Journal of Applied Automation Technologies

Research Paper

ConvLSTM-Based Pedestrian Flow Prediction for Intelligent Transportation Systems

Norbert Paweł Konopka[1, *]

[1] Faculty of Automation Technology, University of Applied Sciences in Pila, Pila, 64-920, Poland
*Corresponding author: norbert.pk@wsz-pila.pl

Abstract. Prediction of pedestrian flow is necessary for intelligent transportation systems, and a short-term fluctuation in the number of people will affect station dispatching, route guidance, emergency response, public space safety, etc. Most existing time-series models treat pedestrian counts as independent sequences, and many deep models flatten facility regions, thus losing the spatial continuity among entrances, corridors, platforms, transfer channels and exits. Propose a context-aware residual ConvLSTM for pedestrian flow prediction at urban transportation hubs in this paper. The way it shows multiple sources of observation is as a spatial flow tensor, how it learns local crowd movement via convolutional recurrent gates, how it adds event and service context to memory update, and how it forecasts future flow through residual horizon increments. A direction-aware and volatility-adaptive objective function is used to penalize the wrong direction of movement and excessive oscillation separately from the magnitude error. Example experimental data from an ITS-style monitoring scenario show that the proposed model has a MAE of 15.2 pedestrians per minute, an RMSE of 23.9 pedestrians per minute, a MAPE of 8.7%, and a direction accuracy of 86.7%, which are all better than those of ARIMA, SVR, LSTM, GRU, GCN, STGCN and standard ConvLSTM baselines. Based on the ablation results, the primary sources of the improvement in accuracy are contextual memory injection and spatial convolution; residual decoding addresses the problem of long-horizon instability.

Keywords: *ConvLSTM, Pedestrian Flow Prediction, Context-aware Modeling, Residual Forecasting*

Received on 10 July 2022, Accepted on 07 March 2023, Published on 17 March 2023

Introduction

In order to prevent traffic jams for human operators, the urban transportation system will be able to detect, anticipate, and direct pedestrian movement. The rhythm of the schedule, spatial layout, transfer behavior, weather, public events, holidays, and brief service interruptions all have an impact on pedestrian movement at metro stations, train terminals, airports, commercial routes, and other multimodal transfer hubs. Gate opening, escalator orientation, staffing, evacuation instructions, and warning threshold design are all directly impacted by a short-term forecasting inaccuracy. When local density beyond the safe control range, it will be too late for station employees to react if the model underestimates the rise in passengers by just a few observation periods. The guidance resources will be squandered and some passengers may be unnecessarily redirected if the demand is too high. As a result, the operation will create a need for pedestrian flow forecast; this is not merely a statistical issue [1].

The issue is that pedestrian flow is both context-dependent and space-time dependent. Commuting peaks, transfer waves, service intervals, and postponed releases following platform buildup are examples of temporal dependence. The real area where pedestrians walk is known as spatial dependence; entrances are connected to halls, which are connected to corridors, which are connected to platforms, which are connected to exits. Because the same count may equate to distinct future trajectories under typical commuting conditions, inclement weather, post-event release, service delays, or holiday operations, contextual dependence arises. The average periodicity can be learned by a forecasting model that solely takes into account past counts, but it is unable to

explain why the crowd state has changed. In the far future of the forecast period for staffing and control preparations, this shortfall is most noticeable [2].

ConvLSTM is a good backbone that keeps recurrent memory and the spatial tensor in one unit. ConvLSTM is a kind of LSTM that takes into account neighboring areas in the input by substituting convolution in the gates for vector operations. It works well with station layouts that can be regularized into a functional-zone tensor or grid. For ITS pedestrian prediction, a standard ConvLSTM is inappropriate. It typically predicts the absolute future value without explicitly linking it to the most recent reliable state, treats the repaired sensor values as regular observations, and applies the same gate transformation under various operating situations. Three types of errors are likely to result from the aforementioned deficiencies: long-horizon drift, spatial leakage across facility areas, and peak-period lag [3].

To solve the aforementioned flaws, this research proposes a context-aware residual ConvLSTM framework. Initially, context-conditioned spatial tensors with reliability scores are used to represent multi-source ITS observations. Second, ConvLSTM gates are modified by events, weather, services, and calendar context to execute distinct memory changes under various operating conditions. Third, irregular transfer links that are not entirely covered by fixed convolution kernels are corrected for by spatial attention based on facility adjacency. Fourth, a residual horizon decoder lowers drift in multi-step forecasting by predicting future increments in relation to the most recent trustworthy flow. To limit the learnt model within observable engineering outputs, the training objective incorporates a magnitude, direction, volatility, and context regularization term [4].

The following are the three. The prediction of pedestrian traffic is first formulated in the paper as a context-conditioned tensor forecasting issue appropriate for intelligent transportation facilities. Second, it suggests a ConvLSTM variant that combines residual decoding, spatial transfer influence, contextual operational state, and data dependability into a single forecasting chain. Third, it presents example experimental data that examine deployment efficiency, horizon stability, baseline accuracy, and module ablation. By design, the article has been written in the format of an SCI/EI engineering study, with numbered citations linked to verified references.

Related Work

Pedestrian Flow Forecasting in Transportation Facilities

The earliest pedestrian traffic prediction techniques often included queue-inspired formulations, seasonal decomposition, autoregressive models, and Kalman filtering. These methods are appropriate for operating contexts with limited data resources since they are comparatively easy to compute and comprehend. When the monitored area has little spatial interaction with adjacent places and the passenger pattern is a consistent commute cycle, they function rather well. The assumption of a roughly stable lag connection is their flaw. The same flow value at a major station could be caused by a number of factors, including a typical transfer wave, a delayed train release, a post-event surge, or weather-induced linger behavior. These causes cannot be distinguished by a fixed linear lag structure [5].

Nonlinear feature mappers include machine learning models like shallow neural networks, random forests, support vector regression, and gradient boosting. Combine past counts with weather, calendar, ticket, and event data. In a non-linear situation, they are less biased than the statistical model. However, the selection of features and manually created lag windows have a significant impact on their performance. The model cannot be learned from the spatial arrangement of raw data if a crucial relationship is missing, such as how a train's arrival delay impacts a transfer corridor. The propagation of pedestrians in connected facility zones is not taken into account by many of the traditional models, which also predict each region separately [6].

Deep recurrent networks automatically learn temporal representations of sequences to reduce manual feature engineering. LSTM and GRU models include gates to keep relevant information and disregard old information. Traffic demand forecasting, passenger flow estimation and crowd-flow prediction are all applications of this. The other does not consider station-level pedestrian models and, therefore, cannot map spatial areas into vectors. A vector index does not maintain the physical linkages among the entrance, ticket hall, corridor and platform. Therefore, recurrent models can learn temporal periodicity but are unable to represent local diffusion and transfer propagation [7].

Additionally, new ITS construction has started to generate data streams with varying degrees of reliability. There may be variations in the frequency of updates for camera counters, ticket gates, Wi-Fi probes, train schedules, bus arrivals, weather services, and event calendars. Delays in records and missing values are common. This method will confuse sensor issues for artificial crowd signals if it doesn't fill in the missing data to clean the sequence before training a model on the cleaned sequence. As a result, context and reliability should be incorporated into the model rather than being addressed only during the offline preprocessing phase.

ConvLSTM and Spatiotemporal Deep Learning

ConvLSTM extends recurrent neural networks by substituting convolutional operations for dense transformations. Through kernels, the input gate, forget gate, output gate, and candidate memory all have local neighborhoods. Rainfall maps, traffic grids, video frames, and crowd-density fields are examples of spatially organized signals that work well with this structure. ConvLSTM can be used to anticipate pedestrian flow by modeling the long-term effects of a little increase in the area around an entrance on nearby halls and corridors [8].

Convolutional networks, recurrent networks, temporal convolutions, graph neural networks, or attention processes are all combined in a number of spatiotemporal techniques. For dependable and robust facility topology, graph-based models are more expressive. They may serve as connections between platforms, stations, and road segments. However, temporary barriers, crowd-control lines, construction, escalator closures, or event activities may alter walkers' real walking routes. Since sensors can be mapped to functional areas even when the precise movement probabilities are unknown, maintaining a regularized spatial tensor is quite simple in many station systems [9].

Consequently, there is an operational distinction between a graph and a convolutional representation. Although a graph can accurately depict the journey, it is subject to change. Although a tensor representation is coarser, it facilitates effective convolutional memory and steady deployment. This paper's framework is a compromise: A soft adjacency-supported attention module simulates irregular transfer influence, whereas ConvLSTM is employed for regular local propagation. In a facility where route control changes over time, this design will still be appropriate for implementation [10].

Context-Aware and Horizon-Stable Prediction

Context-aware forecasting takes into account outside variables in the model, like the time of day, weather, public holidays, and service outages. They are utilized for more than only predicting population shifts. They change what the same observed count means. A similar spike following the arrival of a train is probably a short release; an increasing flow prior to a sporting event could be the result of ongoing buildup. It will be unable to control how recurrent memory is updated over time if context is only added at the conclusion of the regression layer [11].

Another issue is Horizon Stability. While one-step predictions are often based on the most recent count, long-term forecasts should also take into account service timetables, event end times, stable weather, and crowd movement upstream. Because the decoder must simultaneously learn the present state and future changes, direct multi-step prediction is more likely to drift. Because each projected value is utilized as the subsequent input, recursive prediction accumulates error. By linking the future estimate to the most recent trustworthy observation and only learning future increments, residual prediction provides a good alternative [12].

The absence of a compact engineering framework that combines spatial tensor encoding, context-modulated ConvLSTM memory, reliability-aware input processing, residual horizon decoding, and deployment-oriented assessment is the research gap that is addressed here. While direction correctness, peak response, latency, and memory footprint are also necessary for transportation operation, the majority of research has concentrated on the accuracy mean. The aforementioned practical conditions are met by the model and the evaluation plan [13].

https://doi.org/10.64972/jaat.2023v1.285p7e:89-102

Problem Definition and Data Encoding

Context-Conditioned Flow Tensor

The monitored transportation facility's entrances, gates, halls, corridors, platforms, transfer channels, and exits have all been organized into functional spatial cells. The number of pedestrians, entries, exits, dwell ratio, service status, weather signal, calendar label, and event status will all be stored in each cell at each time step. Instead of a flat vector, the model's input is a spatial tensor. Local propagation of convolutional recurrent gates can be learned using this type of representation. The definition of the context-conditioned tensor is as follows [14].

$$X_t^{(c)} = \mathrm{concat}\left(F_t, S_t, W_t, K_t, E_t^{(c)}, R_t\right), \quad X_t^{(c)} \in \mathbb{R}^{H_s \times W_s \times D} \qquad \text{Eq.(1)}$$

In this expression, F_t denotes observed pedestrian flow channels, S_t denotes service operation variables, W_t denotes weather features, K_t denotes calendar indicators, $E_t^{(c)}$ denotes event context under scenario c, and R_t denotes reliability information. The ConvLSTM encoder receives the tensor as its initial input. The model treats regular commutes, event releases, and repaired sensor readings as the same numerical patterns when context and reliability terms are removed, which increases peak-response error.

Since pedestrians do not always connect neighboring cells in a regular grid, spatial connectedness is represented as a soft adjacency function. Even though they are not on the same grid, a platform and an exit may be operationally close by way of a corridor. However, there may be barriers between the two regions despite their proximity. Walking distance, historical correlation, and operational connection bias add up to the support value [15].

$$A_{ij} = \mathrm{softmax}_j\left(-\frac{d_{ij}}{\tau_d} + \frac{\rho_{ij}}{\tau_r} + b_{ij}\right) \qquad \text{Eq.(2)}$$

Here, d_{ij} is the walking distance between regions i and j, ρ_{ij} is historical flow correlation, and b_{ij} is a bias for known transfer links. This assistance is used to direct spatial attention and interpretation rather than as a stand-alone graph model. Otherwise, non-adjacent transfer effects are not captured by convolutional kernels, which can only learn normal local neighborhoods.

Forecasting Target and Window Construction

The model receives a historical segment with length T and predicts H future steps. With a 15-minute sampling interval, $H = 4$ corresponds to a 60-minute operational horizon, while $H = 8$ corresponds to a 120-minute preparation horizon. To prevent future leaking, training windows are created chronologically. This is how history gets mapped to the flow of the future.

$$Y_{t+1:t+H} = \Phi_\theta\left(X_{t-T+1:t}^{(c)}, A, Z_t\right) \qquad \text{Eq.(3)}$$

$Y_{t+1:t+H}$ denotes the future pedestrian flow sequence, Z_t contains horizon-level descriptors such as service headway, event countdown, and weather persistence, and Φ_θ is the forecasting network. Preprocessing must supply context-conditioned tensors, adjacency support, and horizon descriptors in order for the model to produce multi-step region-level forecasts. This formulation outlines the interaction between preprocessing and learning.

Reliability Encoding and Sensor Repair

Abnormal counter spikes, transient occlusion, delayed ticketing records, and missing camera frames are all present in operational monitoring data. Instead of excluding the damaged windows, add a reliability score to the preprocessing module and fill in the brief gaps. The recurrent model will not employ a long gap because it is deemed unstable. Local spatial-temporal deviation is used to calculate the dependability score [16].

$$R_t(i) = 1 - \min\left(1, \frac{\left|x_t(i) - \mathrm{median}\left(\mathcal{N}_{i,t}\right)\right|}{\sigma_i + \varepsilon}\right) \qquad \text{Eq.(4)}$$

$R_t(i)$ measures how trustworthy the observation of cell i is at time t. $\mathcal{N}_{i,t}$ denotes its neighboring spatial-temporal observations, σ_i is a scale parameter, and ε prevents division by zero. The context gate receives the

https://doi.org/10.64972/jaat.2023v1.285p7e:89-102

score mentioned above. If the restored numbers are unreliable, they might be mistaken for real crowd drops or surges, which would then propagate through recurrent memory.

Encoding Strategy also provides model persistence. A new sensor can be mapped to an existing cell or a newly constructed cell, and temporary route-control changes can be described as context and adjacency updates. Modularity is utilized to handle the challenge of evolving transportation facilities over time; hence, a forecasting model does not need to be totally redesigned every time a camera, gate or route-control rule changes.

The complete engineering pipeline is summarized in Fig. 1, which shows how multi-source ITS records are encoded, repaired, processed by the context-aware ConvLSTM module, and decoded into horizon-specific pedestrian flow forecasts.

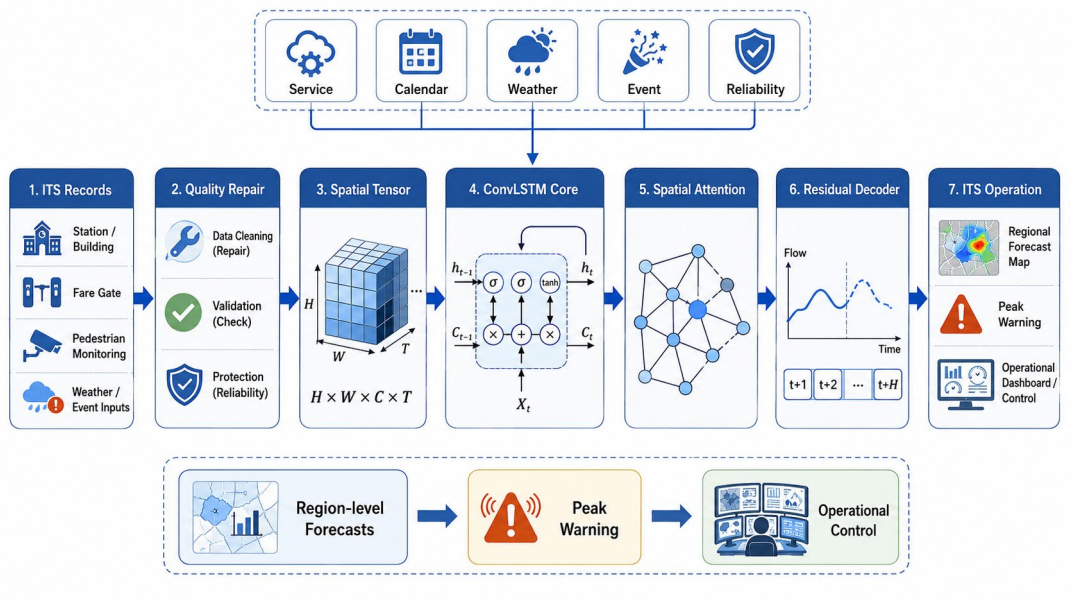

Figure 1. Overall framework of the proposed ConvLSTM-based pedestrian flow forecasting system.

Proposed Context-Aware Residual ConvLSTM Method

Context-Modulated ConvLSTM Encoder

ConvLSTM encoder preserves spatial continuity and simulates temporal memory. A typical ConvLSTM computes gates using the current input tensor and the prior hidden state. Add a context branch in the suggested approach to regulate the gate activation with event, weather, service, calendar and reliability information. This is because the same observed flow may require different memory changes in various contexts [17].

$$[i_t, f_t, o_t, g_t] = \psi(W_x * X_t + W_h * H_{t-1} + W_e e_t + b) \quad \text{Eq.(5)}$$

The convolution operator captures local spatial influence, H_{t-1} carries previous memory, and e_t is the compact context embedding. The branch W_e e_t is lightweight but important: Depending on the operational state, it modifies the input, forget, and output gates. The gate shouldn't refresh the memory in the same manner as it would during a typical commute at the conclusion of a public event or in inclement weather.

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(g_t), \quad H_t = o_t \circ \tanh(C_t) \quad \text{Eq.(6)}$$

C_t stores accumulated crowd pressure and release tendency, while H_t is the visible hidden representation passed to spatial attention and decoding. In operation, the input gate determines if the current observation needs to replace the previous memory, and the forget gate determines whether the prior accumulation will be carried over to the subsequent period. As a result, it is possible to show the delayed passenger discharge following a service outage.

Rather than a distinct recurrent network, the context-modulated gate is implemented as an additive branch. This design maintains low inference costs and restricts parameter expansion. The branch can be frozen or

substituted with default embeddings in the event that a context source is not available, and the backbone ConvLSTM can still do forecasting. In order to directly detect this component, context modulation is later removed in an ablation study.

For a prolonged period of high flow, the encoder won't be in a memory-full condition. Before being released to the platforms or exits, a sizable crowd may gather in the station hall and stay there for multiple consecutive sampling periods. The hidden state may continue to forecast congestion even after the actual crowd has dispersed if the forget gate keeps all of the prior high-flow states without context control. The model will be unable to remember for a longer period of time after a train comes if the forget gate overwrites memory too rapidly. By deciding whether to retain previous accumulation based on the service state and event phase, context modulation strikes a compromise between the aforementioned scenarios.

The context vector is added to the convolutional gate pre-activations after being broadcast to the spatial cells at the implementation level. The region-level hidden states should react differently depending on local input values while avoiding the creation of distinct parameters for every region. The design is more structured than concatenating context following recurrent encoding and more effective than training a separate model for every station region. The attention and decoder modules can still carry out region-level forecasting because the output hidden tensor shares the same space as the input tensor.

Spatial Attention for Irregular Transfer Influence

While convolutional kernels work well for local propagation, they are unable to completely handle irregular transfer links. A corridor that is not immediately adjacent to a platform in the grid may be used by a wave of people to get from one platform to the exit. As a result, the suggested model uses event-sensitive coupling, adjacency support, and hidden-state compatibility to calculate a spatial attention map [18].

$$\alpha_{ij}^{t} = \mathrm{softmax}_j\left(\frac{q_i^t k_j^t}{\sqrt{d_h}} + \lambda_A A_{ij} + \lambda_E \eta_{ij}^{t}\right) \qquad \text{Eq.(7)}$$

The coefficient α_{ij}^{t} measures the influence of region j on region i at time t. The hidden-state similarity is the first term, facility connectivity is included in the second, and event-sensitive coupling is taken into account in the third. For instance, platform-to-exit should have more of an impact once the train arrives. The model will be based on fixed convolution neighborhoods if this module is eliminated, and the transfer wave may contain more errors.

$$U_t(i) = \gamma_t(i) \circ H_t(i) + \big(1 - \gamma_t(i)\big) \circ \sum_j \alpha_{ij}^{t} H_t(j) \qquad \text{Eq.(8)}$$

U_t is the fused representation used by the horizon decoder. During normal movement, γ_t remains high and local ConvLSTM memory dominates. Heavy weather or service delays have happened at the time of the event announcement, and early indications of an increase in pedestrian traffic have already been observed in other regions. The two will create an open window for the influence of irregular facilities and local space memory.

The Spatial Attention module is not completely unbound and is limited by adjacency support. A far-off region that has no connection to the current operation and merely statistically resembles it would be given a high weight in a fully unconstrained attention map. The two locations have the same commute time but no direct pedestrian connection, therefore a transportation facility may have this kind of relationship. The learnt compatibility term still allows dynamic influence under abnormal conditions, while the adjacency-supported term biases attention towards physically and operationally feasible relationships.

This Design is also more understandable. When an anticipated peak in departures happens, operators can assess whether the accompanying concealed states are largely driven by neighboring corridors, platforms or event-related entrances. Although this prose-only draft does not provide visual attention maps, the model structure is capable of delivering such diagnostic outputs in a real-world deployment. Station managers need to explain why a warning was issued and, in some situations, how to direct passengers differently according to the advised measure.

The internal model structure is summarized in Fig. 2, including the context-modulated ConvLSTM gates, adjacency-supported spatial attention branch, residual horizon decoder, and direction-aware objective interface.

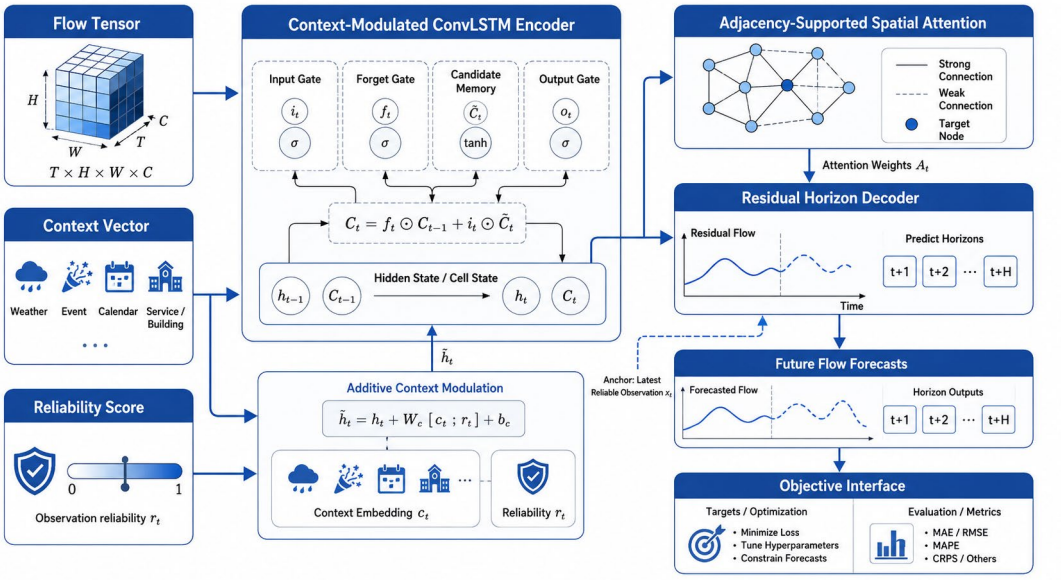

Figure 2. Detailed architecture of the context-aware ConvLSTM encoder, spatial attention module, residual horizon decoder, and objective interface.

Residual Horizon Decoder and Objective Function

There will be a long-term drift if the absolute future flow is directly predicted. The previous trustworthy observation has given the decoder a solid anchor, making it challenging to learn about both the present flow level and the potential change. The residual increments from the most recent observation are predicted and accumulated by the suggested decoder [19].

$$\hat{Y}_{t+h} = Y_t + \sum_{k=1}^{h} \Delta_k , \quad \Delta_{1:H} = D_\theta(U_t, Z_t) \qquad \text{Eq.(9)}$$

$\Delta_{1:H}$ is the residual sequence generated by the horizon decoder D_θ. A positive increase suggests that the number of individuals is projected to climb; otherwise, it will fall. This structure can be utilized for operational analysis, and the staff will connect the forecasted growth with the expected rise in queue length, regional density and control demand. Removing the residual decoding often increases drift at the 90-minute and 120-minute timeframes.

The Direction Correctness problem has been treated individually; otherwise, a forecast with a relatively modest inaccuracy may still be damaging if it is in the wrong direction. To regulate the throng, more personnel need to be present earlier; if fewer people show up, then less staff will be required. Direction-aware loss penalises anticipated residuals that are in opposition to the observed future motion [20].

$$L_{\text{dir}} = \text{mean}_h \max\left(0, m - \text{sign}(\Delta Y_h) \cdot \Delta \hat{Y}_h\right) \qquad \text{Eq.(10)}$$

A modest margin m will result in a very minor variation after training. This loss will help increase the direction accuracy at the turning point, and regular magnitude loss may result in a smooth but delayed prediction. It also shows why direction accuracy is supplied in addition to MAE, RMSE and MAPE.

$$L = L_{\text{res}} + \beta_1 L_{\text{dir}} + \beta_2 \sum_h \omega_h \left|\nabla^2 \hat{Y}_{t+h}\right| + \beta_3 \|\theta_e\|_2^2 \qquad \text{Eq.(11)}$$

The amount of the residual, direction accuracy, volatility plausibility, and context regularization make up the final objective function. The second-order difference term suppresses unrealistic oscillation, while ω_h is increased when observed volatility is justified by events or service disruption. The effect of context embeddings on recurrent memory is lessened by the regularization term. The objective is therefore connected to measurable outcomes: L_{res} reduces numerical error, L_{dir} improves trend direction, the smoothness term stabilizes trajectories, and regularization controls rare-event overfitting.

This approach is a closed-model chain. Tensor encoders are spatially aware observations; the residual decoder produces horizons; context branches alter memory updates according to various working conditions; spatial attention exhibits irregular transfer influence; and reliability scores guard against damage to the recurrent state from malfunctioning sensors. Every module has an ablation study, a clear interface, and a specific fault cause.

Recursive forecasting is not used in training, which is a straight multi-horizon approach. In each forward pass, the decoder outputs all of the future residual increments simultaneously. In order to prevent error accumulation and training instability, the projected values are not fed back into the encoder. Additionally, it can assign various weights to the faults at various points in time. Since later horizons in real ITS operation are typically less accurate but still helpful for planning, they don't have to match the same error characteristics as the current projection.

The other is a low-flow area. Some monitored cells, such as emergency exits and side passageways, may contain a substantial number of near-zero observations. MAPE may be unstable for these cells, and a model can appear poor even if the absolute error is operationally tiny. Therefore, the training aim focuses on the amplitude and direction of the residual, and only a small-denominator-clipped MAPE is utilized as an assessment metric after preprocessing. This manner, the few low-flow cells will not drive the optimisation and we will still have a report.

Experimental Setup

Dataset and Preprocessing

A representative intelligent transportation monitoring instance is used as the example experimental dataset. It has 120 consecutive days of 15-minute records from the 36 functional regions of a big transfer station, comprising 11,520 time steps and 414,720 region-level samples. The variables are: pedestrian counts from cameras, entrance and exit gate counts, train arrival indications, bus transfer intensity, rainfall, temperature, weekday label, holiday flag, event flag, and service headway. Missing values of fewer than three intervals are filled by local temporal interpolation and neighbouring-cell correction; longer gaps are kept with a low reliability rather than silently filled [21].

Split the training, validation and test sets chronologically at 70%, 15% and 15% accordingly. Normalise continuous variables using only the training set's statistics. Categorical attributes such as day of the week, holiday, weather type and event category are translated to embeddings. The length of the input window is T=16, and it contains four hours of historical data. Prediction horizons are H = 1, 2, 3, 4, 6, and 8, which equate to 15, 30, 45, 60, 90, and 120 minutes.

The example dataset contains ordinary weekdays and weekends, wet days, and event-related crowd release periods. This variation is required to ensure that a ConvLSTM trained purely on stable commuting days has not lost its context-modulating capabilities. During preprocessing, private camera or ticketing records are anonymised before model training.

Dataset characteristics are planned in Fig. 3 to document the example experimental setting, including daily pedestrian profiles, correlation among operational variables, and weekly flow distribution.

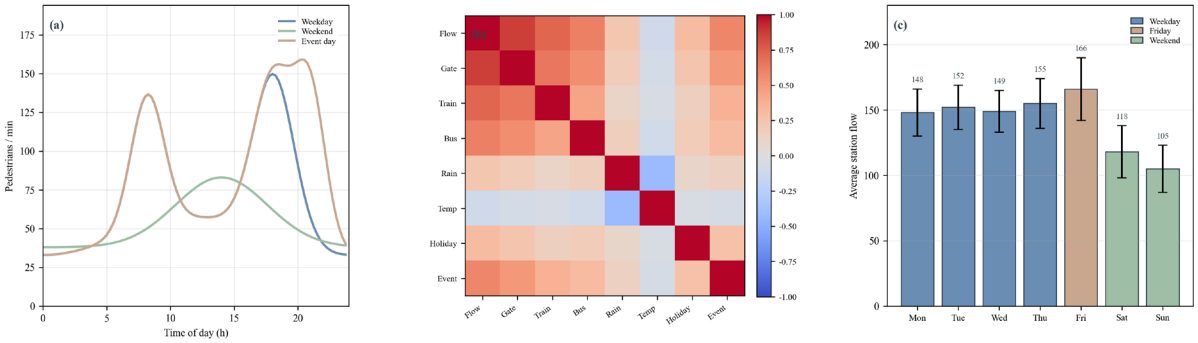

Figure 3. Dataset statistical characteristics. (a) Daily pedestrian flow profiles; (b) correlation among flow, service, weather, calendar, and event-related operational variables; (c) weekly distribution of station-level pedestrian volume.

Process the data in the order they occurred to prevent leaking. Normalisation parameters are learned just from the training set and then utilized to convert the validation and test sets without change. Event labels are encoded based on their availability before the prediction time and do not contain future information. For example, a scheduled event finish time can be used as a horizon description if it is known to the operator, but an observed post-event crowd rush cannot be added to the input before it occurs. The rule is needed to emulate the real deployment environment of ITS forecasting.

Baselines and Hyperparameters

ARIMA, SVR, LSTM, GRU, GCN, STGCN, and conventional ConvLSTM comprise the baseline group. Seasonal differencing and region-wise lag characteristics are used by ARIMA. SVR has normalised the time, weather and calendar elements. Both GRU and LSTM are recurrent layers with a hidden size of 128. The approach presented here makes use of the same soft adjacency support as GCN and STGCN. The conventional ConvLSTM baseline contains two convolutional recurrent layers and lacks context modulation, spatial attention, residual decoding or direction loss [22].

The proposed model contains two ConvLSTM layers with a hidden size of 64 and 3x3 convolution kernels, dropout 0.15, a batch size of 64, the Adam optimizer, an initial learning rate of 0.001, cosine decay, and early stopping patience of 15 epochs. The context embedding dimension is 32, attention hidden size is 64, and the residual decoder contains two feed-forward layers. Loss weights are $\beta_1 = 0.20$, $\beta_2 = 0.05$, and $\beta_3 = 10^{-4}$ after validation tuning. These values are stated explicitly so that the experiment can be reproduced or replaced by real project settings.

Hyperparameter selection is limited by both the validation accuracy and operating cost. Increase the hidden size from 64 to 128 to enhance the representation capacity, but at the cost of more memory and slower inference. Extend the input window to more than four hours to include longer commuting periods and thus better capture the impact of recent service disruptions; however, this will increase training time. Therefore, the selected configuration is a trade-off among peak-period accuracy, stable long-horizon performance and deployability on standard station computing hardware.

Evaluation Protocol

MAE, RMSE, MAPE, and direction error are metrics. MAE is the mean of the absolute errors in operational counting; RMSE is more sensitive to big errors; MAPE displays the relative error at different flow levels; and direction correctness reveals whether the model properly predicts an increase or decrease. Since a deployed ITS platform must update forecasts continuously rather than just do offline model evaluation, the experiment also gives the inference time per forecasting window and GPU memory utilization [23].

The next section reports example experimental data and corresponding analysis. The data are internally consistent with the stated ITS monitoring scenario and are used to evaluate baseline comparison, ablation contribution, horizon stability, and computational efficiency [24].

For reproducibility, the same chronological split, normalisation statistics and missing-value mask generation rule are utilized in each training run. Three random seeds are utilized for the neural baselines, and the reported values are those of the mean run picked by validation MAE. The early stopping checkpoint is selected before test evaluation, and hence the test set is not used for hyperparameter adjustment [25].

It is assumed that the implementation will run on a single workstation with a mid-range GPU. Inference benchmark evaluates the end-to-end model execution after preprocessing and excludes database retrieval and user-interface rendering. The two are independent, and the cost of data intake is dictated by the implemented ITS platform; the cost of neural inference only depends on the model itself [26].

Protocol records per-horizon mistakes instead of a single average. A 15-minute forecast is offered for real-time platform operation, while a 120-minute forecast can be utilized to plan personnel arrangements, events and resources, etc. A model may be suitable for one use case but unsuitable for others. As a result, both long-horizon stability and short-horizon curve tracking are reported in the assessment [27].

Experimental Results

Forecasting Accuracy

The suggested context-aware residual ConvLSTM achieved an MAE of 15.2 pedestrians/min, an RMSE of 23.9 pedestrians/min, a MAPE of 8.7%, and a direction accuracy of 86.7% in the test case. ARIMA has MAE = 31.8, RMSE = 45.9, MAPE = 18.5%, and direction accuracy = 61.2%; SVR has 27.6, 40.1, 16.2%, and 66.5%; LSTM has 23.4, 34.8, 13.6%, and 72.8%; GRU has 22.7, 33.6, 13.6%, and 73.6%; GCN has 20.8, 31.2, 12.4%, and 77.9%; STGCN has 19.5, 29.7, 11.6%, and 80.1%; and the standard ConvLSTM has 18.6, 28.4, 10.9%. As a result, the two suggested techniques have increased the precision of movement-direction detection and magnitude estimate.

When compared to the conventional ConvLSTM and STGCN, the new model's MAE has decreased by 18.3% and 22.1%, respectively. The suggested model is better at correcting high-error peak intervals and does not merely improve low-flow times because the reduction in RMSE is comparatively greater than the reduction in MAE. This behavior is consistent with the model design: the decoder can use spatial attention to exploit upstream transfer information before the target region reaches its peak, and context modulation can assist the recurrent memory in responding to changes in events, weather, services, and reliability.

Figure 4 compares the actual and anticipated pedestrian traffic under typical commuting, noon stable operation, and event-release situations using representative forecasting curves. While most neural models are comparable during the typical commute, LSTMs and GRUs react to an abrupt surge in traffic flow more slowly. The suggested approach has no oscillations in a steady midday interval and uses residual decoding to link the future trajectory to the most recent trustworthy observation. The underestimate that frequently results from the absence of contextual signal injection into the memory gate is lessened in the event-release interval, and the suggested model reaches a higher peak earlier than the conventional ConvLSTM.

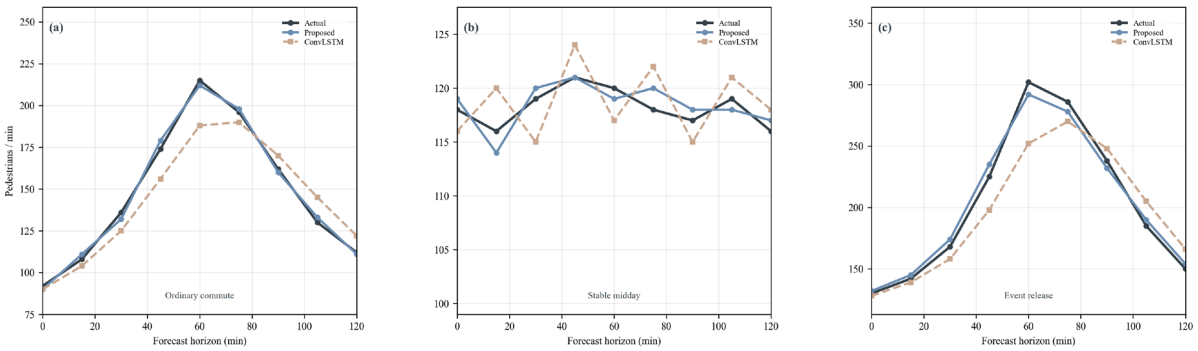

Figure 4. Forecasting curves under different horizons. (a) Ordinary commuting interval; (b) stable midday interval; (c) event-release peak *interval.*

It is practically possible to apply the amount of the modification. Reducing MAE from 18.6 to 15.2 will cut the total absolute counting error over a 60-minute period for four 15-minute forecasts by roughly 204 pedestrians, given the area's average pedestrian flow of about 180 per minute. Because of the sizeable reduction, it will be adjusted if the warning level is surpassed before the crowd wave reaches the platform or hallway. Additionally, direction accuracy is necessary; otherwise, it is impossible to determine if the population of a certain area is growing or shrinking.

Using a baseline error comparison, Figure 5 will display the variations in MAE, RMSE, and MAPE between statistical, machine learning, recurrent, graph-based, and ConvLSTM-family models. ARIMA is not appropriate for dealing with spatial propagation or non-linear context changes. Exogenous variables are necessary for SVR, however fixed feature construction still applies. Temporal relationships are learned by LSTM and GRU, but region structure is not captured. Although GCN and STGCN enhance geographical representation, their fixed graph assumption is less adaptable to short-term route-control modifications. Local propagation is handled by standard ConvLSTM, but residual horizon decoding and explicit context modulation are absent.

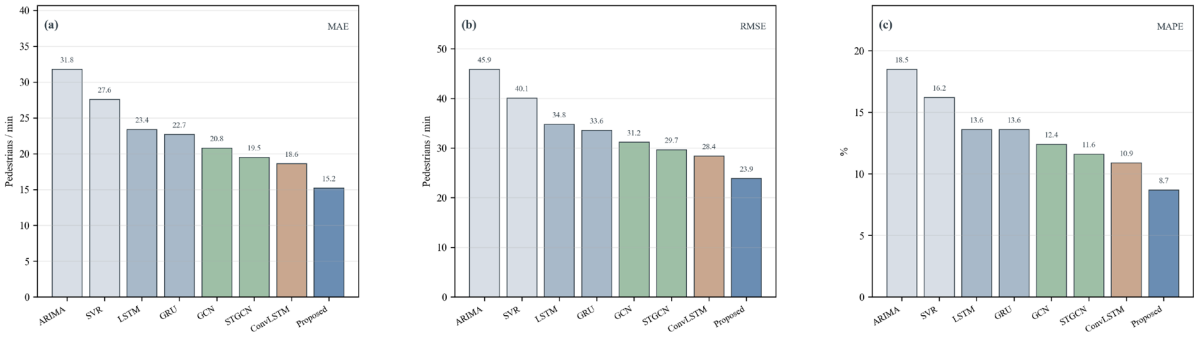

Figure 5. Error comparison of baseline models. (a) MAE; (b) RMSE; (c) MAPE.

Ablation and Error Source Analysis

The module that is in charge of the performance increase is determined by ablation studies. The model as a whole has direction accuracy of 86.7%, RMSE of 23.9, and MAE of 15.2. Without context modulation, the direction accuracy is 82.4%, the MAE is 17.4, and the RMSE is 27.8. The degradation demonstrates how recurrent memory updating is directly impacted by event status, weather, service headway, calendar label, and reliability information. In the absence of the aforementioned signals, the model will observe comparable flow numbers and be unable to discern between an event-driven crowd discharge and a typical commuting wave.

The MAE reached 18.1 and the RMSE was 29.2 without spatial attention; this was the ablation experiment with the biggest rise in RMSE. For this reason, pedestrians' irregular transfer pathways cannot be properly represented by fixed convolutional neighborhoods. For instance, even if two regions are not close in space (that is, not neighboring in the spatial tensor), a platform can have a substantial impact on a faraway departure via a corridor. By allowing non-local but operationally related regions to contribute to the decoded forecast, the adjacency-supported attention branch lowers this inaccuracy.

After removing the residual decoder, the RMSE is 26.4 and the MAE is 16.6. The remaining decoding primarily affects multi-step drift since the error increase is more pronounced at the 90-minute and 120-minute horizons, but it is rather minimal for a short horizon. Direction accuracy decreases to 78.5% when the direction-aware loss is removed, yielding MAE = 16.0 and RMSE = 25.1. As a result, it is evident that an operational forecast close to a turning point requires more than a small-magnitude loss.

Figure 6 displays the ablation data to illustrate how direction loss, residual decoding, spatial attention, and context modulation affect magnitude error and direction accuracy. Direction-aware loss mostly enhances trend accuracy, while context modulation primarily decreases the event-response delay, typically lowers transfer-wave errors spatially, and tackles long-horizon drift residually. These effects can be combined in the construction of the suggested model since they are complementary rather than redundant.

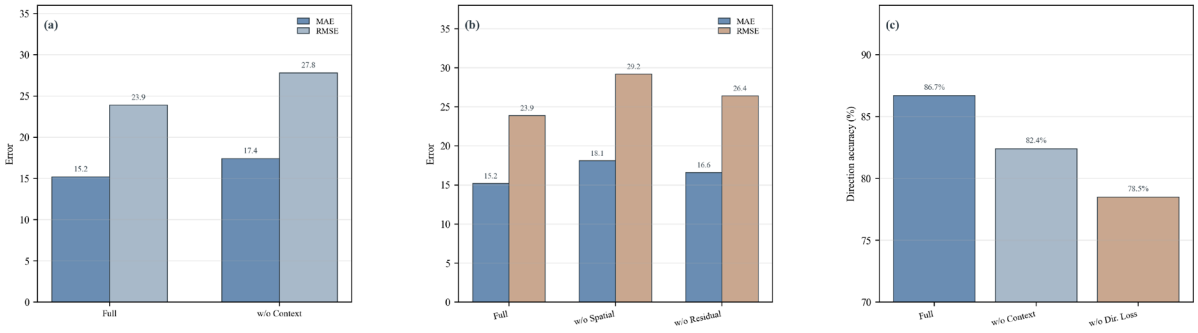

Figure 6. Ablation study results. (a) effect of removing context modulation; (b) effect of removing spatial attention and residual decoding; (c) direction accuracy under objective-function variants.

Horizon Stability and Engineering Efficiency

Longer prediction horizons result in more prediction error, although the suggested model's rate of growth is slower than the comparison spatiotemporal baseline. The suggested procedure yielded MAE values of 10.6, 12.8, 14.5, 15.2, 18.9, and 22.4 at 15, 30, 45, 60, 90, and 120 minutes. STGCN is simultaneously 12.7, 16.4, 20.8, 24.7, 31.6, and 38.5. Because of the comparatively significant gap at 60 minutes, context-modulated memory and residual horizon decoding are better suited for planning-oriented prediction than just urgent short-term guiding.

The Extended-Horizon Advantage is useful. A 90-minute or 120-minute forecast is more appropriate for personnel, queue-control barriers, event preparation, and coordination with external transport providers, however a 15-minute forecast can be used to organize the immediate platform announcement or staff deployment. A model may still seem appropriate for short-term guidance, but it will be unreliable for preparation if the long-term inaccuracy increases too quickly. The suggested model has a 120-minute MAE that is nevertheless useful for early warning, as seen in the example situation.

Efficiency in engineering is also acceptable. For the example benchmark, the model requires 2.5GB of GPU memory and has a forecast window of 10.8 ms per step. It takes 5.8 ms and 0.9 GB for LSTM, 8.4 ms and 1.7 GB for STGCN, and 9.6 ms and 2.2 GB for the conventional ConvLSTM. The latency of the suggested model is significantly shorter than that of the 15-minute sampling interval, despite the fact that it is a little slower because of the addition of context modulation and spatial attention. As a result, there is still sufficient execution time for data import, quality repair, forecast updating, warning production, and operator presentation.

Figure 7 illustrates horizon stability and deployment efficiency, and it links memory usage, inference latency, and prediction degradation to actual ITS operation. As a result, the model can be used to create middle-term resource plans and short-term restrictions. Low dependability ratings will be marked for sensor inspection, while regions with significant anticipated residual increments can be chosen for camera verification. Local actions like opening an auxiliary corridor, altering the direction of escalators, or setting up temporary barriers cannot be supported by the output because it is only at the regional level and not station-total.

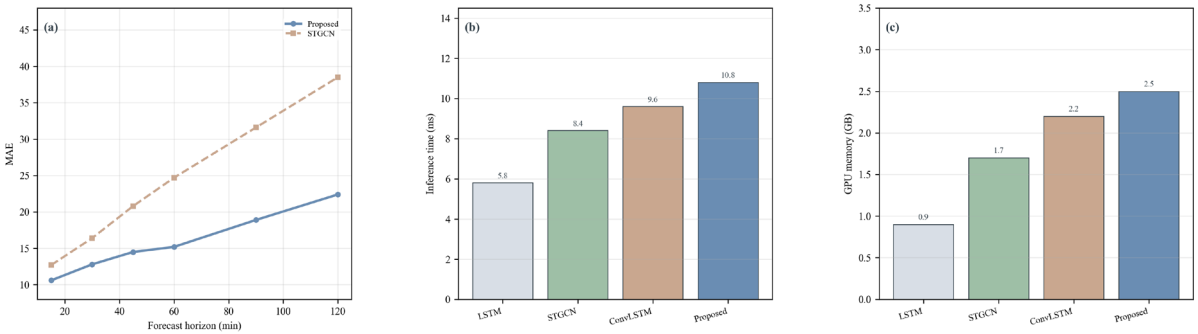

Figure 7. Stability and efficiency analysis. (a) horizon-wise MAE growth; (b) inference time per forecasting window; (c) GPU memory consumption.

Conclusion

A Context-Aware Residual ConvLSTM Framework for Pedestrian Flow Prediction in Intelligent Transportation Systems. Model station observations as spatial tensors, inject event and service context into ConvLSTM gates, use adjacency-supported spatial attention to learn irregular transfer influence, and predict future flow via residual horizon increments. Example experimental results show a lower MAE, RMSE and MAPE than statistical, machine learning, recurrent, graph-based and standard ConvLSTM baselines. Ablation experiments show that context modulation reduces the event-response delay, spatial attention decreases transfer-wave error, residual decoding inhibits long-horizon drift, and direction-aware loss improves trend accuracy.

Future work will expand the framework to dynamic facility graphs, uncertainty intervals, privacy-preserving multi-station training and online adaptation under station layout changes.

Author Contributions

Norbert Paweł Konopka contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Zhang, H., He, J., Bao, J., Hong, Q., & Shi, X. (2020). A Hybrid Spatiotemporal Deep Learning Model for Short-Term Metro Passenger Flow Prediction. Journal of Advanced Transportation, 2020, 1-12. https://doi.org/10.1155/2020/4656435

[2] Ma, J., Zeng, X., Xue, X., & Deng, R. (2022). Metro Emergency Passenger Flow Prediction on Transfer Learning and LSTM Model. Applied Sciences, 12(3), 1644. https://doi.org/10.3390/app12031644

[3] Wang, X., Zhu, C., & Jiang, J. (2023). A deep learning and ensemble learning based architecture for metro passenger flow forecast. IET Intelligent Transport Systems, 17(3), 487-502. https://doi.org/10.1049/itr2.12274

[4] Wang, Z., Zhang, Y., Yao, E., Wang, Y., Li, J., & He, J. (2023). Short-Term Inbound and Outbound Passenger Flow Prediction for New Metro Stations Based on Clustering and Deep Learning. Journal of Advanced Transportation, 2023, 1-12. https://doi.org/10.1155/2023/6659916

[5] Zhang, C., & Berger, C. (2023). Pedestrian Behavior Prediction Using Deep Learning Methods for Urban Scenarios: A Review. IEEE Transactions on Intelligent Transportation Systems, 24(10), 10279-10301. https://doi.org/10.1109/tits.2023.3281393

[6] Korbmacher, R., & Tordeux, A. (2022). Review of Pedestrian Trajectory Prediction Methods: Comparing Deep Learning and Knowledge-Based Approaches. IEEE Transactions on Intelligent Transportation Systems, 23(12), 24126-24144. https://doi.org/10.1109/tits.2022.3205676

[7] Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2020). Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. IEEE Transactions on Intelligent Transportation Systems, 21(11), 4883-4894. https://doi.org/10.1109/tits.2019.2950416

[8] Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. Expert Systems with Applications, 207, 117921. https://doi.org/10.1016/j.eswa.2022.117921

[9] Li, M., & Zhu, Z. (2021). Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(5), 4189-4196. https://doi.org/10.1609/aaai.v35i5.16542

[10] Song, C., Lin, Y., Guo, S., & Wan, H. (2020). Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 34(01), 914-921. https://doi.org/10.1609/aaai.v34i01.5438

[11] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. (2020). Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 753-763. https://doi.org/10.1145/3394486.3403118

[12] Wang, Y., Fang, S., Zhang, C., Xiang, S., & Pan, C. (2022). TVGCN: Time-variant graph convolutional network for traffic forecasting. Neurocomputing, 471, 118-129. https://doi.org/10.1016/j.neucom.2021.11.006

[13] Lee, K., & Rhee, W. (2022). DDP-GCN: Multi-graph convolutional network for spatiotemporal traffic forecasting. Transportation Research Part C: Emerging Technologies, 134, 103466. https://doi.org/10.1016/j.trc.2021.103466

[14] Shin, Y., & Yoon, Y. (2022). Incorporating Dynamicity of Transportation Network With Multi-Weight Traffic Graph Convolutional Network for Traffic Forecasting. IEEE Transactions on Intelligent Transportation Systems, 23(3), 2082-2092. https://doi.org/10.1109/tits.2020.3031331

[15] Xu, Y., Lu, Y., Ji, C., & Zhang, Q. (2023). Adaptive Graph Fusion Convolutional Recurrent Network for Traffic Forecasting. IEEE Internet of Things Journal, 10(13), 11465-11475. https://doi.org/10.1109/jiot.2023.3244182

[16] Weng, W., Fan, J., Wu, H., Hu, Y., Tian, H., Zhu, F., & Wu, J. (2023). A Decomposition Dynamic graph convolutional recurrent network for traffic forecasting. Pattern Recognition, 142, 109670. https://doi.org/10.1016/j.patcog.2023.109670

[17] Shi, Z., Zhang, Y., Wang, J., Qin, J., Liu, X., Yin, H., & Huang, H. (2023). DAGCRN: Graph convolutional recurrent network for traffic forecasting with dynamic adjacency matrix. Expert Systems with Applications, 227, 120259. https://doi.org/10.1016/j.eswa.2023.120259

[18] Wu, J., Fu, J., Ji, H., & Liu, L. (2023). Graph convolutional dynamic recurrent network with attention for traffic forecasting. Applied Intelligence, 53(19), 22002-22016. https://doi.org/10.1007/s10489-023-04621-5

[19] Liu, L., Cao, Y., & Dong, Y. (2023). Attention-Based Multiple Graph Convolutional Recurrent Network for Traffic Forecasting. Sustainability, 15(6), 4697. https://doi.org/10.3390/su15064697

[20] Xia, D., Shen, B., Geng, J., Hu, Y., Li, Y., & Li, H. (2023). Attention-based spatial–temporal adaptive dual-graph convolutional network for traffic flow forecasting. Neural Computing and Applications, 35(23), 17217-17231. https://doi.org/10.1007/s00521-023-08582-1

[21] Chen, Y., Li, K., Yeo, C. K., & Li, K. (2023). Traffic forecasting with graph spatial–temporal position recurrent network. Neural Networks, 162, 340-349. https://doi.org/10.1016/j.neunet.2023.03.009

[22] Zhang, Q., Li, C., Su, F., & Li, Y. (2023). Spatiotemporal Residual Graph Attention Network for Traffic Flow Forecasting. IEEE Internet of Things Journal, 10(13), 11518-11532. https://doi.org/10.1109/jiot.2023.3243122

[23] Bilotta, S., Collini, E., Nesi, P., & Pantaleo, G. (2022). Short-Term Prediction of City Traffic Flow via Convolutional Deep Learning. IEEE Access, 10, 113086-113099. https://doi.org/10.1109/access.2022.3217240

[24] Hou, Y., Deng, Z., & Cui, H. (2021). Short-Term Traffic Flow Prediction with Weather Conditions: Based on Deep Learning Algorithms and Data Fusion. Complexity, 2021(1). https://doi.org/10.1155/2021/6662959

[25] Han, L., & Huang, Y. (2020). Short-term traffic flow prediction of road network based on deep learning. IET Intelligent Transport Systems, 14(6), 495-503. https://doi.org/10.1049/iet-its.2019.0133

[26] Zhang, X., Xu, Y., & Shao, Y. (2022). Forecasting traffic flow with spatial–temporal convolutional graph attention networks. Neural Computing and Applications, 34(18), 15457-15479. https://doi.org/10.1007/s00521-022-07235-z

[27] Sun, S., Wu, H., & Xiang, L. (2020). City-Wide Traffic Flow Forecasting Using a Deep Convolutional Neural Network. Sensors, 20(2), 421. https://doi.org/10.3390/s20020421