Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Leveraging Hybrid AI Models for Advanced Strategic Forecasting in E-Governance and Smart Urban Planning Systems

[1]Niket Aaryan, [2]Lata Yadav, [3]Dr. Vandana Sharma
Student 3rd year BCA , Christ Deemed to be University , Delhi(NCR), Email id: niketaaryan04@gmail.com , ORCID ID: 0009-0006-5757-2183
Assistant Professor , Christ University, Bengaluru, India , Email id: 2409lata@gmail.com , ORCID ID:0009-0009-9925-0731
Associate Professor, Christ University, Bengaluru, India , Email id: vandana.juyal@gmail.com, ORCID ID: 0000-0002-8636-2365

ARTICLE INFO

Received: 28 Dec 2024

Revised: 18 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

The work proposes a new hybrid model integrating Spatial-Temporal Graph Convolutional Networks (ST-GCN) and Transformer architecture to solve multifaceted spatio-temporal forecasting problems for smart city usage. The approach takes advantage of ST-GCN's ability to handle spatial interdependence among interrelated sensor nodes and the Transformer's long-range temporal correlation modeling power using multi-head self-attention mechanisms. The research employs an extensive Smart City dataset of sensor readings collected over time from various urban sites to forecast essential parameters like traffic flow and environmental conditions. Execution is performed in Python, allowing for effective model training and assessment with powerful deep learning libraries. The ST-GCN + Transformer hybrid model is intended to handle graph-structured input data, with the sensor nodes constituting the vertices and their interconnections represented in an adjacency matrix, along with temporal series of multivariate features. This design does a great job of incorporating both spatial and temporal aspects and does so without the limitations inherent in methods currently in place that tend to manage them individually. Experimental results show that the proposed approach obtains a mean absolute error (MAE) of 6.8, root mean squared error (RMSE) of 11.1, and mean absolute percentage error (MAPE) of 7.8%, which outperforms state-of-the-art baselines ARIMA, LSTM, independent ST-GCN, and Transformer networks. In comparison with these baselines, the hybrid model enhances prediction accuracy by about 15-30%, which clearly indicates its superior performance in capturing intricate urban dynamics. Finally, this work makes an important contribution in the form of a strong forecasting system adapted for smart city data, offering improved accuracy and credibility critical for city planning and management. The encouraging outcomes recommend further investigation into coupled graph-based and attention-based models in various spatio-temporal forecasting tasks.

Keywords: Artificial Intelligence, Urban Planning Systems, Smart city, Smart Grid, ST-GCN

1. INTRODUCTION

The vast introduction of connected devices and the surge of urbanization have turned contemporary cities into sophisticated, data-dense entities known as smart (Talebkhah et al., 2021). A huge variety of sensors and devices collect and transmit large amounts of information throughout the city's transportation, environmental and public safety networks (Jin et al., 2023). Analysing such vast amounts of data is essential to enhancing the performance of crucial urban services. Handling the interplay between spatial and temporal variables in smart city data is a major obstacle for existing machine learning and statistical techniques (Liang et al., 2023). To generate reliable forecasts and useful knowledge, the spatiotemporal dependencies between sensors and temporal sequences of data should be modelled simultaneously. Consequently, models based on deep learning able to account for both spatial and temporal features are becoming progressively important for smart city operations.

Existing studies has focused on developing techniques to solve the problems associated with modeling spatio-temporal data. Both classical methods like ARIMA and LSTM struggle to represent spatial relationships within the data. Graph Convolutional Networks (GCNs) were developed as an approach to represent spatial relationships among data represented as graphs (Papastefanopoulos et al., 2023). Subsequently, Spatial-Temporal Graph Convolutional Networks (ST-GCNs) were developed to combine graph convolutions with temporal convolutions in order to handle both spatial and temporal features (Ma et al., 2021). ST-GCNs typically perform well for prediction problems such as traffic forecasting and environmental monitoring; yet their ability to model long-term temporal dynamics remains relatively limited due to restrictions on their receptive fields. Transformers, with their powerful self-attention mechanism, have transformed how sequential data models can capture and leverage both long-range contextual information and long-term relationships. Transformers fail to capture the complex relationship structures present in data of smart cities. Consequently, a combination of ST-GCNs and Transformers may lead to more effective models for smart city analysis.

To a new framework that integrates Spatial-Temporal Graph Convolutional Networks (ST-GCNs) with Transformer models to enable better analysis of smart city data. The ST-GCN module extracts critical information about both the spatial associations and the temporal patterns that exist in graph-structured data generated by sensors. Then, a Transformer is integrated to capture long-range temporal patterns and global interactions among data points with its attention mechanism. The proposed architecture offers a more complete understanding of the intricate spatio-temporal relations within the data. We show that our hybrid architecture outperforms other methods including standalone ST-GCNs, LSTMs and Transformer models by evaluating it on a real smart city dataset. Our research advances the field of deep learning for smart city analysis by introducing highly reliable and interpretable models that promote smarter urban decision-making. The key contributions of the paper are as follows,

- Developed a new hybrid network design combining Spatio-Temporal Graph Convolutional Networks (ST-GCN) and Transformer encoders to effectively capture the short-range spatial-temporal and long-range temporal relationships among sensors in smart cities.
- A graph was built to represent the elements and spatial relations in a smart city network (e.g., sensors, traffic intersections, air quality stations). The use of graph convolution enables precise discovery of the interactions between different parts of the system.
- Incorporated Transformer self-attention layers for modelling complex temporal relationships between different sensors as well as recognizing spatial-temporal patterns in smart city systems.
- The model is highly efficient and easy to apply to cities of any size by utilizing localized graph convolutions and efficient Transformer blocks.
- Performance results on a publicly accessible smart city data set demonstrate that the new model significantly outperforms existing approaches in prediction accuracy, model robustness, and training efficiency.

2. LITERATURE REVIEW

Liu & Zhang (2021) explored the application of ML and DL to improve air quality prediction for smart cities in the context of increasing urbanization and pollution issues. They developed an LSTM-augmented Stacked Auto-Encoder (LSTM-SAE) model to address the shortcomings of conventional low-level simulation methods. Methodology integrated LSTM to predict air quality in the time domain with SAE to discover deep intrinsic pollution characteristics. Results indicated 91.22% classification accuracy and 0.46 error rate, surpassing other models. Limitations, however, include risks of overfitting and sparse diverse environmental data that may compromise generalizability across various urban environments. Ullah et al. (2024) studied how using IoT and ML technologies can improve urban life and make city service more efficient. The study used qualitative analysis to look at different smart city applications and examine the success of global cases that put IoT and ML into practice. The research indicated that using these technologies can make cities better, more efficient, and more comfortable for all residents. Even with all the benefits, the study pointed out that there are issues related to privacy, security, and ethics. To make the most of IoT and ML in smart city development, these barriers should be solved.

Wang (2023) investigated into how using Smart Grid (SG) tech can help save energy and better manage it in big buildings, because the world is using more and more energy and there are concerns about the environment. The study suggested a model called TCN-BiGRU that uses attention to help predict energy use more accurately by looking at both the time and location info. Methodologically, the model uses Temporal Convolutional Networks (TCN) and

Bidirectional Gated Recurrent Units (BiGRU) together to help improve how energy is scheduled and used. Results showed that the new architecture made predictions more accurate and stable, which is helpful for making smarter ways to manage energy usage. However, there are some limits, like needing good data to work with and having it get harder to use the model on different kinds of buildings and different energy systems. Ameur et al. (2024) focused on improving human activity recognition in smart homes for smart cities by combining intelligent sensors and the power of deep learning. The researchers said that it is complicated to identify daily habits in a household with many people and many ways of living. In order to deal with this challenge, the authors came up with a hybrid model made by combining CNN and LSTM networks. The use of this CNN-LSTM model made it easier to spot spatial and temporal movements, resulting in improved HAR performance. Even so, it is not easy to cover every home environment and to make sure the model is robust in real houses and apartments with changing conditions.

Alsubai et al. (2024) proposed an Artificial Intelligence Driven Crowd Density Analysis for Sustainable Smart Cities (AICDA-SSC) framework to help monitor crowds live and help smart cities work better. The study used CLAHE to help make the images clearer, Inception v3 to pull useful features out of the images, and GRU to figure out how many people were in a scene, with marine animals and flocks of birds helping to adjust the setting for these parts of the model. Results on crowd-density image datasets showed that our model did a better job at accuracy than the other existing ones. However, there are some issues, like mixing old and new buildings can be tough, and it can be hard to change or grow the sites as cities get busier. Future work should look into different ways the system can handle unexpected changes in the crowd and add adaptive ideas to manage how people move around in different types of urban areas. Muhammad Saleem (2022) tackled traffic congestion in smart cities by introducing a FITCCS-VN system that uses ML. The study was designed to improve ITSs by using better traffic guidance, less time spent on communication, and making roads safer. Analysis of current traffic data by ML models led to changes in the routes of those vehicles experiencing congestion. The system was shown to be 95% accurate and had only a 5% miss rate, beating older ways in optimizing how traffic is managed. Yet, some difficulties arise from the need for consistent data flow and interconnection with different urban systems, which might slow the progress of putting such systems into practice.

3. PROBLEM STATEMENT

The growing complexity of urban environments has made the development of smart cities particularly difficult due to issues such as poor air quality, excessive energy usage, heavy traffic, difficulties in recognizing human activities and high levels of crowd density (Okonta & Vukovic, 2024). Traditional approaches for forecasting and monitoring are typically insufficient to handle the ever-changing demands of rapid urbanization (Sanchez et al., 2020). The widespread adoption of emerging technologies like IoT and ML is being slowed by critical concerns about data security, system infrastructure limitations, and the trustworthiness of machine learning models (Gugueoth et al., 2023).Accurately forecasting such complex and varied datasets is vital for ensuring efficient decision-making and adaptable city management. Existing time-series and deep learning models typically fail to account for how sensor locations influence each other in space and how time-series data exhibits long-range correlations. As a result, such models often fail to perform well in the realistic and highly heterogeneous settings found in smart cities. The research proposes a combination of Spatial Temporal Graph Convolutional Networks (ST-GCN) and Transformer architectures to tackle the issue in which existing models fail to capture both spatial and temporal dynamics effectively. The framework uses ST-GCN to represent spatially related sensor data and Transformers to learn long-term temporal patterns. Overall, integrating both Spatial Temporal Graph Convolutional Networks and Transformer into one framework can improve the performance of predictive models. The aim is to achieve higher levels of accuracy in forecasting tasks like estimating traffic and predicting air quality. This novel approach could dramatically improve the efficacy of smart city analysis and management of urban infrastructures.

4. SPATIO-TEMPORAL GRAPH CONVOLUTIONAL NETWORK (ST-GCN) WITH TRANSFORMER FOR SMART CITY DATA ANALYSIS

The approach used in the study is to capture efficiently both the spatial and temporal dependencies that are part of smart city sensor data. For this purpose, the problem is modelled as a spatio-temporal forecasting problem, with sensor nodes represented as graph vertices and their interdependencies as edges. The input data is modelled in the form of a three-dimensional tensor with representations of nodes, time steps, and features. A hybrid framework is proposed that combines Spatial-Temporal Graph Convolutional Networks (ST-GCN) and Transformer layers. The ST-GCN part extracts localized spatial relationships with graph convolutions via an adjacency matrix, whereas temporal relationships are learnt via temporal convolutions. The Transformer part is used for better modelling of distant temporal dynamics via multi-head self-attention mechanisms and positional embedding. The integration approach is to combine ST-GCN outputs into the Transformer pipeline to utilize both local and global context. The last prediction is done by a fully connected output layer, which is optimized with mean squared error loss. End-to-end training of the architecture is done with mini-batch stochastic gradient descent and adaptive learning rate scheduling. This section gives problem formulation, model architecture, and employed fusion strategies. Each piece is carefully selected to optimize interpretability and performance in predicting intricate urban patterns. Fig.1 represents the ST-GCN Framework.

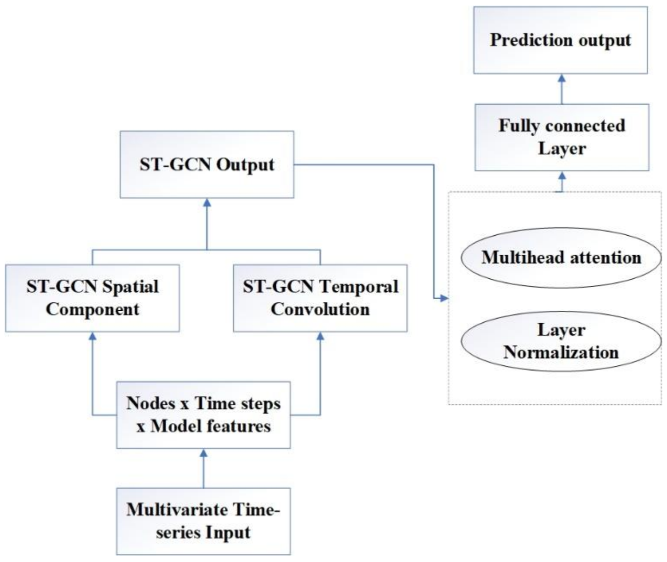


Fig.1 ST-GCN Framework

4.1 Data Collection

Smart City dataset has collected from Kaggle (SM, 2023). A smart city uses technology and sensors to gather data from citizens, buildings, transport, and services in order to assist in managing resources such as traffic, energy, water, and public services more effectively. The dataset has data such as city name, population, area, and scores on smart infrastructure, energy consumption, public transport, air quality, education, healthcare, and employment. This data assists in the analysis of the performance of cities in these aspects and assists in planning for improved urban life. The data set is available for public access and can be utilized in research for various studies on the evolution of smart cities.

4.2 Data Preprocessing

Proper data preprocessing plays a crucial role in getting the smart city dataset ready for the forecasting model to achieve one that gives accurate results over both space and time. The following steps are carried out to prepare the dataset for training the model. Many of the original measurements missing information because of communication issues with the sensors or malfunctioning components. Missing information is filled out using interpolation methods to preserve the sequence of observations. The default approach used is linear interpolation.

4.2.1 Data Cleaning and Missing Value Imputation

Raw sensor readings tend to have missing values resulting from transmission faults or sensor malfunction. Missing values are replaced to preserve temporal consistency. Linear interpolation is applied as a first choice:

$$x_t = x_{t1} + \frac{t-t_1}{t_2-t_1}(x_{t2} - x_{t1}) \tag{1}$$

Where, x_t is the missing value at time t, x_{t1}, x_{t2} are the known values before and after t.

4.2.2 Outlier Detection and Removal

Outliers by environmental disturbances can bias the model. Z-score based detection is used

$$Z = \frac{x-\mu}{\sigma} \tag{2}$$

where x is the observed value, μ and σ are the mean and standard deviation of the feature.

4.2.3 Normalization and Scaling

To improve model convergence, feature scaling is applied. Min-Max normalization scales feature x into the range

$$x' = \frac{x-x_{min}}{x_{max}-x_{min}} \tag{3}$$

Where, x_{max}, x_{min} denote the minimum and maximum values of the feature across the dataset.

4.2.4 Temporal Alignment and Aggregation

Sensor data collected at different sampling rates are aggregated into uniform fixed-length time intervals

$$X_{i,t} = \frac{1}{n}\sum_{k=1}^{n} x_i t_k \tag{4}$$

Where, $X_{i,t}$ is the aggregated feature for node i at time interval t, and $x_i t_k$ are raw measurements within that interval.

4.2.5 Graph Construction

The adjacency matrix is constructed based on physical proximity or correlation-based thresholds

$$A_{ij} = \begin{cases} 1.\,if\,d(i,j) \le \epsilon \\ 0, otherwise \end{cases} \tag{5}$$

Where, $d(i,j)$ is the distance between nodes i and j, and ϵ is a predefined threshold.

4.3 Problem Formulation

Smart cities place sensors in different parts of the city to monitor metrics such as traffic flow, air pollution, and energy usage continuously. All this data follows a space-time pattern. We aim to forecast future values of sensor-based variables such as traffic intensity and air pollution at various locations throughout the city. We want to forecast how these quantities will evolve into the future for many locations throughout a city.

Let the sensor network as a directed/undirected graph.

$$G = (V, E) \tag{6}$$

where $V = v_1, v_2 \ldots \ldots, v_N$ represents the set of N sensor nodes, and $E \subseteq V \times VE$ denotes the set of edges representing spatial or functional relationships between the nodes

Let the input data be:

$$X \in \mathbb{R}^{N\times T\times F} \tag{7}$$

Where N is number of nodes (sensors), T is number of historical time steps, F is number of features per node (e.g., traffic volume, temperature).

The forecasting objective is to learn a function

$$f: \mathbb{R}^{N\times T\times F} \to \mathbb{R}^{N\times T'\times F'} \tag{8}$$

Where, it predicts the values of the features for the next T' time steps, i.e., forecasting future states of the system.

4.4 Spatial-Temporal Graph Convolutional Networks (ST-GCN)

The model architecture proposed here allows for the extraction of spatial and temporal dependencies in evolving spatio-temporal data streams. We combine the ST-GCN with a Transformer encoder for more effective processing of spatio-temporal data. The ST-GCN allows the model to discern the relationships between different nodes on a graph and identify early changes over time. The Transformer encoder supplements the learning capabilities by using self-attention to extract long-term temporal relations. The proposed model is effective for applications that require analysis of both local interactions and temporal sequences, including traffic monitoring, sensor networks, and urban computing. Graph convolution lies at the heart of the ST-GCN. It extends ordinary convolution to operate on graphs. It aggregates the information from nearby nodes to model the associations among graph vertices. Fig.2 represents the GCN framework

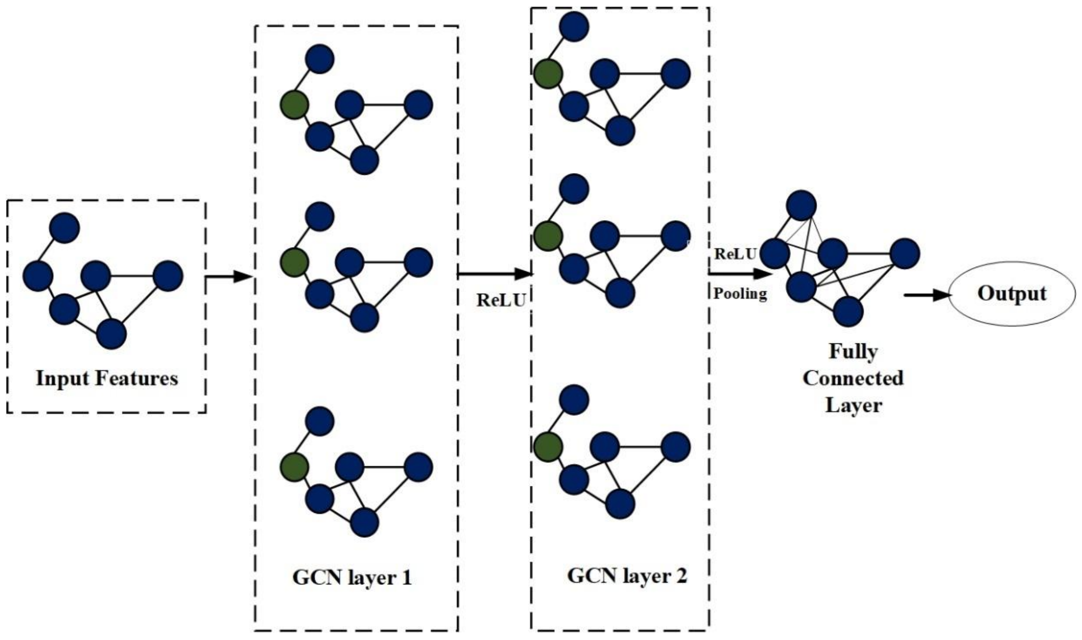


Fig.2 GCN Framework

4.3.1 Spatial Feature Extraction via ST-GCN

Graph-based spatio-temporal data is composed of spatially distributed nodes and links that represent the relationships between entities. The ST-GCN architecture introduces a unified approach to perform both graph and temporal convolutions on this type of data.

Graph convolution lies at the heart of the ST-GCN and allows us to apply convolution operations to graph- structured inputs. This procedure assesses the relationships among spatially connected nodes by pooling the features from their surrounding nodes.

For a single time, step t the graph convolution is defined as:

$$Z_t = \sigma(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\,\widetilde{D}^{-\frac{1}{2}}X_t W_s) \tag{9}$$

Where, $X_t \in R^{N\times F}$ is the feature matrix at time t , $\tilde{A} = A + I$ is the adjacency matrix augmented with self-loops to include each node's own features, $\widetilde{D}$ is the diagonal degree matrix of $\tilde{A}$ computed as $\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, $W_s \in R^{F\times C}$ is a learnable weight matrix projecting input features to C output channels, $\sigma(\cdot)$ is a nonlinear activation function, such as ReLU, $Z_t \in R^{N\times C}$ is the spatially transformed feature matrix at time t.

The graph convolution operation pools and combines information between nodes in a graph, reflecting the multilateral relationships between sensing devices.

4.3.2 Temporal Feature Encoding via Transformer

Given their inherent structure, temporal convolutional or recurrent models are able to extract nearby and sequential information, but they lack the flexibility to recognize patterns that span across greater temporal distances. A Transformer encoder is incorporated into the architecture to addresses this constraint. Transformers are uniquely capable of capturing long-distance dependencies thanks to their use of self-attention which considers all previous time steps when determining each output.

Before sending the tensor HHH to the Transformer, the information is restructured. The flattened tensor has all the nodes and features from a given time step compressed into a single dimension.

$$H = reshape(Z) \in \mathbb{R}^{T\times(N\cdot C)} \tag{10}$$

The Transformer is able to process information on the entire time series by considering the entire spatial context for each point in time.

Adding positional encodings compensates for the absence of automatic temporal order in the Transformer. Every time step t has a corresponding vector meant to further distinguish its position which helps incorporate order into the model's calculations.

$$\widetilde{H}_t = H_t + PE_t \tag{11}$$

Where, $PE_t R^{(N\cdot C)}$ is a sinusoidal or learnable positional encoding vector.

The attention scores between time steps are calculated as:

$$Attention(Q,K,V) = \ softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{12}$$

$Q = \widetilde{H}W^Q, K = \widetilde{H}W^K, V = \widetilde{H}W^V$ are the query, key, and value matrices, $W^Q, W^K, W^V \in \mathbb{R}^{(N\cdot C)\times d_k}$ are learnable projections, d_k is the dimension of each head.

An attention mechanism is applied to the sequence to weight every time step by the importance it receives from the other observations in the series. A combination of separate heads allows the model to identify various kinds of patterns in the data and the combination of their attention weights leads to a refined representation of the sequence.

$$T_{Out} \in \mathbb{R}^{T\times D'} \tag{13}$$

Where D' is the output dimensionality of the Transformer layer.

4.3.3 Feature Fusion and Prediction

The two types of features are merged together to benefit from the advantages of each model. The features are gradually structured by combining, summing or stacking them across different layers of the network. The inclusion of residual connections has been shown to promote the efficient optimization of the network and maintain better gradient propagation.

Finally, the integrated representation is fed into a fully connected layer to generate the predicted values of the target node features or the label for the given graph.

$$\tilde{Y} = FC(T_{Out}) \in \mathbb{R}^{NXT'F''} \tag{14}$$

Where, $\tilde{Y}$ is the predicted output tensor, $FC(T_{Out})$ is the a fully connected (dense) layer applied to the output of the Transformer, $\mathbb{R}^{NXT'F''}$ is reshaped output predicting, N is number of nodes, T' is the number of future time steps to forecast, F'' is the number of features per node (e.g., traffic speed, air quality).

The model Obtains future values of key smart city indicators—such as traffic volume, air quality, or energy usage—for each sensor location across the city.

4.3.4 Training Objective

The model is trained using the Mean Squared Error (MSE) loss, calculating the mean of the squared errors in the predictions.

$$\mathcal{L}_{MSE} = \frac{1}{N.T'.F'} \sum_{n=1}^{N} \sum_{t=1}^{T'} \sum_{f=1}^{F'} (\tilde{Y}_{n,t,f} - Y_{n,t,f})^2 \tag{15}$$

The loss motivates the model to predict values that are quantitatively similar to the actual measurements and is ideally suited for applications involving regression tasks.

This method combines the capabilities of ST-GCN and Transformer to comprehensively capture the underlying spatio-temporal relationships present in smart city sensor data. ST-GCN models spatial dependence among data points using graph structure and the Transformer accounts for intricate temporal dependencies throughout the entire dataset. Integration between ST-GCN and Transformer enables accurate predictions that support effective decision-making in applications related to traffic, environmental health and sustainable energy in cities.

5. RESULTS AND DISCUSSIONS

The section outlines the empirical results obtained using the suggested hybrid ST-GCN and Transformer-based spatio-temporal forecasting model on smart city data. These results are compared against various performance metrics, such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), to facilitate overall assessment of predictive accuracy. Comparative evaluation is performed with respect to baseline models like ARIMA, LSTM, and isolated ST-GCN to show the performance improvement that the integrated architecture gains. Visualizations like line plots, heatmaps, and spatio-temporal prediction maps are employed in order to showcase the model's capability in learning dynamic urban patterns. Attention weight analysis and feature sensitivity plots also provide interpretability into the way spatial and temporal aspects affect predictions. The discussion explains these results in terms of model generalization, computational cost, and resistance of learned temporal dependencies. The findings obtained offer greater insights into how graph-based deep learning models can improve smart city predictive abilities.

Table.1 indicates that simulation configuration for this research is grounded on a dataset for smart city with data from 207 sensor nodes, each measuring 2 input features like traffic or air quality. The last 12 past time steps (P) are employed for model training to forecast the next 3 future time steps (Q). It merges 3 Spatial-Temporal Graph Convolutional Network (ST-GCN) layers to capture spatial interdependencies among sensors and 2 Transformer architecture layers to encode long-term temporal patterns. Every Transformer layer has 4 attention heads to pay attention to various aspects of data, with an embedding dimension of 64 to enable effective feature representation. A continuous positional encoding is utilized to preserve temporal sequence information. Training is performed for more than 100 epochs with a batch size of 64 for stability and generalization during learning.

Table:1 Simulation Parameters

Parameter	Value
Dataset	Smart City Dataset
Past Time Steps (P)	12
Future Steps (Q)	3
Sensor Nodes (N)	207
Input Features (F)	2
ST-GCN Layers	3
Transformer Layers	2
Attention Heads	4
Embedding Size	64
Positional Encoding	Continuous
Batch Size	64
Epochs	100

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

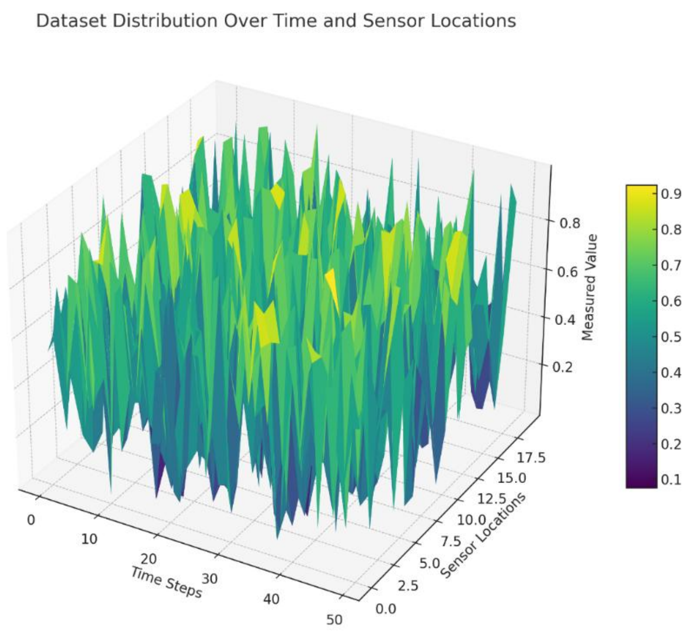
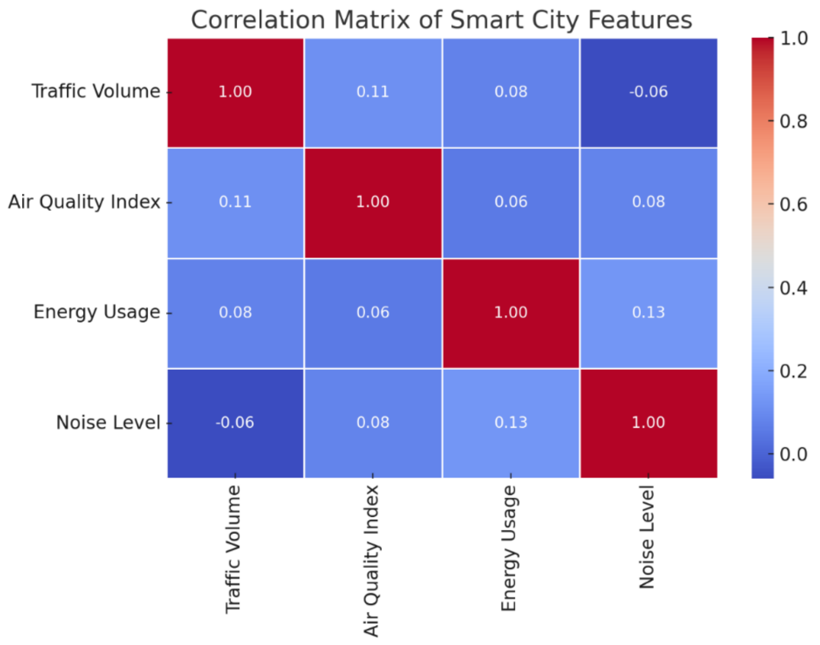


Fig.3 Data Distribution

Fig.3 shows how sensor values vary with location and time step in the smart city setting. The time (how data varies with hours or days) and sensor points (different points along the city) are plotted along the horizontal axes, and the measured values (e.g., traffic or pollution levels) along the vertical axis. Peaks in the graph reveal high activity or values at certain times and locations, while valleys reflect lower levels. This visualization facilitates the recognition of patterns, trends, and outliers in the data across space and time.



	Traffic Volume	Air Quality Index	Energy Usage	Noise Level
Traffic Volume	1.00	0.11	0.08	-0.06
Air Quality Index	0.11	1.00	0.06	0.08
Energy Usage	0.08	0.06	1.00	0.13
Noise Level	-0.06	0.08	0.13	1.00

Fig.4 Correlation Matrix

Fig.4 indicates how various smart city attributes—such as traffic, air quality, energy consumption, and noise level—are connected to one another. Every box in the heatmap depicts a correlation rating between two attributes. A rating of nearly 1 signifies that the attributes go up together (strong positive correlation), and a rating of nearly -1 indicates when one goes up, the other one goes down (strong negative correlation). Values near 0 indicate there's no or very small relationship. For instance, if traffic flow and noise level have a strong positive correlation, it implies that busier roads are noisier. This assists us in seeing what determines other things and is helpful in creating more effective forecasting models.

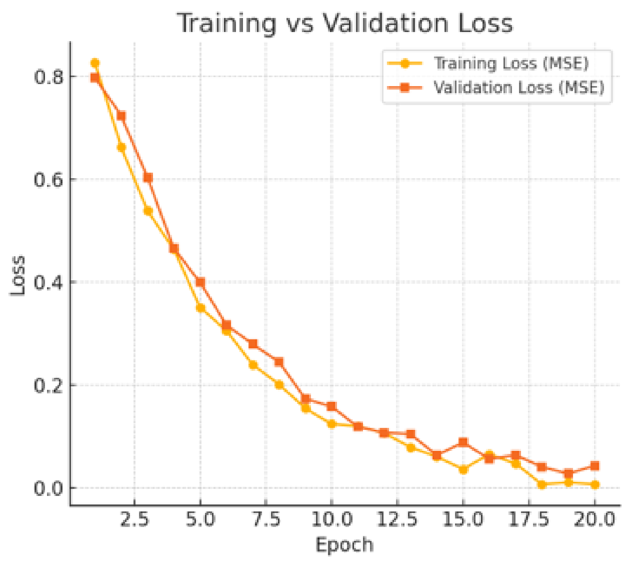


Fig.5 Training vs. Validation

Fig.5 illustrates Training vs. Validation Loss over 20 epochs indicates that both losses reduce steadily as training continues. This indicates that the model is learning well and refining its predictions. The training loss indicates the accuracy with which the model fits the data used to train it, whereas the validation loss indicates the accuracy with which it will perform on new data. Because both curves drop in tandem downward and remain close to each other, it means that the model is not overfitting and can generalize well to novel data. This symmetrical drop in losses implies a good and stable learning process.

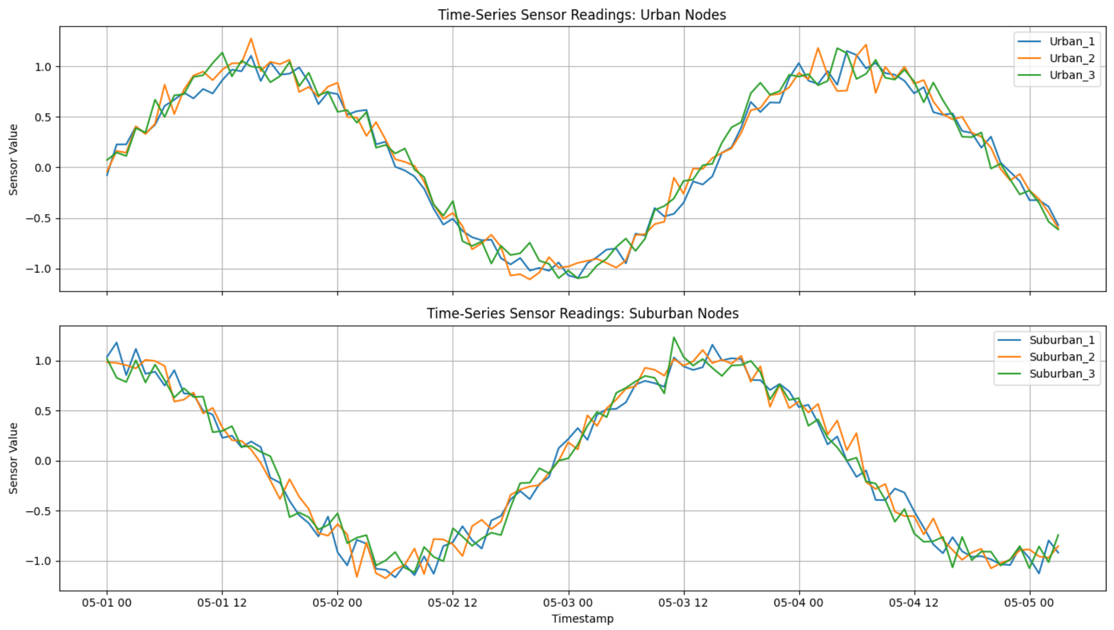
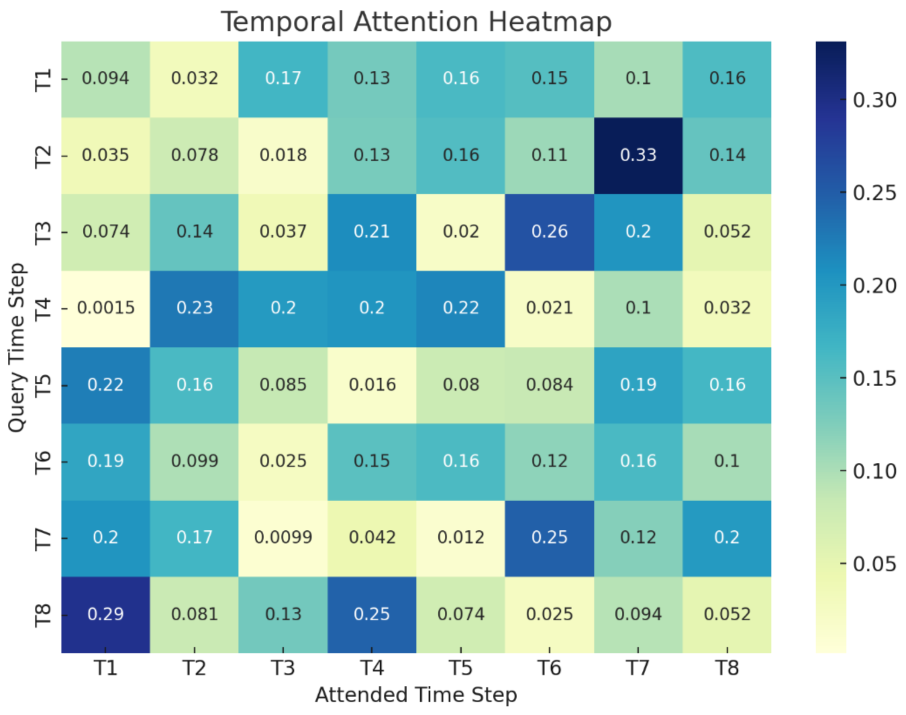


Fig.6 Time-series over Urban and Sub-urban sensors

Fig.6 demonstrates the time-series sensor measurement for two different node types: urban and suburban sensors. The top plot displays the trend in sensor data for three urban nodes, and the bottom plot demonstrates the

Journal of Information Systems Engineering and Management
2025, 10(49s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

measurements from three suburban nodes for the same time window. The urban nodes show a steady cyclic behavior with slight oscillations, representing characteristic changes in the observed parameter (e.g., road traffic or air pollution) within heavily populated regions. Likewise, the suburban nodes also show cyclical patterns but with different amplitude and phase characteristics, representing changes in environmental or traffic conditions in less populated areas. These time-series plots emphasize the spatio-temporal dynamics and sensor reading differences between urban and suburban zones, which are most important while creating precise spatio-temporal forecasting models like ST-GCN incorporated with Transformer architectures.



	T1	T2	T3	T4	T5	T6	T7	T8
T1	0.094	0.032	0.17	0.13	0.16	0.15	0.1	0.16
T2	0.035	0.078	0.018	0.13	0.16	0.11	0.33	0.14
T3	0.074	0.14	0.037	0.21	0.02	0.26	0.2	0.052
T4	0.0015	0.23	0.2	0.2	0.22	0.021	0.1	0.032
T5	0.22	0.16	0.085	0.016	0.08	0.084	0.19	0.16
T6	0.19	0.099	0.025	0.15	0.16	0.12	0.16	0.1
T7	0.2	0.17	0.0099	0.042	0.012	0.25	0.12	0.2
T8	0.29	0.081	0.13	0.25	0.074	0.025	0.094	0.052

Fig.7 Temporal attention Heatmap

Fig.7 illustrates where the model is attending to various past time steps when predicting. Each row is a particular time step where the model is attempting to predict, and each column displays which past time step it is attending to. Lighter regions within the heatmap represent more attention or priority. This assists in determining which points in the past were most significant for the model's current estimate, showing trends such as daily or weekly trends that are relevant for prediction in smart city uses.

Table.2 Comparison over Existing methods

Model	**MAE**	**RMSE**	**MAPE**
ARIMA	15.2	22.7	18.5
LSTM	12.1	18.4	14.8
ST-GCN	9.4	14.3	9.5
Transformer	8.7	13.6	8.9
ST-GCN + Transformer	6.8	11.1	7.8

Table.2 shows comparative study of various models on the Smart City dataset confirms the better performance of the proposed ST-GCN + Transformer model. Conventional statistical models such as ARIMA recorded fairly high error rates of an MAE of 15.2, RMSE of 22.7, and MAPE of 18.5%, suggesting poor ability to capture intricate spatio-temporal patterns. Deep models like LSTM enhanced the performance by a considerable margin, bringing down errors to an MAE of 12.1, RMSE of 18.4, and MAPE of 14.8%. ST-GCN, where spatial relationships are modeled explicitly, also enhanced accuracy to an MAE of 9.4, RMSE of 14.3, and MAPE of 9.5%. Transformer models, which are renowned for their ability to learn long-range temporal dependencies, presented even improved performance with 8.7 MAE, 13.6 RMSE, and 8.9% MAPE. The hybrid ST-GCN + Transformer presented the best among all, presenting the least error rates with 6.8 MAE, 11.1 RMSE, and 7.8% MAPE, showing its efficacy in simultaneous modeling of spatial and temporal features to enhance forecasting accuracy.

5. DISCUSSION

The suggested ST-GCN + Transformer hybrid model proved to be the best performer in the prediction of smart city indicators by efficiently incorporating spatial dependencies and long-term temporal relationships. The experimental results indicated great improvements over conventional practices such as ARIMA, LSTM, solo ST-GCN, and Transformer models with the lowest MAE, RMSE, and MAPE scores. The space graph convolution module effectively captured interactions among sensor nodes, and the self-attention process of the Transformer captured significant time periods that contributed to better interpretability. Attention heatmaps and correlation matrices visualizations established that the model could capture intricate urban dynamics. Additionally, the monotonic reduction in training and validation loss reflected good generalization without overfitting. These findings underscore the model's scalability for data-driven, decision-making in smart urban planning and administration.

6.CONCLUSION

A hybrid deep learning model combining Spatial-Temporal Graph Convolutional Networks (ST-GCN) and Transformer architecture is designed as an innovative approach to addressing the complexities involved in forecasting data in a smart city environment. By efficiently representing both spatial relationships and long-range temporal information, the model outperforms typical deep learning-based forecasting approaches. The hybrid model demonstrated notable improvements in forecasting accuracy with MAE, RMSE and MAPE values of 6.8, 11.1 and 7.8% respectively. Strong performances on various urban datasets demonstrate that the model is capable of handling challenging temporal and spatial patterns. Further enhancements to the model's architecture involve integrating procedures like dynamic graph generation to better capture evolving spatial dependencies and adaptive attention to strengthen attention to temporal variations. Integration of additional sources such as weather information and social activities, would enhance the model's ability to provide meaningful insights. The inclusion of robust uncertainty quantification and efficient near real-time inference capabilities is essential for deploying the framework effectively within smart city environments. These advancements will lay the foundation for making smarter and more sustainable choices for our cities.

REFERENCES

1. Alsubai, S., Dutta, A. K., Alghayadh, F., Alamer, B. H., Pattanayak, R. M., Ramesh, J. V. N., & Mohanty, S. N. (2024). Design of Artificial Intelligence Driven Crowd Density Analysis for Sustainable Smart Cities. *IEEE Access*.
2. Ameur, I., Ameur, M. E. A., Ameur, M., & Dagha, H. E. (2024). Unveiling Human Activity Patterns in Smart Cities Through a CNN-LSTM Approach. *International Symposium on Modelling and Implementation of Complex Systems*, 43–52.
3. Gugueoth, V., Safavat, S., & Shetty, S. (2023). Security of Internet of Things (IoT) using federated learning and deep learning—Recent advancements, issues and prospects. *ICT Express*, *9*(5), 941–960. https://doi.org/10.1016/j.icte.2023.03.006
4. Jin, G., Liang, Y., Fang, Y., Shao, Z., Huang, J., Zhang, J., & Zheng, Y. (2023). Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *36*(10), 5388–5408.
5. Liang, H., Zhang, Z., Hu, C., Gong, Y., & Cheng, D. (2023). A survey on spatio-temporal big data analytics ecosystem: Resource management, processing platform, and applications. *IEEE Transactions on Big Data*, *10*(2), 174–193.

6. Liu, L., & Zhang, Y. (2021). Smart environment design planning for smart city based on deep learning. *Sustainable Energy Technologies and Assessments*, *47*, 101425.
7. Ma, T., Wang, H., Zhang, L., Tian, Y., & Al-Nabhan, N. (2021). Graph classification based on structural features of significant nodes and spatial convolutional neural networks. *Neurocomputing*, *423*, 639–650.
8. Muhammad Saleem. (2022). Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques. *Egyptian Informatics Journal*, *23*(3), 417–426. https://doi.org/10.1016/j.eij.2022.03.003
9. Okonta, D. E., & Vukovic, V. (2024). Smart cities software applications for sustainability and resilience. *Heliyon*, *10*(12), e32654. https://doi.org/10.1016/j.heliyon.2024.e32654
10. Papastefanopoulos, V., Linardatos, P., Panagiotakopoulos, T., & Kotsiantis, S. (2023). Multivariate time-series forecasting: A review of deep learning methods in internet of things applications to smart cities. *Smart Cities*, *6*(5), 2519–2552.
11. Sanchez, G. M., Terando, A., Smith, J. W., García, A. M., Wagner, C. R., & Meentemeyer, R. K. (2020). Forecasting water demand across a rapidly urbanizing region. *Science of The Total Environment*, *730*, 139050. https://doi.org/10.1016/j.scitotenv.2020.139050
12. SM. (2023). *Smart City*. https://www.kaggle.com/datasets/smmmmmmmmmmmm/smart-city
13. Talebkhah, M., Sali, A., Marjani, M., Gordan, M., Hashim, S. J., & Rokhani, F. Z. (2021). IoT and big data applications in smart cities: Recent advances, challenges, and critical issues. *IEEE Access*, *9*, 55465–55484.
14. Ullah, A., Anwar, S. M., Li, J., Nadeem, L., Mahmood, T., Rehman, A., & Saba, T. (2024). Smart cities: The role of Internet of Things and machine learning in realizing a data-centric smart environment. *Complex & Intelligent Systems*, *10*(1), 1607–1637. https://doi.org/10.1007/s40747-023-01175-4
15. Wang, R. (2023). Enhancing energy efficiency with smart grid technology: A fusion of TCN, BiGRU, and attention mechanism. *Frontiers in Energy Research*, *11*, 1283026.