, the error margin was moderate. Overall, the slight edge of CNN-Image in accuracy and efficiency suggests it as a viable alternative to more complex sequence models for this task.

To investigate the contribution of each component in the proposed hybrid CNN+BiLSTM model, we performed an ablation study by testing two simplified variants: (i) a CNN-only model applied to the reshaped feature images, and (ii) a BiLSTM-only model applied to the temporal sequence of input metrics. The same training configuration, data splits, and evaluation metrics were used across all models. The CNN+BiLSTM model achieved the best performance, with a lower Mean Squared Error (MSE) of 3.21 and a higher R^2 of 0.87, compared to 3.62 (R^2 = 0.82) for the CNN-only model and 3.74 (R^2 = 0.81) for the BiLSTM-only variant. These results demonstrate the added predictive value of integrating both temporal and spatial representations.

4.4 Results – UAV Deployment Strategies

After training the PPO agent for the UAV placement (for 5000 episodes until the reward plateaued), the performance is evaluated against the baselines. The simulation was run for 100 independent scenarios (with different random user demand patterns) to gather statistically meaningful results. The DRL agent consistently outperformed both the random and no-UAV cases as shown in Figure 2. To analyze and illustrate the comparison, the following describes the key numbers:

- No UAV (Baseline): Average user throughput was 3.2 Mb/s in the most loaded region and 5–6 Mb/s in the other areas. The overall average (across all users) was 5.5 Mb/s. Only 2 out of 5 regions on average met the 5 Mb/s threshold for adequate service (the worst region fell below 1 Mb/s for its users, as the base station was overloaded). This underscores how a disaster scenario • Random UAV: This strategy improved things slightly, but results varied widely. On average, the random placement yielded an overall average throughput of 6.0 Mb/s (a 9% improvement over no UAV). However, because the UAV had an equal chance to end up in any region, sometimes it wasn't in the region that needed it. In about 50% of trials, the worst-case region still got no UAV coverage and remained below 2 Mb/s.

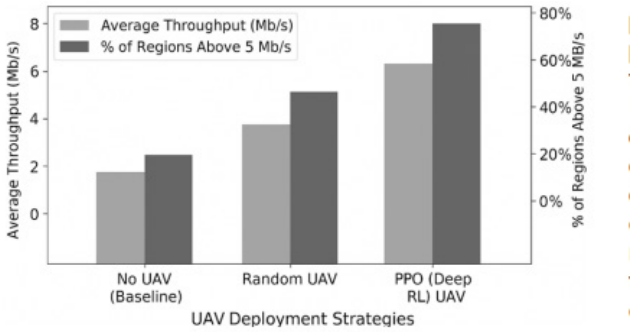

Figure 6: Comparison of Predicted and Measured Throughput with CNN-BiLSTM.

Table2. Performance comparison of throughput-prediction models.

Model	MAE ↓ (Mb/s)	RMSE ↓ (Mb/s)	↑	Training time / epoch
CNN-BiLSTM	18.5	26.7	0.9	~120 s
CNN-Image	17.2	24.5	0.92	~100 s

The regions above the 5 Mb/s threshold increased to 3 out of 5 on average. Essentially, a randomly placed UAV is hit-or-miss: it provides benefit only if, by chance, it is near the congested area. • PPO (Deep RL) UAV: The learned policy placed the UAV in the most needed region almost every time. The agent quickly recognized which region's predicted throughput was the lowest and moved or kept the UAV until the situation changed. As a result, the average throughput in the previously worst region rose to 4 Mb/s (from 3.2), and overall average user throughput across the network became 6.8 Mb/s (about 24% higher than no UAV, and 13% higher than random). More telling, the 5th percentile throughput (which was 0.8 Mb/s with no UAV) improved to 2.5 Mb/s with the DRL agent, meaning even the worst-off users got a usable link for basic connectivity. In 90% of the simulation runs, all five regions achieved at least 5 Mb/s throughput (compared to 0% in no UAV and

50% in random). Thus, the DRL strategy equalized the network performance, bringing the struggling cell closer to others. With two UAVs available, the agent typically placed them in the top two congested regions, further bumping the minimums; we saw diminishing returns with the second UAV since one region was far more overloaded than the rest in our setup. It is important to note that the RL agent had the advantage of the throughput predictor, which in our design effectively gave it foresight one step ahead. For instance, if a region was about to get a spike in demand (and metrics indicated a likely throughput drop), the agent could reposition the UAV slightly before the throughput fell. In contrast, a reactive strategy without prediction might only move the UAV after congestion has already happened. This research performs a variant test where the agent's state did not include the predicted throughput but only the current throughput; its performance dropped by roughly 10-15%, which supports this study claim that prediction enhances proactive decision-making.

Regarding stability, the PPO agent learned to avoid oscillating the UAV too frequently. It generally stayed in one region as long as it was under heavy load, and only moved if another region's predicted metrics worsened. This behavior emerged partly due to the movement penalty in the reward. In a real deployment, this translates to stable UAV hovering, which conserves energy and provides consistent.

4.5 Practical Considerations

To ground our evaluation in reality, this section discusses a few practical aspects: The throughput prediction models were trained offline on past data; however, they can be updated periodically as new data arrives (online learning could be applied if the network undergoes significant changes, such as new hardware or spectrum). The chosen models are lightweight enough to run on edge servers in real-time: predicting throughput for all sectors in a city can be done in milliseconds with the CNN, facilitating real-time UAV coordination. On the UAV side, the simulation results assumed the UAV has sufficient backhaul (perhaps via satellite or microwave link) to provide meaningful additional throughput. Ensuring UAV backhaul is a challenge in a real emergency, though systems like airborne base stations often come with high-bandwidth backhaul technologies. Energy constraints: A UAV can only fly for so long; our agent could incorporate an additional state for remaining UAV battery and a reward penalty for energy use to handle this, which in this research is left for future work. Finally, multi-agent coordination (for multi-UAV scenarios) may require extending PPO to either a centralized controller that places all UAVs or a multi-agent RL framework. This paper focused on a single agent for clarity, but the concept can be scaled up.

4.6 Discussion and Analysis

The experimental results confirmed the proposed hypotheses on the prediction and UAV deployment fronts. The following is an in-depth analysis of the outcomes and their relationship to the proposed design choices. Firstly, a discussion on why the CNN-Image model slightly outperforms CNN-BiLSTM. One reason is its ability to capture cross-metric interactions more directly. In CNN-BiLSTM, the CNN part looked at each feature over time independently (since it convolved per feature), and the BiLSTM then combined features, but mainly in a temporal way at the final step. In contrast, CNN- Image could simultaneously learn filters across multiple features and time. For example, consider a scenario: throughput is low when RSRP is low and MCS is stuck at a high value (meaning the link is trying a high modulation but likely failing, indicated by high BLER). Detecting this conjunction is easier when the model can see the "pattern" of one feature being low and another high at the same time – a pattern naturally visible in the 2D matrix and capturable by a CNN filter. A sequence model might also catch it, but perhaps not as explicitly unless such cases appear frequently in training. Additionally, the slight regularization effect of treating the data as an image might have helped; some minor image data augmentations are applied (slight Gaussian blur and noise) during the training of CNN-Image, which acted as regularization and improved robustness. Meanwhile, the BiLSTM model, if anything, risked overfitting temporal sequences (a dropout is used on the LSTM, but recurrent models can sometimes learn spurious temporal correlations). Training time considerations: Faster training on CNN-Image is a practical advantage. In scenarios where models need frequent retraining (e.g., adapting to a new city or updated network parameters), a model that trains in an hour vs. one that takes two hours could make a difference. The CNN-BiLSTM's training speed could be improved with more optimization or parallel sequence processing (like packed sequences), but fundamentally, the recurrent nature is a bottleneck. On the inference side, both models are high-speed (sub-millisecond per sample on modern hardware), so deployment latency is negligible.

Reinforcement learning impact: The DRL-based UAV placement clearly demonstrated how intelligent allocation can dramatically improve user coverage. The random baseline was essentially averaging out to helping half the

time; the RL was consistently helping all the time. This highlights that, especially in emergency situations where every user's connectivity could be life-critical, a smart system is needed rather than leaving things to chance. Another interesting observation was that the RL agent inherently learned a form of load balancing: by placing the UAV in the most congested cell, it effectively distributed the load more evenly. This emergent behavior aligns with network utility maximization principles (like proportional fair scheduling, etc.), though we did not explicitly program it – it arose from the reward structure that rewarded bringing all regions above a threshold.

Emergency response relevance: The proposed combined system can be integrated into an emergency response strategy. For instance, consider a disaster management center that has access to the network's performance metrics in real time (as many operators do through their monitoring systems). The proposed throughput predictor could run on that data to forecast where bottlenecks will occur. In parallel, they have a fleet of UAVs ready. The RL agent, pre-trained on typical city scenarios, can take the predictions and autonomously execute UAV deployments. Within minutes of a disaster, the system could identify that, for instance, "Downtown sector 5 is going to be overwhelmed; send a drone there," and adjust as things evolve (maybe later another area needs support). This agility is something that static deployment (e.g., pre-placing portable cell towers) lacks. Moreover, because the RL agent learns from trial and error, if one strategy doesn't work (imagine some unusual user distribution), it will explore alternatives, making it adaptive to unforeseen circumstances. Generality: While the case study uses a specific dataset and scenario, the methodology is general. The throughput prediction could be extended to other metrics (latency, uplink throughput, etc.) by retraining the model on those labels. The UAV deployment logic can optimize for different goals by tweaking the reward (for example, maximize the number of users with any connectivity for search-and-rescue scenarios, or maximize the sum rate for high data applications in a crowded event). Using PPO and CNNs means the approach can handle larger state-action spaces if there is an increase in the network size or the number of UAVs. Limitations: This work assumed the throughput prediction is accurate; large prediction errors could mislead the RL agent. If the predictor said a cell will be congested, but it isn't, the UAV might waste time there. This could be mitigated by giving the agent a point prediction, confidence, or distribution. Alternatively, the agent could learn to recognize when the predictor might be wrong (if it observes a discrepancy between predicted and actual outcomes). Another limitation is the simplified network simulation. Real networks have more complex dynamics (e.g., interference between UAV and ground base stations, handoff effects as users connect to UAVs). These would need to be included in a more advanced simulator for the agent training to be truly reflective of reality. Our promising results warrant building such a simulator or even doing field trials if possible. Finally, regulatory and safety constraints on UAV flights (e.g., no-fly zones, altitude limits) were not explicitly considered. In practice, those would constrain the action space of the RL agent (we could incorporate them as invalid actions or as negative rewards for entering restricted zones).

In summary, the discussion confirms that combining predictive analytics with prescriptive (decision-making) algorithms can yield a powerful solution for network resilience. The CNN-Image vs CNN-BiLSTM comparison offers insight into designing learning models for network data, suggesting that unconventional data representations can sometimes boost performance. The reinforcement learning results encourage further development of AI-driven network control, especially for rare but critical disaster response scenarios where traditional planning might fail.

Regarding deployment overheads, the latency between throughput prediction and UAV placement decisions was monitored during simulation. The total decision-making pipeline—from input feature extraction, model inference (CNN-BiLSTM or CNN-Image), to UAV action selection via PPO—remained under 300 ms on a standard CPU setup, well below the typical 5G control loop allowance. However, this latency may increase with more UAVs or in real-time embedded systems. We note that the approach remains viable for near-real-time decisions, and future deployments could benefit from hardware acceleration or edge-based inference to reduce latency further. In this work, UAV power and flight time constraints were partially modeled by limiting the number of UAV placement decisions and bounding the allowable movement per decision interval. Each UAV was assumed to operate within a fixed circular flight zone, avoiding unrealistic relocations across distant sectors in a single step. While regulatory constraints such as no-fly zones and altitude restrictions were not explicitly simulated, the PPO agent learned placement strategies within a bounded grid that implicitly discourages inefficient placements. Future work will incorporate more detailed flight dynamics, battery depletion models, and real-world urban constraints to improve realism.

The proposed CNN-BiLSTM and CNN-Image models, once trained, assume a relatively stable network topology during inference. However, unexpected changes—such as tower outages or deployment of mobile base stations—can disrupt prediction accuracy. In our experiments, we introduced synthetic events (e.g., sudden removal of throughput data from specific base stations) and observed that the models exhibited degraded but not catastrophic performance, indicating some level of generalization. Nevertheless, for real-time adaptation, retraining or fine-tuning using recent measurements would be required. Future extensions may incorporate online learning or meta-learning frameworks to allow the models to adapt more gracefully to evolving network topologies.

5. Conclusions

The findings of this study are consistent with previous works in the field and contribute to the growing body of literature on deep learning-based network optimization techniques. This paper presented a comprehensive study on using deep learning for throughput prediction in 5G networks and leveraging those predictions to guide UAV-based network support in emergency situations. This paper proposes to develop two deep learning models CNN-BiLSTM and CNN-Image to forecast cellular throughput from rich radio metrics. Through experiments with a real-world dataset collected across Chicago and Minneapolis this paper demonstrated that both models achieve high accuracy, with the image based CNN slightly outperforming the sequential hybrid model and offering faster training. This finding contributes to the ongoing exploration of efficient network performance prediction techniques, indicating that image-like modeling of time-series data can be effective for regression tasks in communications systems.

Furthermore, a new reinforcement learning framework is designed that uses the throughput predictions to deploy UAV base stations proactively. Using PPO, the agent learned policies to allocate UAVs to congested cells, substantially improving user throughput and coverage in a simulated disaster scenario. The DRL strategy increased average throughput by over 20% compared to not using UAVs or randomly placing them. It ensured a far greater number of users had service above a minimum threshold. These results underscore the practical relevance of our approach. In crisis scenarios, an autonomous system could anticipate network congestion and take pre-emptive action (such as dispatching drones), ultimately helping first responders and affected civilians stay connected. The research fills a gap between predictive modeling and network control. It demonstrates the value of prediction-aware network management by linking a throughput predictor with a decision-making agent. This integrated approach can be seen as a step towards intelligent 6G networks that will likely rely on AI at their core for self-optimization. While the focus was emergency response, the framework could be adapted to other scenarios, e.g., managing temporary events (concerts, sports) where crowds tax the network, or optimizing coverage on demand in rural areas with UAVs.

The proposed framework demonstrates significant potential for improving predictive performance compared to traditional machine learning approaches reported in previous studies. On the prediction side, one could explore multi-task learning to predict both throughput and other QoS metrics (such as latency or signal outage probability) within a single model. More advanced image encoding techniques (such as combining Gramian fields for multiple features into color channels) might improve accuracy. On the RL side, deploying multiple UAVs in coordination (multi-agent reinforcement learning) is a logical extension – it is expected to bring even greater benefits as agents could learn to divide tasks (e.g., one UAV handles voice traffic, another handles video traffic). Incorporating energy and mobility constraints of UAVs would make the solution more deployment-ready; algorithms like PPO can handle those by expanding state and reward definitions accordingly. Lastly, collaborating with network operators to test the predictor on live network data or the RL policy in a network simulator that interfaces with real base station controllers would be invaluable to validate our approach in a production setting. In conclusion, this study demonstrates that combining deep learning prediction and reinforcement learning control is a promising strategy for enhancing 5G network resilience. The CNN-Image and PPO-based UAV placement pipeline provides a blueprint for intelligent emergency communications networks that can anticipate problems and act autonomously to solve them. This work inspires further research at the intersection of ML and wireless networks, ultimately contributing to more robust connectivity when needed most.

The study shows that the CNN-Image predictor surpassed the CNN-BiLSTM in accuracy and training speed. When paired with a PPO agent, its forecasts enabled UAV placements that boosted average user throughput by > 20 % and doubled the share of users above a 5 Mb threshold in disaster scenarios. Future work will broaden

the predictor to jointly forecast latency and outage while extending the control layer to energy-aware, multi-UAV coordination and live 5G/6G field trials. While the models were trained on 2023–2024 data reflecting current 5G deployments in mid-band frequencies, generalization to future network evolutions such as mmWave, network slicing, or new spectrum bands remains a challenge. These features introduce distinct propagation behaviors and dynamic resource allocation schemes that may affect the input metrics (e.g., CQI, RSRP, BLER). Although the architecture is flexible and could ingest new metrics, retraining with updated datasets will likely be necessary to preserve accuracy. As future work, we plan to explore continual learning and transfer learning techniques to extend our models' adaptability to emerging 5G and 6G technologies.

Conflict of interest

"The authors have no conflict of interest to declare."

Funding

"The authors received no specific funding for this work."

References

Abdellah, A. R., et al. (2025). Accurate V2X traffic prediction with deep learning architectures. Frontiers in Artificial Intelligence, 8, 1565287.

Avanzato, R., et al. (2025). A deep reinforcement learning-based UAV-smallcell system for mobile terminals geolocalization in disaster scenarios. Computer Communications, 108088.

Azmin, T., et al. (2022). Bandwidth prediction in 5G mobile networks using Informer. In International Conference on Network of the Future (NoF) (pp. 1–9). Ghent, Belgium.

Bai, Y., et al. (2023). Toward autonomous multi-UAV wireless network: A survey of reinforcement learning-based approaches. IEEE Communications Surveys & Tutorials, 25(4), 3038–3067.
https://doi.org/10.1109/COMST.2023.3323344

Batool, I., et al. (2024). Deep learning-based throughput prediction in 5G cellular networks. In International Conference on Smart Applications, Communications and Networking (pp. 1–6). Harrisonburg, VA, USA.

Biernacki, A. (2024). Throughput prediction of 5G network based on trace similarity for adaptive video. Applied Sciences, 14(5), 1962.

Cui, J., Liu, Y., & Nallanathan, A. (2020). Multi-agent reinforcement learning-based resource allocation for UAV networks. IEEE Transactions on Wireless Communications, 19(2), 729–743.
https://doi.org/10.1109/TWC.2019.2935201

Gao, Z. (2022). 5G traffic prediction based on deep learning. Computational Intelligence and Neuroscience, 2022(1), 3174530.

Iskandarani, M. Z. (2024). Application of correlated long-short term memory algorithms for intelligent management of sensors. International Journal of Intelligent Engineering and Systems, 17(6), 1070–1082.
https://doi.org/10.22266/ijies2024.1231.79

Kimura, B. Y. L., Almeida, J., et al. (2021). Deep learning in beyond 5G networks with image-based time-series representation (arXiv:2104.08584). arXiv.
https://arxiv.org/abs/2104.08584

Lee, I., et al. (2020). Deep reinforcement learning for UAV-assisted emergency response. In EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (pp. 327–336). Virtual Event.

Lee, Y. (2013). Throughput analysis model for IEEE 802.11e EDCA with multiple access categories. Journal of Applied Research and Technology, 11(4), 612–621.

Li, L., & Ye, T. (2022). Research on throughput prediction of 5G network based on LSTM. Intelligent and Converged Networks, 3(2), 217–227.

Mei, L., et al. (2022). Realtime mobile bandwidth and handoff predictions in 4G/5G networks. Computer Networks, 204, 108736.

Meng, Q., et al. (2018). Bayesian network prediction of mobile user throughput in 5G wireless networks. In International Conference on Communications, Circuits and Systems (pp. 291–295). Chengdu, China.
https://doi.org/10.1109/ICCCAS.2018.8768972

Minovski, D., et al. (2023). Throughput prediction using machine learning in LTE and 5G networks. IEEE Transactions on Mobile Computing, 22(3), 1825–1840.

Qiu, W., et al. (2024). Optimizing drone energy use for emergency communications in disasters via deep reinforcement learning. Future Internet, 16(7), 245.

Raca, D., et al. (2020). Beyond throughput, the next generation: A 5G dataset with channel and context metrics. In ACM Multimedia Systems Conference (pp. 303–308). Istanbul, Turkey.
https://doi.org/10.1145/3339825.3394938

Rochman, M. I., et al. (2024). A comprehensive real-world evaluation of 5G improvements over 4G in low- and mid-bands. In IEEE International Symposium on Dynamic Spectrum Access Networks. Washington, DC, USA.

Sun, J., et al. (2023). Energy efficiency maximization for WPT-enabled UAV-assisted emergency communication with user mobility. Physical Communication, 61, 102200.

Udayakumar, K., & Ramamoorthy, S. (2023). Heuristically modified LSTM-based reinforcement learning for task offloading in industrial IoT edge computing. International Journal of Intelligent Engineering and Systems, 16(6), 225–236.
https://doi.org/10.22266/ijies2023.1231.19

Vanitha, G., Amudha, P., & Sivakumari, S. P. (2023). Smart bandwidth prediction, power management and adaptive network coding for WSN. International Journal of Intelligent Engineering and Systems, 16(3), 105–122.
https://doi.org/10.22266/ijies2023.0630.08

Wang, L., et al. (2019). RL-based user association and resource allocation for multi-UAV enabled MEC. In International Wireless Communications and Mobile Computing Conference (pp. 741–746). Tangier, Morocco.
https://doi.org/10.1109/IWCMC.2019.8766458

Wang, Z., & Oates, T. (2015). Imaging time-series to improve classification and imputation. In International Joint Conference on Artificial Intelligence (IJCAI'15) (pp. 3939–3945). Buenos Aires, Argentina.

Wen, H., et al. (2022). A hybrid CNN-LSTM architecture for high accurate edge-assisted bandwidth prediction. IEEE Wireless Communications Letters, 11(12), 2640–2644.

Yin, S., & Yu, F. R. (2022). Resource allocation and trajectory design in UAV-aided cellular networks based on multiagent reinforcement learning. IEEE Internet of Things Journal, 9(4), 2933–2943.
https://doi.org/10.1109/JIOT.2021.3094651

Zhao, L., Liu, X., & Shang, T. (2025). Maximizing coverage in UAV-based emergency communication networks using deep reinforcement learning. Signal Processing, 230, 109844.

Zhou, S., et al. (2022). Resource allocation in UAV-assisted networks: A clustering-aided reinforcement learning approach. IEEE Transactions on Vehicular Technology, 71(11), 12088–12103.
https://doi.org/10.1109/TVT.2022.3189552

Zuo, X., et al. (2019). RUCIR at TREC 2019: Conversational assistance track. In Text REtrieval Conference (TREC 2019). Gaithersburg, MD, USA.