Intelligent Control and System Engineering 2024, 2(1), 1871.
https://doi.org/10.59400/icse1871

Article

Comparison of the elevator traffic flow prediction between the neural networks of CNN and LSTM

Mo Shi[1,2,*], Yeol Choi[1]

[1] School of Architecture, Kyungpook National University, Daegu 41566, Korea
[2] HaXell Elevator Co., Ltd., Shanghai 201801, China
*** Corresponding author:** Mo Shi, shimo0204@outlook.jp

CITATION

Shi M, Choi Y. Comparison of the elevator traffic flow prediction between the neural networks of CNN and LSTM. Intelligent Control and System Engineering. 2024; 2(1): 1871.
https://doi.org/10.59400/icse1871

ARTICLE INFO

Received: 14 October 2024
Accepted: 3 December 2024
Available online: 18 December 2024

Abstract: With urbanization rapidly increasing, the demand for efficient elevator systems is becoming ever more pressing, particularly in crowded urban centers where high-rise buildings are prevalent. To solve this issue, elevator traffic analysis and prediction have emerged as critical components for optimizing elevator control systems. The elevator traffic flow prediction not only ensures smoother operations during peak usage times but also significantly reduces waiting periods for passengers, thereby enhancing overall convenience. By leveraging neural networks, the performance of elevator control systems is expected to be improved, leading to more efficient and convenient elevator utilization in both residential and commercial environments. Over the past few decades, the rapid advancements in neural networks have provided valuable tools for predicting traffic flows. In this research, a total of 655 actual ETF (Elevator Traffic Flow) data points from a typical office building on a weekday are utilized to analyze and predict traffic patterns using CNN (Convolutional Neural Networks) and LSTM (Long Short-Term Memory). The objective is not only to demonstrate the applicability of the neural networks in predicting elevator traffic flow but also to conduct a comparative analysis to identify which offers greater accuracy and suitability for the elevator traffic flow prediction. By enhancing the capabilities of elevator control systems through CNN or LSTM, this research seeks to improve not only the efficiency of elevator operations but also the overall living and working environment in urban cities. The findings from this research can inform subsequent research efforts, encouraging a deeper exploration of how synthetic predictions can further optimize elevator control systems, while the synthetic elevator control system is expected to lead to significant improvements in passenger experience, reducing wait times and increasing overall satisfaction in both residential and commercial buildings. Therefore, the insights gained from this research are expected to play a crucial role in shaping the future of smart buildings, aligning with the demands of modern urban living.

Keywords: ETF (Elevator Traffic Flow); neural networks; CNN (Convolutional Neural Networks); LSTM (Long Short-Term Memory); prediction

1. Introduction

As urbanization continues to accelerate in contemporary society, an increasing number of people are moving into cities, including residential needs, creating significant challenges. The sheer size of urban populations has given rise to numerous social issues that require immediate attention, with traffic congestion emerging as one of the most pressing problems. To better understand and address traffic concerns, extensive research has been conducted over the years [1–6], emphasizing the importance of this issue and offering various analytical approaches to solve the traffic issues. In recent decades, with the rapid development of neural network technology, traffic flow prediction methods have been developed, showing considerable promise in addressing traffic congestion. These predictive models have contributed significantly to reducing traffic jams, thus improving mobility and convenience for people in urban cities.

According to the previous research of Zhao et al. [7], the LSTM (Long Short-Term Memory) neural network was identified as a highly suitable tool for short-term traffic forecasting. Through a detailed comparison of various neural networks, the research discussions highlight the superior accuracy and robustness of the LSTM model, particularly when applied to traffic data. The analysis was conducted using three observation points within the traffic network of Beijing City, where the LSTM model consistently outperformed other neural network approaches. This research underscores the effectiveness of LSTM in addressing complex urban traffic prediction tasks, offering a reliable solution for managing short-term traffic conditions in urban cities. In addition, the previous research of Kang et al. [8] also emphasizes the application of the LSTM model in the field of traffic prediction. The research demonstrates that the LSTM model has significant potential in mitigating the growing problem of traffic congestion in urban cities. With the increasing complexity of traffic conditions in densely populated areas, the LSTM model offers an advanced approach to forecast traffic patterns and identify potential bottlenecks accurately.

Moreover, the previous research of Lazaris et al. [9] also highlighted the extensive use of the LSTM model in network traffic prediction, owing to its ability to deliver highly accurate forecasts with minimal prediction errors. The research also explored several variations of the LSTM model, including vanilla LSTM, Delta LSTM, Cluster LSTM, and Cluster Delta LSTM, each of which offers unique advantages in modeling network traffic. These variants enhance the flexibility and precision of traffic modeling, making the LSTM family of models particularly suitable for managing the complexities of modern network systems.

Referring to many previous research. the LSTM neural network model has applications beyond standard traffic prediction, as it is also being utilized in the field of ETA (Elevator Traffic Analysis). According to research by Zheng et al. [10], and Shi et al. [11], the LSTM model provides a promising solution to typical elevator traffic problems. By accurately forecasting elevator usage patterns, especially during peak periods such as mornings, lunch breaks, and evenings [12], LSTM-based systems are significantly expected to reduce elevator congestion in tall office buildings. By integrating LSTM models to predict and manage elevator traffic more efficiently, the overall accessibility and functionality of high-rise buildings are expected to improve, particularly for office spaces where elevator bottlenecks are most common. The LSTM model offers a practical method to enhance the daily operations of tall buildings, ensuring smoother vertical transportation during high-demand periods.

In addition to the widely recognized capabilities of LSTM networks in general traffic analysis and network traffic forecasting, CNN (Convolutional Neural Network) has established its presence across numerous domains throughout the decades [12–16]. The integration of these neural network architectures not only enhances the accuracy of traffic predictions but also streamlines urban mobility [17–19]. Despite the differences between LSTM and CNN, both models offer valuable approaches for predicting traffic patterns, ultimately contributing to more efficient and convenient urban dwellings and working environments. As cities continue to grow and face increasing transportation challenges, the application of LSTM and CNN models becomes ever more critical. These advanced technologies offer innovative solutions that facilitate smarter traffic management, ultimately contributing to the convenience and sustainability of urban living.

Elevator systems play a critical role in managing vertical transportation within buildings, significantly influencing the efficiency of traffic flow and the overall user experience. Prolonged waiting times and extended travel durations often frustrate passengers, leading to dissatisfaction and frequent complaints about an inconvenient living environment. Recognizing the importance of this issue, many researchers have directed their efforts toward optimizing elevator traffic management. For instance, Luo et al. [20] explored the use of LS-SVM (Least Squares Support Vector Machines) models for elevator traffic prediction. Their findings highlight the model’s ability to accurately predict elevator traffic, providing a foundation for improving the efficiency and reliability of these systems.

In this research, both LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) are applied to analyze typical elevator traffic flow, the whole day traffic flow in the typical office building on the weekdays upon the analytical results of Luo et al. [20]. To streamline the analysis, this research uses an elevator traffic dataset collected exclusively from a 14-floor office building, including a basement floor designated for parking. The detailed specifications of the four elevators are outlined in **Table 1**, and each elevator has a design capacity of 1000 kg, with door opening and closing times set at 2.9 s and 3.3 s respectively. As **Table 1** indicates, the elevators are designed to operate at a speed of 2 m/s and an acceleration of 0.8 m/s^2. Furthermore, the home floor for all four elevators is the 1st floor as **Table 1** emphasizes, serving as the default starting floor of the four elevators within the typical office building in this research.

Table 1. Overview of the elevators.

Number of Elevators	**4**
Capacity	1000 kg
Door Pre-opening Time	0 s
Door Open Time	2.9 s
Door Close Time	3.3 s
Speed	2 m/s
Acceleration	0.8 m/s^2
Jerk	1.2 m/s^3
Start Delay	0.5 s
Home Floor	1 F

Although the elevator traffic in a single typical office building helps reduce variables and complexity, this research acknowledges the inherent limitations of relying on data from a single source. Despite these constraints, the findings are anticipated to provide valuable insights for future research. They are expected to serve as a foundational reference for studies involving multiple buildings or more diverse environments, aiming to improve the generalizability and applicability of elevator traffic flow models. This research focuses on predicting elevator traffic flow specifically on weekdays, as the volume and complexity of traffic during these days are significantly greater compared to weekends. Due to the increased variety in weekday elevator usage, which makes accurate prediction particularly challenging, elevator traffic data were collected from a typical office building, with samples taken every minute between 7:00 and 18:00, resulting in 655 data points recorded daily. Recognizing that raw data often contain noise, this research employed a database consisting of averaged traffic flow data over four weeks (one month) to ensure a more accurate and reliable analysis.

This research leverages the unique characteristics of LSTM and CNN neural network models to examine the respective roles in ETA (Elevator Traffic Analysis) and ETP (Elevator Traffic Prediction). By comparing these neural network structures, the discussions of this research seek to identify how the differences influence elevator traffic prediction performance, highlighting the advantages each model offers in the prediction performance of the elevator traffic flow. Although minimizing elevator cycle time $v(t)$ remains a primary focus for ETA, incorporating neural networks provides new opportunities for improving elevator control systems [21,22]. As the research of Markos et al. [23] indicates, reducing acceleration, braking times, and dwell times at stations effectively shortens cycle times. Following by the previous theories of ETA, the research explores how neural networks (LSTM and CNN) can advance elevator traffic control systems through predictive capabilities. The findings from this research are expected to contribute to developing efficient prediction models that enhance elevator system functionality, especially in high-rise buildings. These advancements are expected to play a vital role in urban environments, where improving elevator efficiency can significantly reduce virticle transport time cost and enhance overall building accessibility, while the insights gained will support the creation of more convenient and efficient living and working spaces in densely populated cities.

2. Discussion of the CNN and LSTM framework

2.1. CNN framework

Referring to much previous research, CNN (Convolutional Neural Network) is a type of deep learning algorithm composed of several key layers, including convolutional layers, pooling layers, and fully connected layers as **Figure 1** illustrates. Traditionally, CNN has shown exceptional performance in identifying patterns in images, making them highly effective for tasks such as object detection, classification, and recognition. Many previous research have demonstrated the ability of CNN to extract features from images and categorize them into specific classes or objects [24–26]. However, the applications of CNN extend far beyond image recognition. CNN is also employed in other areas such as NLP (Natural Language Processing) [27,28], time series analysis [29,30], and speech recognition [31–33].

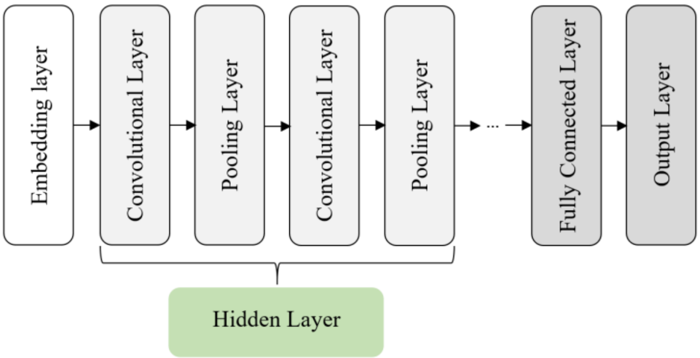


Figure 1. CNN framework.

Referring to **Figure 1**, the core components of CNN of the convolutional and fully connected layers are critical to their deep learning architecture. Convolutional layers apply filters to input data from the embedding layer that highlight various important characteristics of the input, whether in images or other forms of data. Meanwhile, fully connected layers link each neuron from one layer to every neuron in the subsequent layer, allowing the network to make its final classification decisions based on the extracted features. This combination of feature extraction and classification is what enables CNN to be such a powerful and versatile tool across multiple domains.

As illustrated in **Figure 1**, the framework of CNN is composed of distinct layers, each playing a specific role in the deep learning process. The visible layer consists of the embedding layer, fully connected layer, and output layer. These layers handle the mapping of input data, the transformation of learned features into final predictions, and the ultimate output of the network. On the other hand, the hidden layers are made up of convolutional layers and pooling layers, which are essential for feature extraction and dimensionality reduction. The convolutional layer applies filters to the input, capturing spatial relationships and features within the data, while the pooling layer reduces the dimensionality, preserving key information while making the model more computationally efficient.

One of the key strengths of CNN lies in its flexibility. The convolutional and pooling layers can be repeated multiple times, allowing the model to build hierarchical representations of the input data. By increasing the depth of the network, CNN is capable of learning more complex patterns and achieving higher accuracy in their predictions.

This research focuses on analyzing elevator traffic in a typical office building using CNN, leveraging data collected over a single weekday. A total of 655 data points are gathered, representing the traffic flow of the elevators throughout the day in the typical office building. To optimize the performance of the CNN model, the dataset is divided into two distinct parts as **Figure 2** illustrates. The first 200 data points are used for the training process which represents the morning peak traffic rush, while the remaining 455 data points are designated for the testing process that corresponds to the noon peak and end-of-day traffic rush. For effective model training and testing, the first 10 data points are specifically used for initializing the learning process as depicted in **Figure 2**. This division of data provides a structured approach to managing the input, ensuring that both training and testing are appropriately set up to reflect real-world traffic patterns.

```
%% Divide train data and test data
temp = 1: 1: 655;

P_train = res(temp(1: 200), 1: 10)';
T_train = res(temp(1: 200), 11)';
M = size(P_train, 2);

P_test = res(temp(201: end), 1: 10)';
T_test = res(temp(201: end), 11)';
N = size(P_test, 2);
```

Figure 2. Dividing and inputting the data.

As illustrated in **Figure 3**, the architecture of the CNN in this research comprises three convolutional layers, each paired with a normalization layer and a ReLU (Rectified Linear Unit) activation layer. These layers work in tandem to extract features from the input data and normalize them to improve the learning ability of the network. The kernel size for all convolutional layers is set to 3 × 1, ensuring that the CNN captures relevant features in a structured and efficient manner. As the network deepens, the number of feature maps increases: the first convolutional layer has 11 feature maps, the second layer has 22, and the third layer has 44. This progressive increase allows the model to detect more complex patterns as it processes the data, thereby improving its overall feature extraction capabilities. At the final stage of the network, the fully connected layer is designed to process the learned features and produce the output result. This layer takes the abstracted features from the convolutional layers and maps them to the final predictions, completing the learning process of CNN.

```
%% Structure of the model
layers = [
    imageInputLayer([10, 1, 1])    % Input Layer Input data [10, 1, 1]

    convolution2dLayer([3, 1], 11) % Kernel size 3×1  Feature maps 11
    batchNormalizationLayer        % Normalization Layer
    reluLayer                      % ReLU Activation Layer

    convolution2dLayer([3, 1], 22) % Kernel size 3×1  Feature maps 22
    batchNormalizationLayer        % Normalization Layer
    reluLayer                      % ReLU Activation Layer

    convolution2dLayer([3, 1], 44) % Kernel size 3×1  Feature maps 44
    batchNormalizationLayer        % Normalization Layer
    reluLayer                      % ReLU Activation Layer

    fullyConnectedLayer(1)         % Connection Layer
    regressionLayer];              % Regression Layer
```

Figure 3. Structure setting of CNN.

Several critical parameters are configured to enhance the training performance of the model in the deep learning setup for the CNN as depicted in **Figure 4**. The Adam (Adaptive Moment Estimation) optimization algorithm is chosen for training, known for its efficiency and adaptive learning rate adjustments that lead to faster and more stable convergence.

```
%% Parameter setting
options = trainingOptions('adam', ...       % Adaptive Moment Estimation
    'MaxEpochs',300, ...                     % Maximum number of epochs
    'InitialLearnRate', 1e-2, ...            % Initial learning rate
    'LearnRateSchedule', 'piecewise', ...    % Learning rate schedule
    'LearnRateDropFactor', 0.1, ...          % Reduce factor
    'LearnRateDropPeriod', 200, ...          % Drop period
    'Shuffle', 'every-epoch', ...            % Data randomize
    'Plots', 'training-progress', ...        % Plot curves
    'Verbose', false);
```

Figure 4. Parameter setting.

Referring to **Figure 4**, the maximum number of epochs is set to 300, allowing the network to go through extensive training iterations to thoroughly learn the underlying patterns in the data. The initial learning rate is configured at 0.01, which determines how quickly the model updates its weights during the learning process. To ensure that the model refines its learning over time, a piecewise learning rate schedule is implemented, while after the first 200 times for epochs, the learning rate is reduced by a factor of 0.1. The gradual reduction allows the model to fine-tune its parameters, promoting more precise learning as it nears completion.

Throughout the entire training and testing process, data shuffling is employed to randomize the order of the data before each epoch. This step ensures that the CNN cannot memorize any fixed order in the dataset, leading to better generalization when tested on unseen data. The shuffle is applied at every epoch, which helps to reduce biases and prevents overfitting, improving both the training and testing performance.

Table 2. Layer information of CNN.

Layer	Name	Type
1	Image Input 10×1×1 images with 'zerocenter' normalization	Image Input
2	conv_1 11 3×1×1 convolutions with stride [1 1] and padding [0 0 0 0]	2-D Convolution
3	batchnorm_1 Batch normalization with 11 channels	Batch Normalization
4	relu_1 ReLU	ReLU
5	conv_2 22 3×1×11 convolutions with stride [1 1] and padding [0 0 0 0]	2-D Convolution
6	batchnorm_2 Batch normalization with 22 channels	Batch Normalization
7	relu_2 ReLU	ReLU
8	conv_3 44 3×1×22 convolutions with stride [1 1] and padding [0 0 0 0]	2-D Convolution
9	batchnorm_3 Batch normalization with 44 channels	Batch Normalization
10	relu_3 ReLU	ReLU
11	fc 1 fully connected layer	Fully Connected
12	regressionoutput mean-squared-error with response 'Response'	Regression Output

In this research, the CNN framework is composed of a total of 12 layers as outlined in **Table 2**. Especially, these layers include convolutional layers which extract key features from the data, normalization layers which help standardize the data and improve model stability, and ReLU (Rectified Linear Unit) activation layers which introduce non-linearity into the model, enabling it to learn complex patterns.

This multi-layered architecture provides the foundation for analyzing the elevator traffic flow in a typical office building over the course of a single weekday. By processing the data through these 12 layers, the CNN is able to capture intricate details in the elevator traffic flow, such as peak times and usage patterns. The combination of convolutional, normalization, and activation functions allows the model to progressively refine its understanding of the data, leading to more accurate predictions of elevator traffic behavior.

2.2. LSTM framework

Many previous studies indicate that the LSTM (Long Short-Term Memory) network emerges as a significant advancement in the realm of deep learning, particularly within the framework of RNN (Recurrent Neural Network) [34–36]. While RNN has proven effective in tasks that involve sequence prediction, it often faces challenges when it comes to long-term dependencies [37,38]. Traditionally, RNN can struggle to maintain relevant information over extended sequences, which limits its ability to learn complex patterns that unfold over time. In contrast, the LSTM network is designed to address this limitation by incorporating multiple specialized components within each LSTM cell, including the forget gate, input gate, and output gate [39,40]. These gates work together to manage the flow of information through the network. The forget gate selectively discards information that is no longer needed, the input gate controls which new information is added, while the output gate regulates the release of information from the cell.

This unique architecture enables the LSTM network to maintain and update state vectors, allowing them to store and selectively modify information over long sequences. As a result, LSTM excels at identifying intricate patterns and dependencies that the conventional RNN might overlook. This makes them particularly well-suited for tasks such as time-series forecasting, speech recognition, and natural language processing, where retaining and understanding long-term relationships in data is critical.

As depicted in **Figure 5**, a memory cell in a typical LSTM network plays a central role in the processing and retention of sequential information. This cell not only utilizes the input data from the current timestamp but also incorporates memory from the previous timestamp to perform feature extraction. By combining these two sources of information, the LSTM is able to capture both the immediate and historical context of the sequence, which is essential for learning long-term dependencies. The processed information is stored in the memory cell (C_t), which holds the relevant features extracted at the current time step. This memory (C_t) is then passed forward to the next time step, ensuring that the LSTM network retains important information across multiple timestamps, enhancing its ability to learn patterns over time.

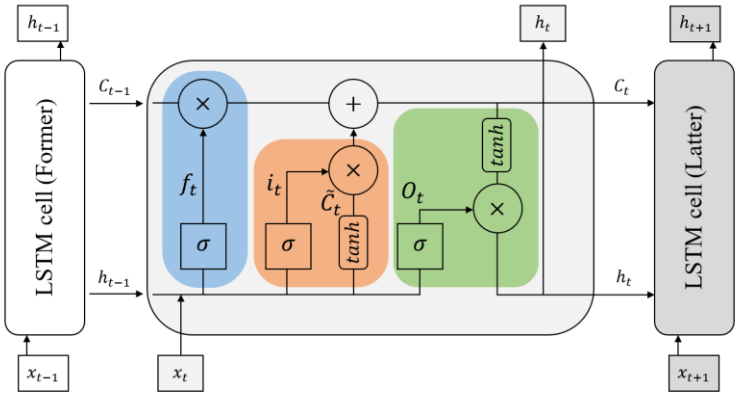


Figure 5. LSTM framework.

In addition to updating the memory cell, the LSTM cell generates output data (h_t) based on the current memory state (C_t). This output represents the response of LSTM to the input at the specific time step, factoring in both the new input and the memory it has retained. As shown in **Figure 6**, this process of memory retention, feature extraction, and output generation is repeated at each time step, allowing the LSTM to effectively process sequences of data.

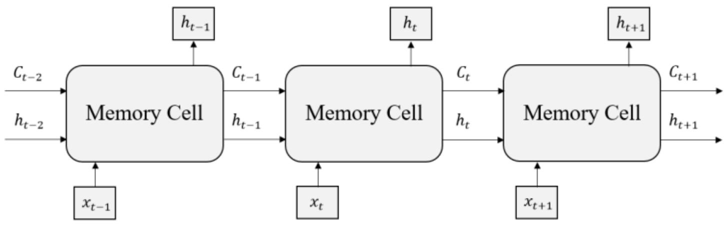


Figure 6. Unrolled form of LSTM network.

In the architecture of an LSTM memory block, three primary gates work together to control how information is stored, updated, and output. These gates are the forget gate, the input gate, and the output gate, each playing a crucial role in the flow of data within the LSTM cell.

As depicted in **Figure 5**, the forget gate is represented by the blue part of the LSTM cell. This gate is responsible for deciding which parts of the existing memory (previously stored information) should be retained or discarded. It evaluates the current memory state and selectively forgets irrelevant or outdated information, allowing the LSTM to focus on important data while avoiding the buildup of unnecessary details. The blue part in **Figure 5** illustrates the forget gate in the LSTM cell, and its mathematical process is explained by the equation provided below:

$$f = \sigma\left(W_f \cdot [h_{t-1}, x_1] + b_f\right) \tag{1}$$

The input gate, shown in orange, manages the new input information entering the LSTM cell at the current time step. It determines which parts of this new input are relevant and should be added to the memory, ensuring that the network incorporates new information effectively without overwhelming the memory with unimportant data. The orange part in **Figure 5** illustrates the input gate in the LSTM cell, and its mathematical process is explained by the equations provided below:

$$i_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_i\right) \tag{2}$$

$$\tilde{C} = tanh(W_c \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

The output gate, represented by the green part of the LSTM cell, controls the generation of the output information (h_t). After the forget and input gates have updated the memory state (C_t), the output gate decides which portion of the memory should be used to generate the current output, which will be passed on to the next layer or time step in the sequence. The green part in **Figure 5** illustrates the output gate in the LSTM cell, and its mathematical process is explained by the equations provided below:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(\, C_t) \tag{6}$$

where, in Equations (4) and (6), $*$ denotes element-wise multiplication, while in Equations (1), (2), and (5), $\sigma(x) = (1 + e^{-x})^{-1}$ represents the sigmoid function. In Equations (1)–(3), and (5), W and b are matrices corresponding to the learning parameters of LSTM.

The LSTM memory cells provide neural networks with the ability to process a sequence of inputs, denoted as (x_1, x_2, ..., x_t, ..., x_T), and generate corresponding timestamped outputs (h_1, h_2, ..., h_t, ..., h_T), where T represents the total length of the input data sequence. At each time step t, the output h_t is generated based on all the input data received from time 1 to time t. This means that LSTM can accumulate knowledge over time, with each output reflecting the information processed up to that specific point in the sequence.

For instance, at time step 1, the network generates h_1 based solely on the first input x_1, while at time step 3, the output h_3 is influenced by the inputs x_1, x_2, and x_3. As the network progresses through the sequence, each subsequent output is increasingly informed by the entire history of inputs. This ability to dynamically incorporate past inputs into the current output makes LSTM networks especially powerful for tasks involving time-dependent or sequential data.

In alignment with the discussion on CNN, the same dataset representing elevator traffic in a typical office building, collected over the course of a single weekday, is also utilized for the LSTM model. This consistent use of data allows for a comparative analysis between the performance of CNN and LSTM in handling the prediction of elevator traffic flows. As described in **Figure 2**, the dataset consists of a total of 655 data points, which are divided into two parts: the first 200 data points are reserved for the training process, representing the morning peak traffic rush, where the highest volume of elevator usage typically occurs. The remaining 455 data points are allocated for the testing process, corresponding to the noon peak and the end-of-day traffic rush. By applying the same data division for both the CNN and

LSTM models, the research ensures a fair comparison of how these two deep-learning architectures handle the task of elevator traffic prediction.

As depicted in **Figure 7**, the LSTM architecture designed for this research features a structured deep-learning model comprising five distinct layers. The first layer in the LSTM architecture is the input layer, which is designed to handle a total of 10 input features. These input features represent the essential variables or data points that are fed into the network for analysis and prediction. The selection of 10 features ensures that the LSTM model has sufficient information to learn from while avoiding excessive complexity that might hinder its performance. The second key component is the hidden unit within the LSTM layer, also set to 10 units. These hidden units are responsible for storing and updating the internal memory of the network over time, capturing both short-term and long-term dependencies within the sequence of input data. The 10 hidden units provide the capacity to model intricate patterns in the elevator traffic data while maintaining computational efficiency. Following the LSTM layer, the network incorporates a ReLU activation layer, and this allows the LSTM to handle more complex relationships between the input features and the predicted outcomes. The ReLU activation is connected to a fully connected layer that is tasked with generating a single prediction unit. This means that the output of the network at each time step is a single value, representing the predicted outcome based on the learned patterns. Finally, the architecture concludes with a regression layer, which serves as the last component of the LSTM structure. This layer is responsible for transforming the output from the fully connected layer into a form suitable for regression tasks, it indicates predicting continuous variables like elevator traffic flow. The regression layer ensures that the LSTM model provides accurate numerical predictions, making it an essential part of the overall architecture.

```
%% Structure of the model
layers = [
    sequenceInputLayer(10)                  % Input Layer

    lstmLayer(10, 'OutputMode','last')  % LSTM Layer
    reluLayer                               % ReLU Activation Layer

    fullyConnectedLayer(1)                  % Connection Layer
    regressionLayer];                       % Regression Layer
```

Figure 7. Structure setting of LSTM.

The parameters of LSTM are configured similarly to those of the CNN in **Figure 4**. The Adam optimization algorithm is used, with a maximum of 300 epochs. The initial learning rate is 0.01, and a piecewise learning rate schedule reduces the rate by a factor of 0.1 after 200 epochs.

Referring to **Table 3**, The architecture begins with the input layer, which receives and organizes the raw data into a format suitable for further processing by the network. Following this, the core of the architecture is the LSTM layer, which is responsible for handling the temporal dependencies within the data. Next, the network employs a ReLU activation layer, and this is crucial for enabling the network to learn complex relationships in the data, as the ReLU helps to avoid the vanishing gradient problem commonly encountered in deep learning. The connection layer then links the LSTM outputs to the final processing stage, ensuring that the information flows smoothly through the network. Lastly, the regression layer is designed to generate the final predictions, which in this case involves forecasting elevator traffic patterns based on the learned data.

Table 3. Layer information of LSTM.

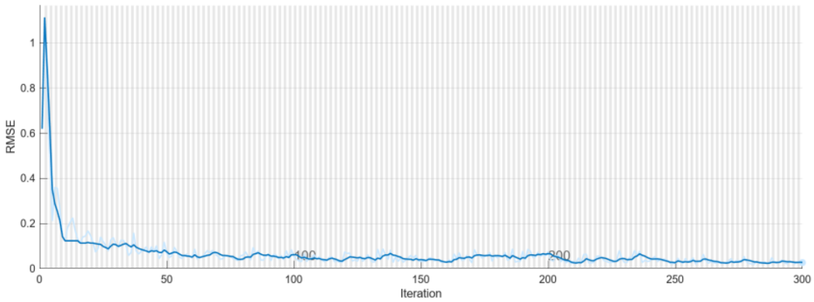
Layer	Name	Type
1	sequenceinput Sequence input with 10 dimensions	Sequence Input
2	lstm LSTM with 10 hidden units	LSTM
3	relu ReLU	ReLU
4	fc 1 fully connected layer	Fully Connected
5	regressionoutput mean-squared-error with response 'Response'	Regression Output

By structuring the LSTM model with 10 input features, 10 hidden units, a ReLU activation layer, a fully connected layer, and a regression layer, the architecture is tuned to capture the complexities of time-series data while producing reliable, precise predictions for elevator traffic.

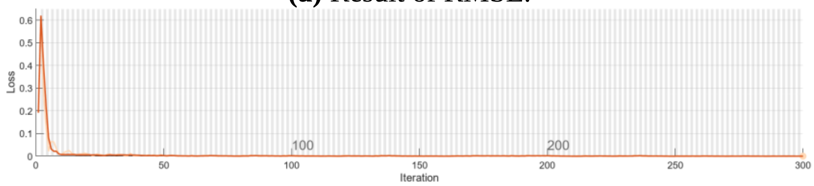
3. Discussion of the prediction results

3.1. Prediction results of CNN

Regarding the elevator traffic flow prediction using the CNN model, the parameter settings include a total of 300 epochs as illustrated in **Figure 8**. The high number of epochs is designed to give the model ample opportunity to learn and adjust its parameters throughout the training process.



(a) Result of RMSE.

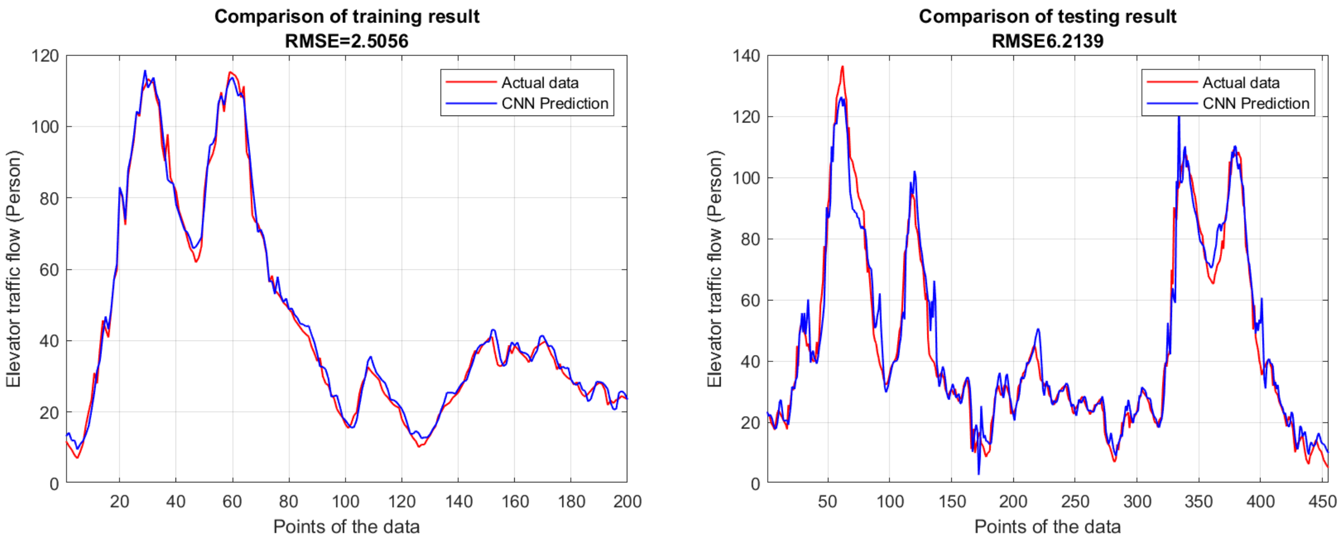


(b) Result of loss.

Figure 8. Training progress of CNN.

Based on the data of the elevator traffic flow in this research, **Figure 8** shows that the RMSE and loss function values stabilize after the first 50 epochs. This indicates that after 50 iterations, the model reaches a level of training stability, where further improvements become minimal. Essentially, the model has already learned the essential patterns in the data by this point, and additional training beyond 50 epochs does not yield significant improvements in performance. The graphs in **Figure 8** of the loss function and RMSE suggest that while 300 epochs were initially planned, 50 epochs may be sufficient to achieve optimal training. Continuing the training process beyond this point may only lead to marginal gains, making it less efficient in terms of time and computational resources.

As shown in **Figure 9**, the training and testing results of the CNN are compared, demonstrating the ability of CNN to learn from the training data and predict elevator traffic flow with accuracy. The graph presents a side-by-side comparison between the actual elevator traffic data and the predictions generated by the CNN, allowing for a visual assessment of how well the model performs during both phases.



(a) Training results **(b)** Testing results

Figure 9. Comparison of training and testing results of CNN.

The training results reflect the model's learning capacity, showcasing how effectively the CNN captures patterns and relationships within the elevator traffic data. Meanwhile, the testing results illustrate the model's predictive capacity, indicating how accurately it can forecast traffic patterns based on unseen data. Both sets of results demonstrate a high level of precision, reinforcing the reliability of the CNN in modeling elevator traffic flows.

One of the key metrics used to evaluate the accuracy of the predictions is the RMSE (Root Mean Square Error), a widely accepted measure for assessing the quality of regression models. In **Figure 9**, the RMSE values are provided for both the training and testing results and the RMSE can also be explained with the equation below:

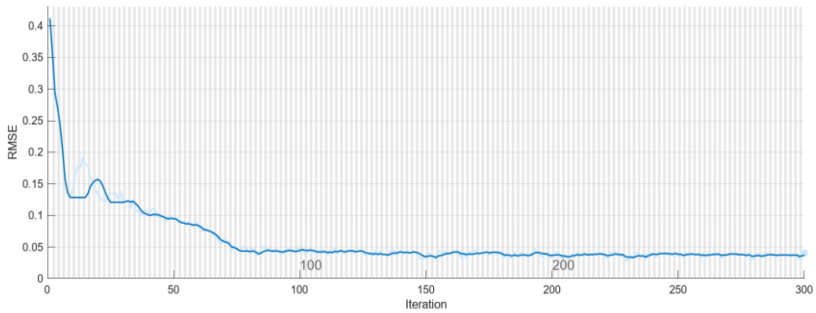
$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} w_i (y_i - \hat{y}_i)^2} \tag{7}$$

As illustrated in **Figure 9a**, the predicted results of CNN during the training phase for the morning traffic rush closely mirror the actual elevator traffic patterns. CNN exhibits high accuracy in its predictions, with a very small RMSE of 2.5056, indicating minimal deviation between the predicted and actual values. This low RMSE reflects the robust ability of CNN to learn the elevator traffic flow patterns during the training process, allowing it to produce reliable predictions for the morning elevator usage.

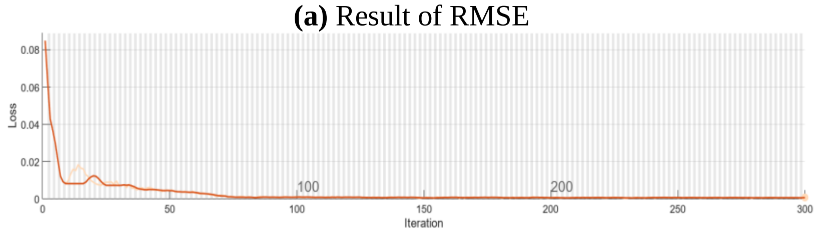
Shifting the focus to **Figure 9b**, the testing results demonstrate the performance of CNN during the noon and end-of-day traffic rush. While the RMSE is higher compared to the training phase at 6.2139, CNN still manages to accurately predict the overall flow of elevator traffic during the testing phase. The increase in RMSE is not uncommon when comparing training to testing results, as the model is exposed to new, unseen data during testing. Nevertheless, the predicted results remain closely aligned with the actual elevator traffic flow, suggesting that CNN retains its predictive capacity despite the increase in error.

3.2. Prediction results of LSTM

The LSTM for elevator traffic flow prediction in this research follows the same parameter settings as designed in CNN, with a total of 300 epochs being utilized throughout the training process, as illustrated in **Figure 10**.



(a) Result of RMSE

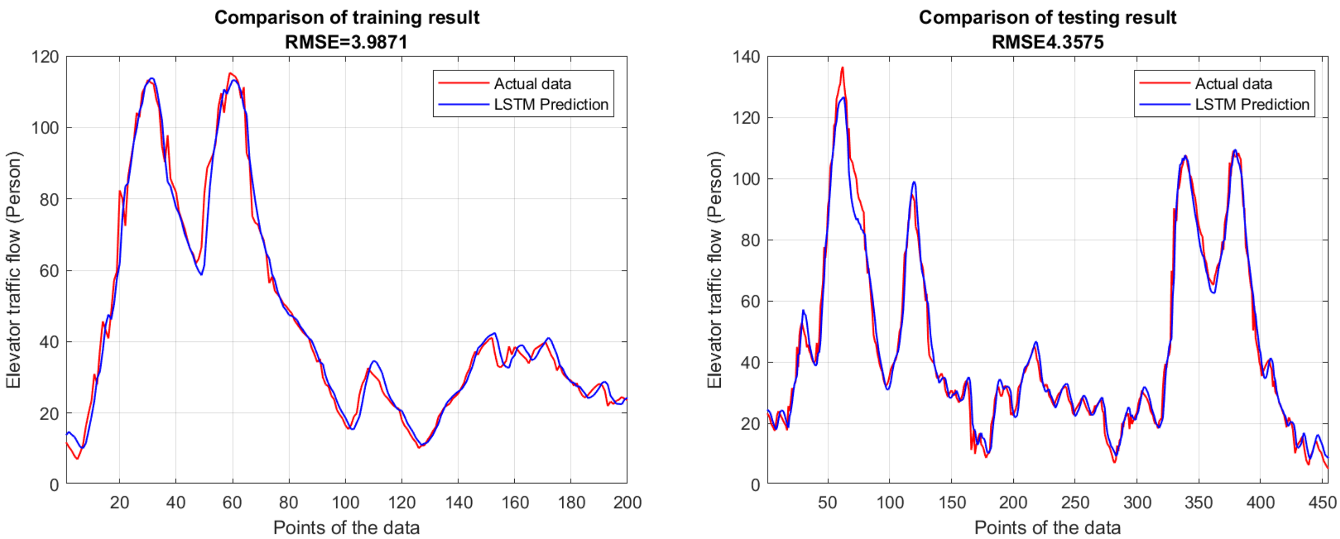


(b) Result of loss

Figure 10. Training progress of LSTM.

Referring to **Figure 10**, the RMSE and loss function exhibit signs of stability after just the first 100 epochs, indicating that the performance of LSTM has converged and further training beyond 100 epochs may not yield significant improvements in accuracy. This observation suggests that, although the model was initially designed to run for 300 epochs, the training process under the LSTM model could be effectively completed within the first 100 epochs.

Figure 11 presents a detailed comparison between the training and testing results of the LSTM used for elevator traffic flow prediction in this research. The predicted results during the training and testing phases are closely aligned with the actual elevator traffic data, underscoring the accuracy of the learning capacity of LSTM during training as well as its predictive capabilities during testing. This comparison offers insight into how effectively the LSTM has generalized from the training data to unseen data during the testing process.



(a) Training results **(b)** Testing results

Figure 11. Comparison of training and testing results of LSTM.

As the discussion of CNN emphasized, the RMSE values reflect the magnitude of the error between the predicted and actual elevator traffic flow, providing a quantitative measure of the effectiveness of the neural networks. Both the training and testing RMSE values are illustrated in **Figure 11**, offering a clear understanding of the performance of LSTM during both phases.

As depicted in **Figure 11a**, the training results for the LSTM highlight its performance during the morning traffic rush of elevator usage. The predicted results closely align with the actual elevator traffic flow, yielding a notably low RMSE of 3.9871. This small RMSE value indicates that the model effectively learned the underlying patterns in the data during the training phase, showcasing its ability to make accurate predictions based on the available input. In contrast, **Figure 11b** illustrates the testing results for the LSTM, which pertain to more complex periods as noon and end-of-day traffic rushes. While the RMSE value during testing is slightly higher at 4.3575, the predictions still reflect a strong correlation with the actual elevator traffic flow. As the discussion regarding CNN indicates, the increase in RMSE compared to the training phase is not unexpected, as testing data introduces variability that the LSTM may not have encountered during training. Nevertheless, this result underscores the predictive capability of LSTM, demonstrating that it can still deliver reliable forecasts even when applied to new, unseen elevator traffic flow.

3.3. Comparison discussion

The analysis of the prediction results for both the CNN and LSTM reveals distinct performance characteristics based on the RMSE values obtained during the training and testing phases. During the training process, the CNN demonstrates a smaller RMSE of 2.5056, indicating that it provides more accurate predictions compared to the LSTM, which has a higher RMSE of 3.9871. This suggests that the CNN is particularly adept at learning the underlying patterns within the training data, allowing it to make precise predictions regarding elevator traffic flow.

In contrast, when examining the testing results, the performance dynamics shift. The LSTM exhibits a smaller RMSE of 4.3575, while the CNN records a higher RMSE of 6.2139. This indicates that, despite its comparatively lower accuracy during training, the LSTM is better equipped to generalize its predictions to unseen data during the testing phase. The ability of the LSTM to maintain a lower RMSE under testing conditions emphasizes its effectiveness in capturing complex patterns over time, especially in the elevator traffic flow where traffic dynamics may change.

Beyond the RMSE, this research employs additional metrics to comprehensively evaluate the prediction capacity of both the CNN and LSTM neural networks. Specifically, this research incorporates the R^2 (Coefficient of Determination), MAE (Mean Absolute Error), and MBE (Mean Bias Error), providing a multi-faceted assessment of the neural networks' performance during both the training and testing processes. As summarized in **Table 4**, the R^2, MAE, and MBE enrich the analysis by providing a broader perspective on the predictive capabilities of the CNN and LSTM in this research.

Table 4. Error index.

	CNN		**LSTM**	
R^2	Training section	0.9931	Training section	0.9823
	Testing section	0.9873	Testing section	0.9936
MAE	Training section	1.9192	Training section	2.5993
	Testing section	4.0276	Testing section	3.0943
MBE	Training section	0.7817	Training section	−0.0385
	Testing section	0.8952	Testing section	−0.0723

The value of R^2 is a crucial metric that evaluates the performance of predictive neural networks, and it offers valuable insight into the extent to which the neural network explains the variance observed in the predicted outcomes. Specifically, R^2 quantifies the proportion of the total variability in the data that can be attributed to the predictions of the neural networks, making it a key indicator of the model's explanatory power, and the mathematical equation of R^2 can be expressed as follows:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{8}$$

In general, a higher R^2 value suggests that the model is effectively capturing the underlying patterns in the data, meaning that a significant portion of the variability in the dependent variable is being explained by the independent variables included in the model. Conversely, a lower R^2 indicates that the model fails to account for a considerable amount of the variance, suggesting that there may be other influential factors not captured by the model or that the model is inadequately specified.

In this research, the comparative analysis of the values of R^2 reveals important insights into the predictive performance of the CNN and LSTM as presented in **Table 4**. During the training process, the CNN exhibits a slightly larger R^2 value compared to the LSTM. This suggests that the CNN is more effective at capturing the relationships within the training data, thereby providing a more accurate prediction of elevator traffic flow during this phase. On the other hand, the results in **Table 4** also illustrate a contrasting performance during the testing process. Here, the LSTM shows a slightly higher R^2 value than the CNN, which indicates that the LSTM is more adept at generalizing its predictions to new, unseen elevator traffic flow. These findings are consistent with the former discussion on RMSE, where LSTM despite having a lower accuracy during training, proves its strength in prediction accuracy during testing. The overall results suggest that while the CNN may excel in learning from existing data, the LSTM's architecture allows it to maintain accuracy even when confronted with new data patterns.

The MAE is a vital metric in evaluating the performance of predictive models. It quantifies the average magnitude of errors in predictions, providing a clear and intuitive measure of accuracy. By expressing these errors in the same units as the predicted values, MAE allows for a direct comparison between the predicted outcomes and actual results. In general, a lower MAE indicates a better fit, as it suggests that the predictions of the model are closer to the actual values, while a higher MAE reveals a greater discrepancy, while the mathematical equation of MAE can be expressed as follows:

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{9}$$

In this research, the analysis of predictive accuracy is further elucidated through the MAE presented in **Table 4**. During the training process, the MAE indicates that the CNN outperforms the LSTM, showcasing a lower MAE value. This result emphasizes the effectiveness of CNN in accurately learning from the training data, capturing underlying patterns and trends more efficiently than the LSTM. The lower MAE suggests that the predictions generated by the CNN are generally closer to the actual outcomes, highlighting its capacity for precise model fitting during training. However, in the phase of the testing, the LSTM exhibits a lower MAE compared to the CNN, which underscores its superior ability to generalize predictions to new, unseen data. This finding suggests that while CNN excels in the training phase, LSTM is better equipped to handle variations and complexities in data that it has not encountered before. The contrasting performance of the two models, as revealed by the MAE values, reinforces earlier discussions concerning the RMSE and the R^2.

In addition to the evaluation metrics of RMSE, R^2, and MAE, this research incorporates the MBE to provide a deeper understanding of the predictive performance of CNN and LSTM as detailed in **Table 4**. The MBE serves as a crucial metric that assesses the bias inherent in the predictions made by CNN and LSTM in this research. Specifically, it indicates whether the predictions of CNN and LSTM tend to overestimate or underestimate the actual values, and the mathematical equation of MBE can be expressed as below:

$$MBE = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - \hat{y}_i) \tag{10}$$

Based on the equation above, a positive MBE suggests that the model generally overestimates the outcomes, while a negative value indicates a tendency to underestimate them.

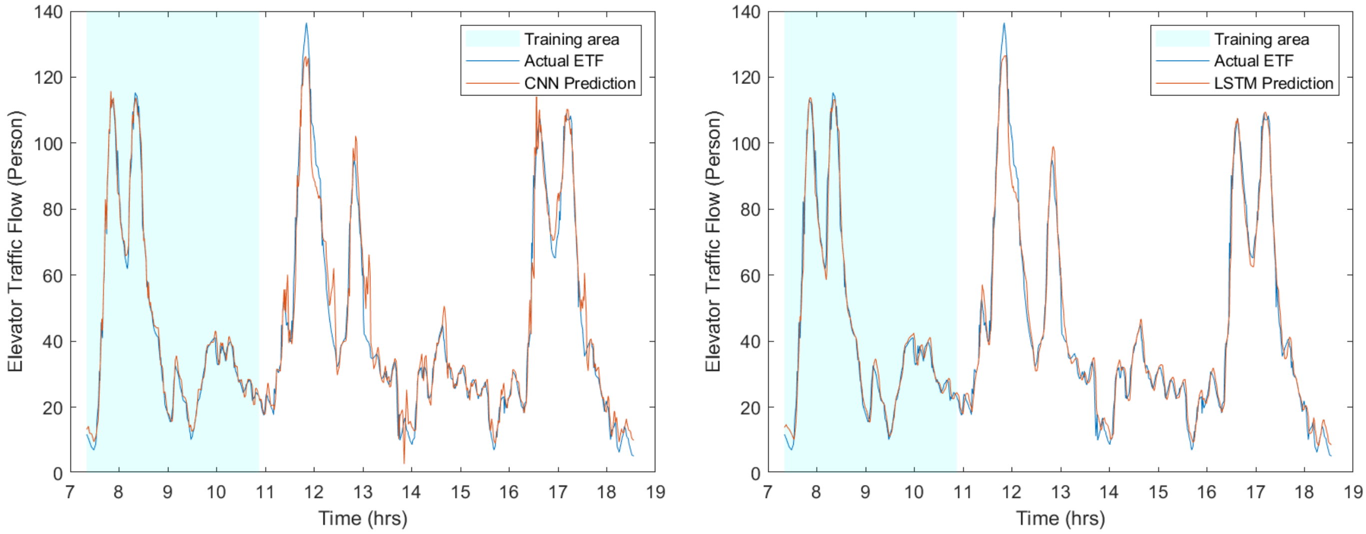
The analysis of the MBE results presented in **Table 4** offers valuable insights into the predictive behaviors of both the CNN and LSTM. The positive MBE values associated with the elevator traffic flow predictions of CNN during both the training and testing phases suggest a consistent overestimation of outcomes. This bias indicates that while CNN may effectively capture trends in the training data, it does so at the cost of inflated predictions, potentially leading to inaccuracies in actual applications. In contrast, the LSTM displays negative MBE values, indicating a tendency to underestimate the actual outcomes during both training and testing. This underestimation suggests that while the LSTM may not capture the full extent of the variations present in the data, it is at least leaning toward a more conservative approach, which can be beneficial in certain predictive contexts.

The proximity of the MBE values to 0 serves as an important marker of accuracy, with values closer to 0 indicating that the predictions are more reliable and aligned with actual outcomes. The consistent findings across MBE, along with other metrics like RMSE, R^2, and MAE, reinforce the narrative that CNN excels during the training phase. However, the LSTM demonstrates a superior ability to generalize its predictions to unseen data, effectively managing the complexities and variations that were not part of its training set. Notably, the MBE results further emphasize that the LSTM is more accurate than the CNN in predicting elevator traffic flow in this research, regardless of whether the predictions are being made during training or testing. The closer MBE values from the LSTM to 0 highlight its reliability, suggesting that it may be a more suitable approach for actual applications where accurate predictions are crucial.

4. Summarization

Figure 12 presents a comprehensive comparison between the actual ETF (Elevator Traffic Flow) and the predicted results generated by both CNN and LSTM. By reflecting on the actual ETF, **Figure 12** reveals three distinct peaks in elevator traffic, each corresponding to key times during the workday.

Referring to **Figure 12**, the first significant peak appears during the morning rush, between 7:00 and 9:00, as employees arrive at work. This period is typically the most concentrated, with a surge in demand for elevator services as building occupants converge in the morning hours. The second peak is observed around noon, between 11:00 and 13:00, likely due to lunch breaks, during which employees exit and re-enter the building for dining or other mid-day activities. The final peak occurs in the evening, from 16:00 to 18:00, as employees finish their workday and leave the office building, creating a high demand for elevator usage.



(a) Prediction results of CNN **(b)** Prediction results of LSTM

Figure 12. Prediction results comparison.

These three peaks are clearly reflected in the actual ETF trends, underscoring the typical elevator traffic behavior observed in office buildings. The actual ETF trends not only highlight the periods of high elevator demand but also serve as a benchmark for evaluating the accuracy of the predictive results in this research. Both CNN and LSTM models aim to predict these traffic flow patterns, and their ability to capture the peaks accurately is a measure of their effectiveness.

Figure 12 highlights the predictive ability of both CNN and LSTM in forecasting elevator traffic flow. The prediction results showcase the ability of the neural networks to capture the weekday daily traffic patterns within a typical office building. This emphasizes the potential of neural networks as effective tools for predicting elevator traffic flow, which is essential for optimizing elevator systems and improving building management. The accurate predictions made by CNN and LSTM suggest that neural networks can play a key role in developing more efficient and responsive elevator traffic control systems.

While CNN and LSTM demonstrate impressive predictive capabilities, a closer analysis of the prediction results reveals that the LSTM may provide slightly more accurate predictions compared to CNN. This observation can be drawn from the visual comparison between **Figure 12a,b**, which illustrate the performance of each neural network. The predictions of the LSTM appear to align more closely with the actual ETF curve, particularly in areas where small deviations are critical, such as the peaks and valleys in elevator traffic.

This conclusion is further supported by the comparison of difference rates in **Table 5**, which provides a more detailed quantitative analysis of the performances.

According to **Table 5**, the difference rates of the prediction results reveal that the maximum actual traffic is 115 people for elevator traffic flow during the morning rush. Regarding the morning rush, CNN predicts a flow of 116 people, while LSTM predicts 114 people, and this yields a difference rate of 0.87% for both models, highlighting their relative accuracy in forecasting elevator traffic flow under the training phase. One extra person predicted by CNN and one less by LSTM reflects the discussion regarding MBE. Specifically, it indicates the positive MBE associated with the CNN suggests that it generally overestimates the actual elevator traffic flow, reflecting a slight inflation in the predictions. Conversely, the negative MBE from the LSTM indicates a tendency to underestimate elevator traffic, which may lead to insufficient resource allocation during peak times.

Table 5. Difference rate of ETF.

	6:30–9:00	**11:00–13:30**	**16:00–18:00**
	(Person)	**(Person)**	**(Person)**
Actual ETF	115	137	109
CNN	116	126	125
LSTM	114	127	109
Difference Rate-CNN	0.87%	8.03%	14.68%
Difference Rate-LSTM	0.87%	7.30%	0.00%

Moreover, **Table 5** also illustrates the actual elevator traffic flow reaches a maximum of 137 people in the lunch rush period. The CNN predicts 126 people, leading to a difference rate of 8.03%, while the LSTM predicts 127 people, with a smaller difference rate of 7.30%. This indicates that the LSTM model is more accurate than CNN in predicting elevator traffic flow during the noon rush based on the testing phase. This phenomenon is also observed during the end-of-day rush, where the prediction of CNN deviates significantly from the actual elevator traffic flow, showing a larger difference rate of 14.68%. On the other hand, the LSTM model demonstrates exceptional accuracy, as there is no difference between its predictions and the actual elevator traffic flow. These findings suggest that the LSTM consistently provides more reliable predictions across different elevator traffic rush, making it a better choice for modeling elevator traffic flow in office buildings, especially during high-demand periods.

5. Conclusion

In this research, the elevator traffic flow of a typical office building during a weekday serves as the basis for the analysis and prediction process, utilizing CNN (Convolutional Neural Networks) and LSTM (Long Short-Term Memory). By employing 655 actual ETF (Elevator Traffic Flow) data points, this research aims to assess the predictive accuracy and overall capacity of both neural networks. Through a comparative analysis of CNN and LSTM, this research highlights the distinct characteristics of each neural network, exploring how CNN and LSTM handle elevator traffic flow prediction differently. By identifying strengths and limitations in the performance of CNN and LSTM, this research provides valuable insights into optimizing elevator traffic control systems in office buildings, ultimately contributing to better efficiency and user convenience. This comparative approach also aims to broaden the understanding of how neural networks can be applied in practical situations like elevator traffic management, showcasing their potential to improve urban infrastructure.

The prediction results of this research demonstrate that both CNN and LSTM are capable of accurately forecasting elevator traffic flow. Both CNN and LSTM consistently align with the actual ETF trends, showcasing their effectiveness in reflecting real-world elevator usage patterns. This research emphasizes that, given their ability to predict elevator traffic flow with high accuracy, CNN and LSTM can both be considered suitable tools for elevator traffic flow prediction work. Moreover, the accuracy of CNN and LSTM provides valuable insights that can be used to optimize elevator control systems, particularly in urban settings where efficient elevator utilization is essential. By integrating the predictive capabilities of CNN and LSTM, elevator systems can be better managed during peak usage times, reducing wait times and enhancing the overall experience for passengers.

In evaluating the effectiveness of the CNN and LSTM for predicting elevator traffic flow, this research delves into the significance of performance metrics of RMSE, R^2, and MAE. The findings indicate that while the CNN demonstrates lower RMSE, higher R^2, and lower MAE during the training phase, and LSTM consistently outperforms in the testing phase with even lower RMSE, higher R^2, and lower MAE values. This discrepancy highlights the superior ability of LSTM to generalize and maintain predictive accuracy when faced with new, unseen elevator traffic flow. The overall results suggest that, despite the strengths of CNN in learning from existing data, the architecture of the LSTM is better suited for the dynamic nature of elevator traffic flow predictions. The capability of LSTM is enabled to effectively capture temporal dependencies and adapt to variations in traffic patterns makes it a more reliable choice for real-world elevator traffic flow prediction.

Additionally, the MBE serves as a vital metric for evaluating the accuracy of predictions, with values that are closer to zero indicating higher reliability and alignment with actual traffic flows. The analysis reveals that the LSTM consistently produces MBE values nearer to zero, reinforcing its effectiveness in predicting elevator traffic flow. This accuracy is particularly crucial in urban settings where precise forecasts can lead to improved operational efficiency. Furthermore, the CNN model presents a positive MBE, suggesting that it tends to overestimate actual elevator traffic flow. Conversely, the negative MBE of LSTM indicates a tendency to underestimate traffic, showcasing a more conservative forecasting approach. This conservativeness can be advantageous, particularly in high-demand situations, as it may help avoid overloading systems and reduce user wait times.

Through an in-depth comparison of CNN and LSTM models, this research demonstrates the superiority of LSTM in predicting elevator traffic flow. It not only highlights the advantages of LSTM but also lays the foundation for future research exploring the integration of advanced neural network models into elevator traffic analysis. The findings in this research hold significant potential for transforming elevator control systems, a critical need as urban environments face growing demands. While traditional methods like increasing elevator capacity, speed, and acceleration aim to reduce elevator cycle time $v(t)$, such remodeling projects are often constrained by financial or other limitations. Instead, advancements in elevator control systems, particularly through GCS (Group Control Systems) and DDS

(Destination Dispatching Systems) offer a more feasible and impactful solution [41–44]. Including CNN and LSTM, the prediction methods by neural networks provide an opportunity to revolutionize traditional elevator control systems, enabling smarter and more efficient elevator operations. Based on the findings in this research, by integrating LSTM-based predictive models with systems like GCS and DDS, this research envisions significant improvements in traffic flow management, leading to enhanced elevator operational efficiency and a seamless passenger experience. These advancements are particularly critical in addressing the vertical traffic issues of modern high-rise buildings, ensuring that elevator systems remain capable of satisfying the demands of increasingly dense urban environments.

Author contributions: Conceptualization, MS; methodology, MS; software, MS; validation, MS and YC; formal analysis, MS; investigation, MS; resources, MS; data curation, MS; writing—original draft preparation, MS; writing—review and editing, MS; visualization, MS; supervision, MS and YC; project administration, MS; funding acquisition, MS. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare no conflict of interest.

References

1. Tchrakian TT, Basu B, O'Mahony M. Real-Time Traffic Flow Forecasting Using Spectral Analysis. IEEE Transactions on Intelligent Transportation Systems. 2012; 13(2): 519-526. doi: 10.1109/tits.2011.2174634
2. Bas E, Tekalp AM, Salman FS. Automatic Vehicle Counting from Video for Traffic Flow Analysis. IEEE Intelligent Vehicles Symposium. 2007; 392-397. doi: 10.1109/ivs.2007.4290146
3. Shi W, Kong QJ, Liu Y. A GPS/GIS Integrated System for Urban Traffic Flow Analysis. International IEEE Conference on Intelligent Transportation Systems. 2008; 844-849. doi: 10.1109/itsc.2008.4732569
4. Peppa MV, Bell D, Komar T, et al. Urban traffic flow analysis based on deep learning car detection from cctv image series. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2018; XLII-4: 499-506. doi: 10.5194/isprs-archives-xlii-4-499-2018
5. Bose A, Ioannou PA. Analysis of traffic flow with mixed manual and semiautomated vehicles. IEEE Transactions on Intelligent Transportation Systems. 2003; 4(4): 173-188. doi: 10.1109/tits.2003.821340
6. Sharma HK, Swami M, Swami BL. Speed-flow analysis for interrupted oversaturated traffic flow with heterogeneous structure for urban roads. International Journal for Traffic and Transport Engineering. 2012; 2(2): 142-152.
7. Zhao Z, Chen W, Wu X, et al. LSTM network: a deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems. 2017; 11(2): 68-75. doi: 10.1049/iet-its.2016.0208
8. Kang D, Lv Y, Chen YY. Short-term traffic flow prediction with LSTM recurrent neural network. In: Proceedings of 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC); 16-19 October 2017; Yokohama, Japan.
9. Lazaris A, Prasanna VK. An LSTM framework for modeling network traffic. Symposium on Integrated Network and Service Management (IM). 2019; 19-24.
10. Zheng Q, Zhao C. Short-term Elevator Traffic Flow Estimation with Hybrid Long Short-Term Memory Network. In: Proceedings of 2020 Chinese Automation Congress (CAC); 06-08 November 2020; Shanghai, China.
11. Shi M, Sun E, Xu X, et al. Prediction and Analysis of Elevator Traffic Flow under the LSTM Neural Network. Intelligent Control and Automation. 2024; 15(02): 63-82. doi: 10.4236/ica.2024.152004
12. Kattenborn T, Leitloff J, Schiefer F, et al. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing. 2021; 173: 24-49. doi: 10.1016/j.isprsjprs.2020.12.010
13. Aloysius N, Geetha M. A review on deep convolutional neural networks. In: Proceedings of 2017 International Conference on Communication and Signal Processing (ICCSP); 6-8 April 2017; Chennai, India.

14. Nebauer C. Evaluation of convolutional neural networks for visual recognition. IEEE Transactions on Neural Networks. 1998; 9(4): 685-696. doi: 10.1109/72.701181
15. Kuo CCJ. Understanding convolutional neural networks with a mathematical model. Journal of Visual Communication and Image Representation. 2016; 41: 406-413. doi: 10.1016/j.jvcir.2016.11.003
16. Kamilaris A, Prenafeta-Boldú FX. A review of the use of convolutional neural networks in agriculture. The Journal of Agricultural Science. 2018; 156(3): 312-322. doi: 10.1017/s0021859618000436
17. Cao M, Li VOK, Chan VWS. A CNN-LSTM Model for Traffic Speed Prediction. Vehicular Technology Conference (VTC2020-Spring). 2020. doi: 10.1109/vtc2020-spring48590.2020.9129440
18. Ranjan N, Bhandari S, Zhao HP, et al. City-Wide Traffic Congestion Prediction Based on CNN, LSTM and Transpose CNN. IEEE Access. 2020; 8: 81606-81620. doi: 10.1109/access.2020.2991462
19. Bogaerts T, Masegosa AD, Angarita-Zapata JS, et al. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. Transportation Research Part C: Emerging Technologies. 2020; 112: 62-77. doi: 10.1016/j.trc.2020.01.010
20. Fei L, Yu GX, Jian ZC. Elevator traffic flow prediction with least squares support vector machines. International Conference on Machine Learning and Cybernetics. 2005; 4266-4270. doi: 10.1109/icmlc.2005.1527686
21. So A, Al-Sharif L. Calculation of the elevator round-trip time under destination group control using offline batch allocations and real-time allocations. Journal of Building Engineering. 2019; 22: 549-561. doi: 10.1016/j.jobe.2019.01.013
22. Tyni T, Ylinen J. Evolutionary bi-objective optimisation in the elevator car routing problem. European Journal of Operational Research. 2006; 169(3): 960-977. doi: 10.1016/j.ejor.2004.08.027
23. Markos PA, Dentsoras AJ. An integrated mathematical method for traffic analysis of elevator systems. Applied Mathematical Modelling. 2022; 105: 50-80. doi: 10.1016/j.apm.2021.12.021
24. Chauhan R, Ghanshala KK, Joshi RC. Convolutional Neural Network (CNN) for Image Detection and Recognition. In: Proceedings of The First International Conference on Secure Cyber Computing and Communication (ICSCCC); 15-17 December 2018; Jalandhar, India.
25. Hossain MdA, Alam Sajib MdS. Classification of Image using Convolutional Neural Network (CNN). Global Journal of Computer Science and Technology. 2019; 13-18. doi: 10.34257/gjcstdvol19is2pg13
26. Liu Y, Pu H, Sun DW. Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. Trends in Food Science & Technology. 2021; 113: 193-204. doi: 10.1016/j.tifs.2021.04.042
27. Yin W, Kann K, Yu M, et al. Comparative Study of CNN and RNN for Natural Language Processing. Cornell University; 2017.
28. Wang W, Gang J. Application of Convolutional Neural Network in Natural Language Processing. In: Proceedings of 2018 International Conference on Information Systems and Computer Aided Education (ICISCAE); 6-8 July 2018; Changchun, China.
29. Canizo M, Triguero I, Conde A, et al. Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study. Neurocomputing. 2019; 363: 246-260. doi: 10.1016/j.neucom.2019.07.034
30. Jain AK, Grumber C, Gelhausen P, et al. A Toy Model Study for Long-Term Terror Event Time Series Prediction with CNN. European Journal for Security Research. 2019; 5(2): 289-309. doi: 10.1007/s41125-019-00061-w
31. Musaev M, Khujayorov I, Ochilov M. Image approach to speech recognition on CNN. Association for Computing Machinery. 2019. doi: 10.1145/3386164.338910
32. Huang JT, Li J, Gong Y. An analysis of convolutional neural networks for speech recognition. In: Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 19-24 April 2015; South Brisbane, QLD, Australia.
33. Huang Z, Dong M, Mao Q, et al. Speech Emotion Recognition Using CNN. Association for Computing Machinery. 2014; 801-804. doi: 10.1145/2647868.2654984
34. Sherstinsky A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena. 2020; 404: 132306. doi: 10.1016/j.physd.2019.132306
35. Muhuri PS, Chatterjee P, Yuan X, et al. Using a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) to Classify Network Attacks. Information. 2020; 11(5): 243. doi: 10.3390/info11050243

36. Al-Selwi SM, Hassan MF, Abdulkadir SJ, et al. RNN-LSTM: From applications to modeling techniques and beyond—Systematic review. Journal of King Saud University-Computer and Information Sciences. 2024; 36(5): 102068. doi: 10.1016/j.jksuci.2024.102068
37. Schaefer AM, Udluft S, Zimmermann HG. Learning long-term dependencies with recurrent neural networks. Neurocomputing. 2008; 71(13-15): 2481-2488. doi: 10.1016/j.neucom.2007.12.036
38. Zhang S, Liu C, Jiang H, et al. Nonrecurrent Neural Structure for Long-Term Dependence. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2017; 25(4): 871-884. doi: 10.1109/taslp.2017.2672398
39. Zhang P, Li C, Peng C, et al. Ultra-Short-Term Prediction of Wind Power Based on Error Following Forget Gate-Based Long Short-Term Memory. Energies. 2020; 13(20): 5400. doi: 10.3390/en13205400
40. Alizadeh B, Ghaderi Bafti A, Kamangir H, et al. A novel attention-based LSTM cell post-processor coupled with bayesian optimization for streamflow prediction. Journal of Hydrology. 2021; 601: 126526. doi: 10.1016/j.jhydrol.2021.126526
41. Ruokokoski M, Sorsa J, Siikonen ML, et al. Assignment formulation for the Elevator Dispatching Problem with destination control and its performance analysis. European Journal of Operational Research. 2016; 252(2): 397-406. doi: 10.1016/j.ejor.2016.01.019
42. Sorsa J, Ehtamo H, Kuusinen JM, et al. Modeling uncertain passenger arrivals in the elevator dispatching problem with destination control. Optimization Letters. 2017; 12(1): 171-185. doi: 10.1007/s11590-017-1130-0
43. Jamaludin J, Rahim NAbd, Hew WP. An Elevator Group Control System With a Self-Tuning Fuzzy Logic Group Controller. IEEE Transactions on Industrial Electronics. 2010; 57(12): 4188-4198. doi: 10.1109/tie.2010.2044117
44. Hanif M, Mohammad N. Metaheuristic algorithms for elevator group control system: a holistic review. Soft Computing. 2023; 27(21): 15905-15936. doi: 10.1007/s00500-023-08843-0