Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Optimizing multi-access edge computing deployment in urban areas using lstm-based vehicle density detection

Mohamed Ben Bezziane[1], Adel Zga[2], Farouq Zitouni[3], Dallal Djoudi[4], Oussama Aiadi[5], and Yacine Khaldi[6]
[1,2,3,4,5,6] *Linati laboratory, Computer Science and Information Technologies Department, University of Kasdi Merbah, Route Ghardaia, Ouargla 30000, Algeria*
[6] *Ouargla Higher Normal School, Mathematics department, Algeria*
Email: benbezziane.mohamed@univ-ouargla.dz

ARTICLE INFO

Received: 29 Dec 2024

Revised: 15 Feb 2025

Accepted: 24 Feb 2025

ABSTRACT

5G technology offers unprecedented high speeds and low latency, enabling next-generation wireless networks. However, deploying Multi-access Edge Computing (MEC) at antenna sites remains a critical challenge, especially in densely populated urban areas. MEC is essential for delivering real-time services by positioning computing resources close to end users. This study investigates employing the Long Short-Term Memory (LSTM), a deep learning model, for detecting and predicting vehicle trajectories as well as vehicle density to improve urban transportation systems and optimize MEC placement. Using the Cabspotting dataset, which provides GPS co-ordinates of taxis in San Francisco, the data was converted into a time-series format to predict vehicle locations. The LSTM model demonstrated superior prediction accuracy compared to traditional Recurrent Neural Networks (RNNs). To further refine the results, the K-Means algorithm clustered the detected and predicted vehicle positions, identifying optimal zones based on vehicle density for MEC deployment. These findings underscore the potential of LSTM-based vehicle density detection and predictions to enhance strategic MEC placement, advancing smart city infrastructure and sup-porting the rollout of 5G technology.

Keywords: MEC, 5G, LSTM, Cabspotting, and K-Means.

INTRODUCTION

The advancement of the 5G technology serves as a great leap for the modernization of mobile communication networks with immense promise of powerful data transfer rates, ultra-low latency, and interconnectivity [1]. Having these promises in mind, it is expected that with the advancement of the 5G networks, it will offer completely new opportunities for creative inventions like self-driving vehicles, live video broadcasting, and futuristic industrial robots [2]. At the same time, the rollout of the 5G technology comes with a number of issues along with it such as the deployment of the network in a city location which has a plenty of structures and towers required to support the hefty requirements of ultrahigh data transfer [3]. One of the important drivers of these 5G services is Multi-access Edge Computing (MEC) which is the solution to the demand of bringing computational resources to the users [4]. The use of MEC devices close to the 5G antennas makes it feasible to perform local data pro-cessing which minimizes the communication lag and improves the responsiveness of real time systems [5].

The means of bringing MEC into 5G networks is particularly important in smart cities, where the processing of data and decision taking is necessary to support a variety of intelligent transportation systems [6]. This is because cities are expanding and the number of connected devices are increasing which means there is a need to predict vehicular movements and optimize the positioning of MEC units strategically. A well-coordinated placement of an MEC unit makes sense only when advanced predictive modelling vis-a-vis vehicle trajectories is possible [7, 8]. Thus, effective SI is an underlying requirement.

Because of its capacity to utilize the temporal structure of sequential data, deep learning, especially Recurrent Neural Networks (RNNs), has made great strides in time series forecasting [9, 10]. A crippled RNN with a vanishing gradient issue, on the other hand, hinders the RNN's overall performance regarding long term predictions. Long short-term memory networks (LSTMs), a more sophisticated type of RNN, use gating mechanisms that are capable of overcoming these problems and are successful at long-range sequential tasks [11,12]. As an example, we will use the Cabspotting dataset, which is an extensive collection of GPS coordinates for taxis in San Francisco [13]. It will allow us to model vehicle movements and test our predictive model.

The K-Means clustering method is further utilized to cluster the predicted vehicle positions which in turn helps to analyse the predicted vehicle positions [14]. The predicted locations of vehicles are clustered which makes it easy to find the places with high number of vehicles which are convenient for the location of the MEC units. This makes it easy for the network operators to position the MEC units in those zones making the communication latency low and increasing the performance of the 5G network. Not only does this result in better services for end users but also aids in the smart city objectives of managing traffic, lower pollution, and improved public safety.

The key contributions of this work can be summarized as follows:

1. We demonstrate the effectiveness of LSTM networks in accurately predicting and detecting vehicle trajectories, achieving higher accuracy compared to simpler RNN models.

2. We present a method to utilize these trajectory predictions to guide the placement of MEC units in a smart city context.

3. Clustering techniques are employed to optimize the MEC deployment strategy, ensuring efficient and strategic placement.

Our findings show that integrating LSTM-based trajectory predictions with MEC placement could noticeably improve the performance of 5G networks, paving the way for more resilient and efficient urban infrastructure.

This paper is structured as follows: the Literature Review gives a wide view of the methodologies so far existing for time-series forecasting in sequential data, showing the strengths and weaknesses of the current techniques; and the Methods section develops the proposed solution in detail, including the use of LSTM networks for trajectory prediction and K-Means clustering for optimal MEC placement. In the Results section, performance metrics for all predictive models that show how the proposed method has outperformed previous approaches will be presented. This paper discusses all the important conclusions, further discussing practical insights about how to im-prove MEC deployment over 5G networks, as well as pointing out some research lines that could be considered to further optimize the strategic placement of 5G antennas.

STATE OF THE ART

The rapid advancement of 5G technology and the increasing demand for low-latency applications have brought Multi-access Edge Computing (MEC) to the forefront of wireless networking [15]. Efficient MEC deployment and optimization are crucial for maximizing the benefits of 5G networks, especially in urban areas with high user density and diverse application requirements. Several studies have explored the use of predictive modelling to optimize MEC deployment strategies. Researchers have investigated using machine learning algorithms to forecast user mobility patterns and predict traffic demands, enabling proactive allocation of MEC resources [16].

Deep learning techniques, in particular Recurrent Neural Networks (RNNs) and their variants, have achieved noticeable success in time-series forecasting tasks. For example, a study by Lin et al. in [17] proposed an innovative content caching strategy for Multi-access Edge Computing (MEC) in 5G/6G IoT networks. The strategy aims to maximize the cache hit ratio by enabling dynamic forecasting in the dynamically changing network and user environments. The system employs an LSTM-based local learning model with seasonal-trend decomposition for accurate demand prediction. Additionally, it integrates an ensemble-based meta-learning model to consolidate user preferences into a unified caching strategy. This method improves the cache hit ratio by up to 30% compared to traditional algorithms and performs within approximately 9% of the ideal caching strategy, which relies on perfect foresight of content popularity. The ability of RNNs to effectively capture temporal dependencies in sequential data makes them not only ideal for content caching but also for applications such as predicting vehicle trajectories to inform MEC placement decisions.

The authors in [18] propose a hierarchical deep learning architecture for content caching in edge computing to satisfy the demands of 5G/6G IoT applications. The system uses an LSTM-based local learning model with seasonal-trend decomposition for demand prediction, and an ensemble-based meta-learning model for orchestrating user preferences into a unified caching strategy. Experiments on MovieLens datasets show up to a 30 \% improvement in cache hit ratio compared to traditional algorithms.

In [19], the authors introduce a content caching strategy for MEC in 5G/6G IoT networks, aiming to enhance the cache hit ratio by leveraging adaptable predictions in evolving network and user conditions. Their approach is built on a hierarchical deep learning framework. This research presents a content caching method for MEC in 5G/6G IoT networks, designed to optimize the cache hit ratio by employing adaptive predictions that adjust to dynamic network and user environments. The system uses an LSTM-based local learning model with seasonal-trend decomposition for demand pre-diction, and an ensemble-based meta-learning model for orchestrating user preferences into a unified caching strategy. The proposed approach enhances the cache hit ratio by up to 30% compared to conventional algorithms and achieves near-optimal performance, coming within approximately 9% of an ideal caching strategy that assumes perfect prior knowledge of content popularity. Utilizing local learning techniques alongside ensemble-based meta-learning, this method significantly improves caching efficiency. Furthermore, it has the potential to be integrated as a core function within the Network Data Analytics Function (NWDAF) module of 5G and future 6G networks.

The research in [20] introduces a novel content caching strategy for MEC in 5G/6G IoT networks, aiming to enhance the cache hit ratio by leveraging adaptive pre-diction techniques in dynamic network and user environments. This approach employs an LSTM-based local learning model with seasonal-trend decomposition for demand forecasting, while an ensemble-based meta-learning model integrates user preferences into a cohesive caching strategy. The proposed method achieves up to a 30% improvement in cache hit ratio compared to conventional algorithms and attains near-optimal performance, approaching within approximately 9% of an ideal caching scheme with perfect knowledge of content popularity.

Similarly, Tran et al. in [21] present a content caching strategy for MEC in 5G/6G IoT networks designed to optimize the cache hit ratio by utilizing predictive analytics in fluctuating network and user conditions. Their system also incorporates an LSTM-based local learning model with seasonal-trend decomposition for demand estimation and employs an ensemble-based meta-learning model to harmonize user preferences into a unified caching mechanism. The proposed solution enhances cache efficiency by up to 30% over traditional methods and achieves a near-optimal cache hit ratio, approximating 9% of the theoretical maximum with complete prior knowledge of content demand. In [22], the authors explore the use of Recurrent Neural Networks (RNNs) for accurately predicting user mobility in automotive scenarios, enabling efficient management of distributed MEC resources. Their study identifies an optimal LSTM-based RNN configuration that delivers highly precise mobility predictions. To validate its effectiveness, they implement an experimental decision algorithm that evaluates the allocation of distributed resources, balancing service scaling and migration decisions while ensuring mobile users receive a satisfactory quality of ser-vice.

Brik et al. focus on optimizing MEC resource allocation for collision avoidance systems in vehicular networks. Their approach leverages deep learning techniques to predict vehicle density and dynamically allocate computing resources. However, their research is confined to simulated environments and does not account for real-world traffic conditions [23]. A comparative analysis of related works and our contributions is presented in Table 1.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 1. Comparative analysis of related works and our contributions.

Study	Objective	Technique	Dataset /Environment	Key Findings
Lin et al. [17]	Content caching in MEC for 5G/6G IoT networks	LSTM with seasonal-trend decomposition; Ensemble-based meta-learning	MovieLens dataset	Achieved up to a 30% im-provement in cache hit ratio compared to conventional algorithms.
Authors in [18]	Hierarchical deep learning for content caching	LSTM-based local learning; Meta-learning	Simulation environment	Demonstrated hierarchical architecture improves content caching efficiency by capturing temporal and user preferences.
Study in [19]	Flexible predic-tion for dynamic environments in 5G/6G MEC	LSTM; Ensemble-based strategies	Simulated IoT networks	Improved cache hit ratio by 30% and achieved near-optimal performance with 9% margin from theoretical maximum.
Tran et al. [21]	Dynamic content caching in MEC	LSTM; Ensemble learning	Simulated IoT environments	Highlighted adaptive caching strategies leveraging deep learning models for user demand prediction.
Study in [22]	Accurate user mobility prediction for MEC in automotive scenarios	RNN with LSTM	Automotive simulation	Found LSTM-based RNN effec-tive for mobility prediction, enabling balanced MEC resource utilization.
Brik et al. [23]	MEC resource optimization for collision avoidance systems in vehicular net-works	Deep learning for vehicle density predic-tion	Simulated vehicular networks	Optimized resource allocation but limited to simulations, not real-world traffic data.
Our Work	Predicting and detecting vehicle trajectories to optimize MEC placement in 5G-enabled smart cities	LSTM for trajectory prediction; K-Means clustering for MEC placement	Cabspotting dataset (real-world GPS data from San Francisco)	First to combine LSTM-based trajectory predictions with K-Means for MEC placement in real-world scenarios. Introduced trajectory clustering to identify high-traffic zones for MEC deployment, addressing urban transportation needs.

PROPOSED FRAMEWORK

3.1. MEC-DEPL framework

This section introduces the architecture of our proposed framework, Multi-access Edge Computing Deployment (MEC-DEPL), which operates in two phases. The first phase leverages a Long Short-Term Memory (LSTM) model to predict vehicle trajectories based on time-series data derived from latitude and longitude coordinates. In the second phase, the predicted positions are clustered to determine optimal locations for 5G antennas hosting MEC servers. The Cabspotting dataset was pre-processed and normalized to ensure accuracy in trajectory prediction and clustering. The frame-work's outcomes include effective model training, trajectory prediction, and the strategic deployment of MEC servers to enhance network performance.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

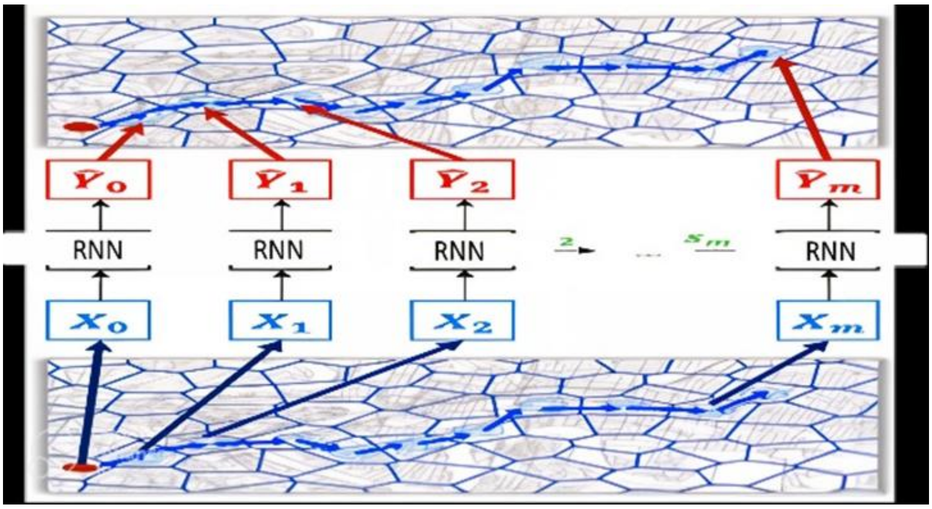

Figure 1. LSTM-based trajectory prediction for optimized modelling.

As shown in Fig. 1, the predictive modelling of vehicle trajectories using an LSTM model effectively transforms raw GPS data into a time-series format, enabling accurate trajectory predictions across San Francisco's urban environment. This data is further utilized in Fig. 2, where the map of San Francisco highlights strategic locations for 5G antennas hosting MEC servers. These locations are identified through clustering predicted vehicle densities, and optimizing MEC infrastructure deployment to enhance 5G network performance.

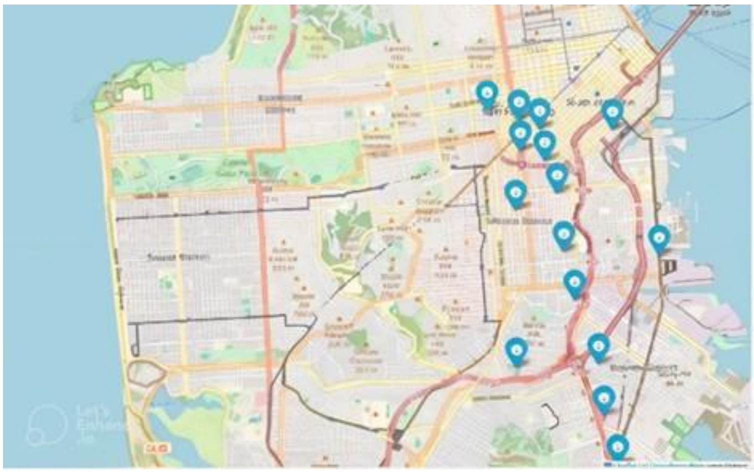

Figure 2. Strategic deployment of 5G antennas with MEC servers in San Francisco city.

3.2. Dataset and preprocessing

Cabspotting was one of the earliest projects that the San Francisco Exploratorium had been working on, in collaboration with Stamen Design-which is one of the earliest applications of tracking data in real time-and showing how the movements of taxis, using GPS, have been moving around San Francisco. This project ran from September 29, 2006 through June 30, 2007, in an effort to illustrate how the flow and patterns of activity occurred around the city. For this data science challenge, we used the Cabspotting Dataset, which contains mobility traces of about 500 taxis in San Francisco over a duration of 30 days. The dataset consists of 537 text files, each representing a single taxi, with a total of 11,220,490 records. Each file represents the GPS movement data for a single taxi. A snippet of the data from one such file is given in Fig. 3.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

37.61632	-122.38803	1	1213034957
37.6158	-122.38463	1	1213034899
37.61639	-122.38593	0	1213034560
37.61574	-122.38801	0	1213034500
37.61393	-122.39732	0	1213034439

Figure 3. Cab spotting dataset.

Columns are, respectively:

● Latitude

● Longitude

● Occupancy (1 for passengers, 0 for empty)

● Time (number of seconds since the Unix epoch: 00:00:00 UTC)

The initial step in preparing the dataset involved combining all 536 text files, each repre-senting the GPS mobility traces of a unique cab, into a single comprehensive CSV file. This con-solidation aimed to streamline the processing and analysis of the dataset. To better identify indi-vidual cabs, a new column, "CarID," was added, assigning unique identifiers ranging from 1 to 536. Simultaneously, the "Occupancy" column, which indicated passenger presence, was removed as it was deemed unnecessary for this study's objectives. Furthermore, the timestamps, originally rec-orded in UNIX epoch format, were transformed into a standard human-readable time format. This transformation, illustrated in Fig. 4, significantly enhanced the interpretability of the dataset, making it more accessible for further analysis.

Index	Timestamp	CarID	Latitude	Longitude
0	2008-05-17 15:10:22	1	37.75050	-122.42086
1	2008-05-17 15:51:01	1	37.73494	-122.40670
2	2008-05-17 16:30:44	1	37.78931	-122.42200
3	2008-05-17 17:13:08	1	37.61393	-122.39731
4	2008-05-17 17:51:38	1	37.76632	-122.42462

Figure 4. Sample of processed GPS mobility dataset with CarID and Human-Readable Timestamps.

To reduce the dataset size while maintaining representative data, 5000 positions were ran-domly selected for each cab. This selection was essential for optimizing computational efficiency during model training. Additionally, the latitude and longitude coordinates were normalized, cantering the data around a mean value. Normalization improved the data's consistency and fa-cilitated better convergence of the deep learning model during training. To convert the dataset into a format suitable for time series analysis, a sliding window of size three was employed. This ap-proach generated input sequences (denoted as X) and corresponding labels (denoted as Y), enabling the prediction of subsequent positions. The dataset was then split into training and testing sets using an 80:20 ratio. This division ensured the model was trained on a considerable amount of the data while letting a separate set as unseen data for the evaluation.

EXPERIMENTAL ANALYSIS

The proposed LSTM-based model of the MEC-DEPL is designed to predict vehicle trajectories efficiently, leveraging the strengths of sequential data modelling to optimize the placement of MEC units in urban environments, especially in San Francisco city. Its architecture emphasizes accurate temporal data analysis and robust performance in real-world applications.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

4.1. Model architecture and key components for efficient trajectory prediction

An in-depth exploration of the LSTM-based framework was conducted, with its architecture comprising several key components. The input layers include time-series GPS data, which are first normalized to promote convergence and maintain consistent scaling. In the feature processing stage, Long Short-Term Memory (LSTM) layers are employed to capture long-term dependencies in sequential vehicle movement data, while a dropout technique is applied to mitigate overfitting by randomly disabling certain neurons during training. The outputs from the LSTM layers are then passed through fully connected (dense) layers to extract meaningful features, culminating in an output layer that uses a linear activation function to generate continuous values representing predicted vehicle coordinates. Finally, the model is trained using the Mean Squared Error (MSE) loss function to minimize prediction errors, and optimized with adaptive algorithms such as Adam to ensure efficient and effective convergence.

This technique effectively utilizes LSTM's ability to retain temporal context in sequential data, crucial for accurate trajectory prediction. The model's predictions inform the strategic deployment of MEC units, improving data throughput and reducing latency in urban networks. By clustering predicted positions, it further enables efficient MEC placement, addressing network congestion and supporting smart city infrastructure.

4.2. The model architecture along with the training phase

The proposed LSTM's architecture is detailed in Table 2. The model consists of three primary layers: an LSTM layer, followed by two dense layers. The LSTM layer outputs a shape of (None, 100) with 41,200 trainable parameters, capturing temporal dependencies in the input data. The first dense layer, comprising 20,200 trainable parameters, increases the feature dimensionality to 200. Finally, the second dense layer maps the features to an output of shape (None, 2) with 402 parameters, enabling classification into two categories.

Table 2. The proposed LSTM model architecture.

Layer (type)	Output Shape	Parameter
lstm (Lstm)	(None,100)	41200
dense (Dense)	(None,200)	20200
dense_1 (Dense)	(None,2)	402

The proposed model has a total of 61,802 parameters, which corresponds to approximately 241.41 KB of memory. Notably, all parameters in the model are trainable, as there are no non-trainable parameters (0.00 bytes). This configuration ensures that the entire parameter set contributes to optimizing the model during the training process. The model was trained over 20 epochs to optimize its performance. During training, both the training and validation losses were monitored to assess the model's learning progression and generalization ability. This process ensured the minimization of overfitting while enhancing the model's accuracy in predicting outcomes based on the given data.

4.3. The model architecture along with the training phase

To assess the performance of the LSTM model, we used two of the most common metrics related to a loss function. The loss function, also called a cost or objective function, essentially calculates the difference between the values predicted by a deep learning model and their target values. The primary objective during training is to minimize this loss, with the model getting closer to correctly predicting the target value. It involves the proper selection of the loss function according to the problem's nature-for instance, regression or classification problems. In the case of regression problems, commonly used loss functions include Mean Squared Error (MSE) and Mean Absolute Error (MAE). For classification tasks, however, Cross-Entropy Loss is often the preferred choice.

a. Mean Squared Error (MSE): this metric (also known as L2 loss) is one of the most adopted loss functions when handling regression tasks. It computes the error by squaring the difference between the model outputs and the true values for all the samples in the dataset.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

MSE is calculated by the equation 1:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \qquad (1)$$

Where:

- n is the number of data points.

- y_i is the actual value of the dependent variable for the ith observation.

- $\tilde{y}_i$ is the predicted value of the dependent variable for the ith observation.

b. Mean Absolute Error (MAE): this metric (also known as L1 loss) is among the most straightforward and interpretable loss function. It is based on calculating the difference between the true and the predicted values for the whole dataset. Formally, it represents the arithmetic mean of absolute errors, concentrating on their magnitude, regardless of direction. A low MAE score mentions a superior model performance.

MAE is calculated by the equation 2:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\tilde{y}_i - y_i| \qquad (2)$$

Where:

- y_i is actual value.

- $\tilde{y}_i$ is predicted value.

- n is the sample size

In our system, we have achieved:

- The loss obtained during evaluation is 0.001814185525290668.

- The Mean Squared Error (MSE) calculated separately is 0.0018141850778880752.

- The Mean Absolute Error (MAE) calculated separately is 0.026066203018259404.

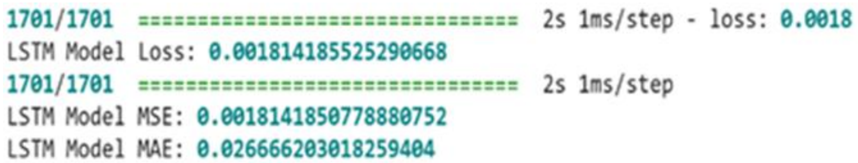
```
1701/1701  ==================================  2s 1ms/step - loss: 0.0018
LSTM Model Loss: 0.001814185525290668
1701/1701  ==================================  2s 1ms/step
LSTM Model MSE: 0.0018141850778880752
LSTM Model MAE: 0.026666203018259404
```

Figure 5. Evaluation Metrics for the LSTM Model: Loss, MSE, and MAE.

For the test set. Fig 5, clearly shows the obtained results. These metrics provide insights into the performance of the LSTM model in predicting the target values. The low values of MSE and MAE indicate that the model performs well in predicting the trajectories of the vehicles. Note that the effectiveness of our model has been demonstrated, we will examine some of its outcomes to assess its accuracy. Fig 6 below presents the training and validation losses, which were plotted to visualize the model's performance across the epochs.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

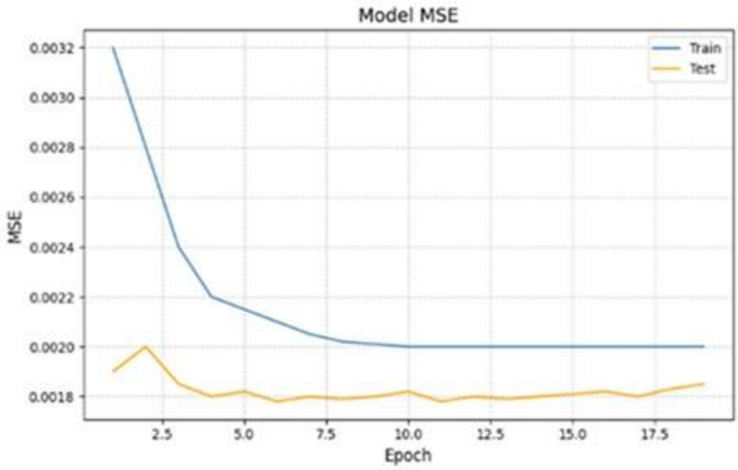

Figure 6. Training and testing MSE convergence over epochs.

To verify the validity of the performance of the LSTM model, we focused on the predicted positions of a particular cab (Cab 1). By plotting the actual positions and the predicted position (in green color) as shown in Fig 7, we found that the predicted position aligns well with the region where the cab is most frequently located, meaning it represents the location of the majority of the cab`s presence.

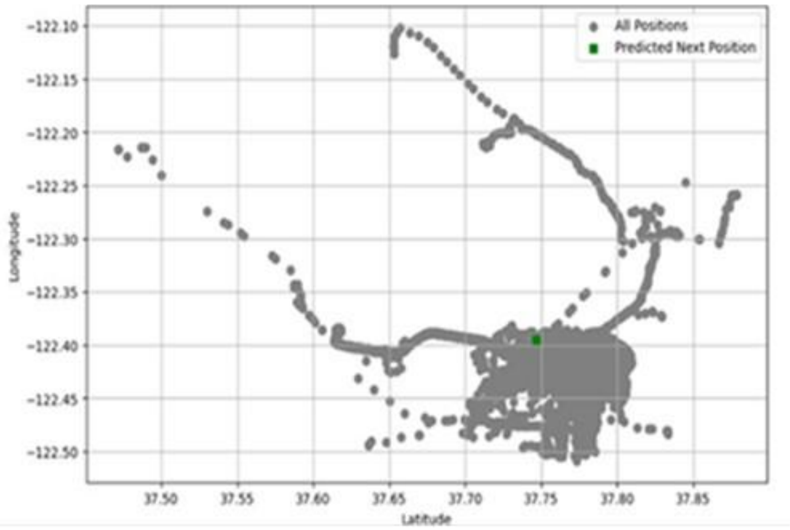

Figure 7. All positions of cab 1 and the predicted point.

4.4. Comparative Analysis of LSTM and Simple RNN Models

Figures 8 and 9 illustrate the comparative performance of the LSTM and Simple RNN models in terms of the Mean Squared Error (MSE) on the test dataset. The MSE serves as a key metric for evaluating the prediction accuracy of both models, where a lower MSE value indicates better predictive performance. In Fig 8, the evolution of the test loss (MSE) across epochs is plotted for both the LSTM and Simple RNN models. The LSTM model demonstrates a more consistent loss reduction trend over the training epochs, with occasional spikes indicating periods of adjustment in the model's learning process. In comparison, the Simple RNN model exhibits a slightly more volatile trend, with higher fluctuations in test loss. Despite these variations, both models achieve relatively low and comparable MSE values by the end of training. Fig 9 presents the final MSE values achieved by each model after training. The LSTM model attains an MSE of approximately 0.001814185, while the Simple RNN model achieves an MSE of 0.001814780. These results highlight that both architectures perform similarly in terms of predictive accuracy, with the LSTM

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

model showing a marginal advantage. This slight edge may be linked to the LSTM's strength in capturing long-term dependencies and mitigate vanishing gradient issues, which are limitations of the Simple RNN architecture.

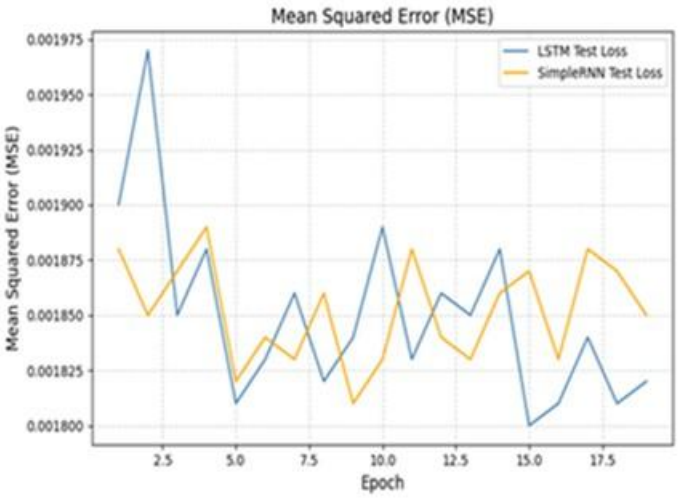

Figure 8. Comparative Test Loss (MSE) Progression Across Epochs for LSTM and Simple RNN.

The comparison underscores the comparable performance of the two architectures in this specific application while reinforcing the theoretical strengths of LSTMs in handling sequential data. The findings emphasize the importance of model selection based on specific use cases, computational resources, and the nature of the dataset.

```
1701/1701 ================================ 3s 1ms/step
LSTM Model MSE: 0.0018141850778880752
1701/1701 ================================ 3s 1ms/step
SimpleRNN Model MSE: 0.0018147800234901606
```

Figure 9. Final Mean Squared Error (MSE) Comparison Between LSTM and Simple RNN Models.

To enhance traffic management and optimize route planning, the positions of all cabs in the dataset were predicted using a trained model. For each cab, the last three recorded positions were extracted, and the model forecasted the next position. The predicted positions, as visualized in Figure 10, provide a comprehensive overview of vehicle density and distribution across the monitored area. These real-time predictions are critical for supporting smart transportation systems by enabling informed decision-making and improving traffic flow efficiency. This technique demonstrates the potential of predictive modelling in modern traffic management, offering valuable insights for reducing congestion and enhancing mobility solutions.

4.5. Comparative Analysis of LSTM and Simple RNN Models

After forecasting the next positions for each cab using the trained model, clustering techniques were employed to group these predicted positions based on their spatial proximity. Clustering is an essential data analysis method that identifies patterns or structures by grouping similar data points into cohesive clusters. In this context, clustering helps to organize the predicted positions of cabs into regions of high density, providing insights into traffic patterns and vehicle distribution. This process is vital for applications such as hotspot identification, resource allocation, and dynamic route optimization in smart transportation systems. Additionally, clustering provides a clear visualization of high-density areas within the city of San Francisco, offering valuable guidance for selecting optimal locations to deploy 5G antennas that will host MEC servers. This framework ensures enhanced network coverage and efficiency in high-demand areas.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

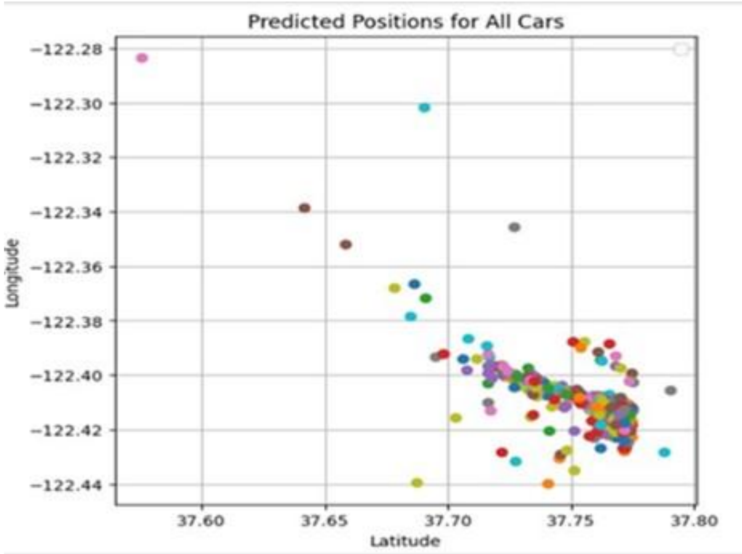

Figure 10. Predicted Positions for All Cabs in the Dataset.

Identifying the optimal number of clusters is a critical step in clustering analysis to ensure meaningful and interpretable groupings while avoiding unnecessary complexity. To achieve this, the elbow method was employed, which is a widely used technique for selecting the optimal number of clusters. The method includes plotting the inertia values, defined as the sum of squared distances of samples to their nearest cluster centre, versus the number of clusters. As shown in Fig 11, the plot forms an "elbow" shape, with the point of inflection marking the optimal number of clusters. This point represents a balance where increasing the number of clusters beyond this value does not lead to significant improvement in clustering performance, thus providing an efficient and robust model for analysing the data.

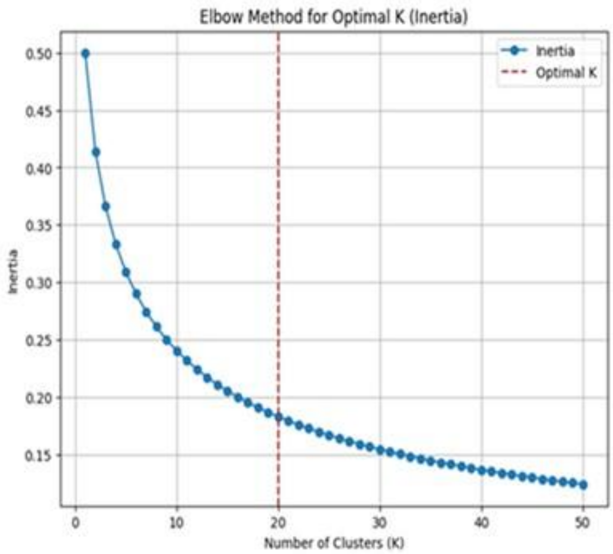

Figure 11. Elbow Method Plot for Determining the Optimal Number of Clusters (K).

After determining the optimal number of clusters using the elbow method, the K-Means clustering algorithm was employed to group the predicted positions into distinct clusters. Each cluster rep-resents a set of predicted positions that exhibit similar spatial characteristics, thereby capturing the inherent patterns within the data. This visualization not only enhances the interpretability of the clustering results but also provides valuable insights into the spatial organization and density of the predicted vehicle positions, which are essential for subsequent analyses and decision-making processes. Fig 12 illustrates the spatial distribution of predicted vehicle positions grouped into clusters. Each

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

coloured marker represents a predicted position belonging to a specific cluster, while the black stars indicate the centroids of the clusters. The clusters visually demonstrate how the predicted positions are organized into distinct groups based on spatial proximity, providing valuable insights into high-density areas and spatial patterns across the dataset. This information is crucial for ap-plications such as urban mobility analysis and strategic planning for infrastructure deployment, such as 5G antennas.

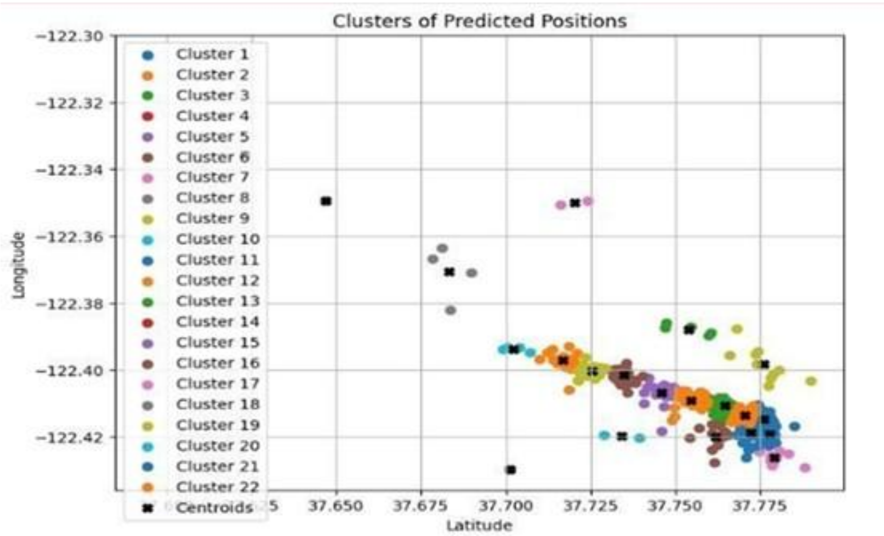

Figure 12. Clusters of Predicted Vehicle Positions with Centroids.

CONCLUSION

This study has demonstrated the potential of integrating deep learning techniques, specifically Long Short-Term Memory (LSTM) networks, with clustering algorithms to enhance urban infrastructure planning for Multi-access Edge Computing (MEC) deployment within the framework of 5G net-works. By leveraging the Cabspotting dataset, the LSTM model effectively predicted vehicle trajectories, outperforming simpler Recurrent Neural Network (RNN) models in accuracy. The application of K-Means clustering to these predictions enabled the detection and identification of high-density vehicle zones, which are critical for strategic MEC placement. The proposed approach not only addresses challenges in optimizing MEC deployment but also supports broader smart city initiatives by improving traffic management, alleviating congestion, and improving connectivity. By aligning 5G network performance with real-time urban mobility patterns, this framework provides a robust foundation for future advancements in urban communication systems. As a future direction, one could explore the integration of additional data sources, such as real-time traffic conditions and demographic information, to further refine the accuracy and applicability of the proposed framework. Moreover, the inclusion of other advanced machine learning techniques could enhance the predictive capabilities, enabling even more efficient and scalable solutions for MEC deployment and 5G network optimization.

REFRENCES

[1] Nuriev, M., Kalyashina, A., Smirnov, Y., Gumerova, G., & Gadzhieva, G. (2024). The 5G revolution transforming connectivity and powering innovations. E3S Web of Conferences, 515, 04008. https://doi.org/10.1051/e3sconf/202451504008

[2] Hakak, S., Gadekallu, T. R., Maddikunta, P. K. R., Ramu, S. P., Parimala, M., De Alwis, C., ... Hossain, M. S. (2023). Autonomous vehicles in 5G and beyond: A survey. Vehicular Communications, 39, 100551. https://doi.org/10.1016/j.vehcom.2022.100551

[3] Shehab, M. J., Kassem, I., Kutty, A. A., Kucukvar, M., Onat, N., & Khattab, T. (2021). 5G networks towards smart and sustainable cities: A review of recent developments, applications, and future perspectives. IEEE Access, 10, 2987–3006. https://doi.org/10.1109/ACCESS.2021.3139436

[4]Pham, Q.-V., Fang, F., Ha, V. N., Piran, M. J., Le, M., Le, L. B., ... Hossain, M. S. (2020). A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art. IEEE Access, 8, 116974–117017. https://doi.org/10.1109/ACCESS.2020.3001277

[5] Spinelli, F., & Mancuso, V. (2020). Toward enabled industrial verticals in 5G: A survey on MEC-based approaches to provisioning and flexibility. IEEE Communications Surveys & Tutorials, 23, 596–630. https://doi.org/10.1109/COMST.2020.3037674

[6] Gohar, A., & Nencioni, G. (2021). The role of 5G technologies in a smart city: The case for intelligent transportation system. Sustainability, 13(9), 5188. https://doi.org/10.3390/su13095188

[7] Xiong, R., Zhang, C., Zeng, H., Yi, X., Li, L., & Wang, P. (2022). Reducing power consumption for autonomous ground vehicles via resource allocation based on road segmentation in V2X-MEC with resource constraints. IEEE Transactions on Vehicular Technology, 71(7), 6397–6409. https://doi.org/10.1109/TVT.2022.3161641

[8] Waqar, N., Hassan, S. A., Mahmood, A., Dev, K., Do, D.-T., & Gidlund, M. (2022). Computation offloading and resource allocation in MEC-enabled integrated aerial-terrestrial vehicular networks: A reinforcement learning approach. IEEE Transactions on Intelligent Transportation Systems, 23(11), 21478–21491. https://doi.org/10.1109/TITS.2022.3179987

[9] Josey, B., & Misbha, D. S. (2025). Elliptic curve cryptography algorithm with recurrent neural networks for attack detection in industrial IoT. International Journal of Computer Networks & Communications, 17(1), 83–98. https://doi.org/10.5121/ijcnc.2025.17105

[10] Boulesnane, A., Saidi, Y., Kamel, O., Bouhamed, M. M., & Mennour, R. (2022). Dzchatbot: A medical assistant chatbot in the Algerian Arabic dialect using Seq2Seq model. In 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS). IEEE. https://doi.org/10.1109/PAIS56586.2022.9946867

[11] Khan, A., Fouda, M. M., Do, D.-T., Almaleh, A., & Rahman, A. U. (2023). Short-term traffic prediction using deep learning long short-term memory: Taxonomy, applications, challenges, and future trends. IEEE Access, 11, 94371–94391. https://doi.org/10.1109/ACCESS.2023.3309601

[12] Chapagain, K., Gurung, S., Kulthanavit, P., & Kittipiyakul, S. (2023). Short-term electricity demand forecasting using deep neural networks: An analysis for Thai data. Applied System Innovation, 6(6), 100. https://doi.org/10.3390/asi6060100

[13] Bock, F., & Di Martino, S. (2019). On-street parking availability data in San Francisco, from stationary sensors and high-mileage probe vehicles. Data in Brief, 25, 104039. https://doi.org/10.1016/j.dib.2019.104039

[14] Ben Bezziane, M., Korichi, A., Kerrache, C. A., & Fekair, M. A. (2021). Rcvc: RSU-aided cluster-based vehicular clouds architecture for urban areas. Electronics, 10(2), 193. https://doi.org/10.3390/electronics10020193

[15] Chen, X., Cheng, X., Wu, N., & Liu, X. (2024). Unmanned aerial vehicle transmission defect detection technology based on edge computing. Diagnostyka, 25. https://doi.org/10.29354/diag/194598

[16] Alqahtani, F., Al-Maitah, M., & Elshakankiry, O. (2022). A proactive caching and offloading technique using machine learning for mobile edge computing users. Computer Communications, 181, 224–235. https://doi.org/10.1016/j.comcom.2021.10.017

[17] Lin, L., Li, W., Bi, H., & Qin, L. (2021). Vehicle trajectory prediction using LSTMs with spatial–temporal attention mechanisms. IEEE Intelligent Transportation Systems Magazine, 14(2), 197–208. https://doi.org/10.1109/MITS.2021.3049404

[18] Nguyen, T.-V., Dao, N.-N., Noh, W., & Cho, S. (2021). User-aware and flexible proactive caching using LSTM and ensemble learning in IoT-MEC networks. IEEE Internet of Things Journal, 9(4), 3251–3269. https://doi.org/10.1109/JIOT.2021.3097768

[19] Nguyen, C., Klein, C., & Elmroth, E. (2019). Multivariate LSTM-based location-aware workload prediction for edge data centers. In 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE. https://doi.org/10.1109/CCGRID.2019.00045

[20] Ma, L., Wang, P., Du, C., & Li, Y. (2022). Energy-efficient edge caching and task deployment algorithm enabled by deep Q-learning for MEC. Electronics, 11(24), 4121. https://doi.org/10.3390/electronics11244121

[21] Do, H. M., Tran, T. P., & Yoo, M. (2023). Deep reinforcement learning-based task offloading and resource allocation for industrial IoT in MEC federation system. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3302518

[22] Fattore, U., Liebsch, M., Brik, B., & Ksentini, A. (2020). AutoMEC: LSTM-based user mobility prediction for service management in distributed MEC resources. In Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (pp. 155–159). https://doi.org/10.1145/3416010.3423246
[23] Hammedi, W., Brik, B., & Senouci, S. M. (2022). Toward optimal MEC-based collision avoidance system for cooperative inland vessels: A federated deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 24(3), 2525–2537. https://doi.org/10.1109/TITS.2022.3154158