A Novel Approach for Urban Traffic Congestion Prediction

Chaimae Kanzouai[1], Abderrahim Zannou[2],
Soukaina BOUAROUROU[3], El Habib Nfaoui[4], Abdelhak Boulaalam[5]
LSATE Laboratory, Sidi Mohamed Ben Abdellah University, Fez, Morocco[1,5]
ERCI2A, Faculty of Sciences and Techniques, Abdelmalek Essaadi University, Tetouan, Morocco[2]
Computer Science Department-Faculty of Sciences, Mohamed V University in Rabat, Rabat, Morocco[3]
L3IA Laboratory, Sidi Mohamed Ben Abdellah University, Fez, Morocco[4]

***Abstract*—Traffic congestion is a global problem in urban areas that creates longer travel times, increased fuel consumption, and elevated levels of pollution. Traffic congestion occurs because of the exponential growth of vehicles along with a finite number of roadways and the inability to manage traffic effectively. This paper studies the question: How well can traffic type factors be used as a predictor for determining the severity of traffic congestion? To answer this question, we present a new methodology to perform clustering and classification based on various types of traffic indicators. In addition, traffic indicators (such as size of roadway, speed of vehicles, number of vehicles, and level of traffic flow) are categorized by using two distinct classifications: homogeneous and heterogeneous. Using these categories, we then apply a modified version of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to do clustering of traffic indicators. The resultant label from the clustering process is then used to develop a prediction model that will provide information regarding the level of traffic congestion along a selected roadway. Results from our experiments were conducted using an actual dataset and demonstrate that our proposed method produced an accuracy rate of 93% with 92% precision and recall, and therefore, outperforming other current methodologies used for predicting traffic congestion. Overall, these findings indicate that incorporating an analysis of traffic type factors into the clustering and classification methodology can result in more accurate predictions of traffic congestion.**

Keywords*—*Traffic congestion; traffic management; traffic factors; congestion level; DBSCAN; GCN

I. Introduction

Urban traffic congestion [1] is a significant challenge in modern cities, impacting daily commutes, fuel consumption, and environmental sustainability. Traffic congestion arises due to an imbalance between the increasing number of vehicles and the capacity of road infrastructure [2]. External factors such as weather conditions [3], peak hours, and road incidents further exacerbate this issue, making congestion prediction and management crucial for efficient urban mobility. Several approaches have been explored to tackle this issue [4]. Traditional traffic prediction models [5]–[7] rely on statistical methods and machine learning algorithms [8]. Recent advancements include deep learning-based techniques such as LSTMs and Bi-LSTMs for short-term traffic flow prediction [9], big data integration with deep learning for large-scale traffic forecasting [10], and hybrid swarm intelligence algorithms for route optimization [11]. While these approaches have demonstrated promising results, they face limitations such as high computational costs, lack of real-time adaptability, and challenges in handling noisy traffic data [12].

Apart from lowering traffic efficiency, urban traffic congestion incurs huge financial losses as well as environmental degradation. Traffic congestion in highly urbanized cities results in higher fuel usage and vehicle emissions, thereby contributing to higher risks to public health and air pollution, according to studies [13]. Additionally, the commuters themselves experience stress and reduced quality of life through traffic jams, making the critical need for smart traffic management systems arise that can learn to keep pace with the changing trends of urban traffic [14].

Advanced traffic control techniques like adaptive signal control, dynamic route guidance, and price-demand responsiveness are made possible through accurate and timely traffic congestion prediction. But urban transportation networks are highly spatial-temporal in complexity, so modeling them is quite hard Models with the capacity to discern local and network-wide interactions are needed since traffic at a point can spread and impact neighboring regions. To generate reliable, scalable, and comprehensible congestion predictions that facilitate smarter urban mobility planning, our system integrates static infrastructure data with dynamic traffic attributes through sophisticated clustering and graph-based learning techniques [15].

Despite these advancements, existing methods often fail to integrate both static (unvaried) and dynamic (varied) traffic factors effectively. Some studies focus on real-time traffic variations, such as vehicle speed and density, while neglecting the influence of fixed constraints like road infrastructure and historical traffic patterns [16]. Others emphasize static features, but lack adaptability to sudden traffic fluctuations caused by external factors like weather or road incidents [18], [19]. The main objective of the current research study is to answer the following research question: What advantages does the incorporation of unvaried and varied traffic features with enhanced clustering methods and graph-based classification bring to the accurate prediction of urban traffic congestion?

To answer this question, this study suggests a novel hybrid strategy that combines supervised and unsupervised learning methods to tackle both diversely changing (dynamic) and stably unchanging (static) traffic measurements concurrently. In particular, we suggest an optimized DBSCAN clustering algorithm that incorporates traffic-related domain constraints and improves parameter selection sensitivity to better handle heterogeneous and noisy traffic data. This allows traffic patterns to be more forcefully structured, reflecting the underlying structure of real-world situations.

We use the resulting cluster labels as an input layer for classifying traffic states after the clustering process. Then, with topological relationships between road segments, a Graph Convolutional Network (GCN) predicts traffic congestion levels. Spatial interdependencies and network-wide interactions underlying congestion dynamics are captured using this graph-based approach. Our model can make precise, scalable, and explainable predictions by integrating time-dependent features (e.g., speed of vehicles, flow, and density) with infrastructure-related static attributes (e.g., lane number and road length). Two-faceted learning in this way empowers real-time responsiveness and smart decision-making in urban traffic management systems, revealing a fuller picture of traffic status.

The remainder of this paper is structured as follows: Section II reviews the related work. Section III presents the problem definition and describes the use of DBSCAN for clustering. Section IV details our proposed approach. Section V discusses the experimental results and compares the performance of our method with existing approaches. Finally, Section VI concludes the paper and outlines future research directions.

II. Related Work

By merging traffic flow physics with spatio-temporal graph neural networks, this study [20] presents the Physics-Guided Spatio-Temporal Graph Neural Network (PG-STGNN), a model for predicting traffic flow at city intersections. With the help of a step-by-step technique, the model goal is to improve short term traffic prediction accuracy by merging essential traffic features like queue building and signal schedule. The framework has mainly applied to crossroads and short-term estimation tasks but however provides a structured manner of giving physical traffic dynamics in combination with deep learning approaches. the particularity can make it less relevant for more Far-reaching or sophisticated traffic Circumstances, Especially where long term analysis or congestion Segmentation is needed. Moreover, the framework might be less Adjustable if utilized in datasets with no such specific field Awareness since it relies on complex tangible simulation.

Authors in [21] presented three most crucial sections Are assimilated to present a novel AI-biased approach for predicting traffic congestion: a Consistent Lizard Search Optimization (CLSO) algorithm to elevate the precision of forecasting; a Paramount Transfer Learning Network (PTLN) for congestion level Classification; and a Cascaded Transition Recurrent Feature Network (CTRFN) to find Pertinent traffic features. The outcome is Created to Surmount major Disadvantages of current approaches with regard to computation operating costs, weak data processing, and extreme error prediction. The structure's reliance on heavy deep learning and optimization components may hamper scalability and real-time response capability, despite exemplifying an orderly approach in managing traffic data and trying to increase the accuracy of forecasts. In addition The system performance can be limited in noisy, incomplete, or heterogeneous urban traffic data environments in dynamic city scenarios due to the assumption of clean and structured dataset availability.

Through the use of linear regression models, feed forward neural networks (FFNN), and radial basis function neural networks (RBFNN), this study [22] proposes a machine learning-based prediction solution to reduce traffic congestion in Beirut. In forecasting congestion, the system takes into account wait times at intersections, time of day, day of week, holidays, and meteorological conditions (e.g., rain). FFNNs were found to work best. The absence of long-term transportation planning and the use of Lebanese reliance on poor infrastructure is also highlighted in the study. While the methodology considers various models and bases its analysis on real-time data, it overlooks key factors of implementation including data noise, scalability at the regional scale, and integration with big traffic management systems. The application of long-term traffic management techniques or the viability of implementation in the real world remains restricted.

To minimize traffic jams at intersections within smart cities, authors in [23] suggests a cloud-based traffic congestion prediction model with a hybrid Neuro-Fuzzy solution. It enhances traffic control automation with IoT sensor feeds and intelligent decision-making. The Neuro-Fuzzy model adapts dynamically to traffic patterns through combining the programmability of fuzzy logic and learning from neural networks. System scalability and data preparation are made possible through the cloud environment. The technique has several drawbacks, including no field testing, reliance on sensor accuracy, and no account being taken of the difficulties in large-scale roll-out and integration with current infrastructure. The model's effectiveness in real urban settings is undermined by its failure to project possible lag in cloud computing and difficulty in real-time decision-making amid dynamic traffic conditions.

To improve traffic prediction, this paper [24] introduces the Congestion-based Traffic Prediction Model (CTPM), a new model that combines congestion propagation patterns with existing prediction frameworks. CTPM uses external congestion data to enhance predictions without overriding existing systems, unlike classical models that often fail during abnormally fluctuating traffic conditions. In order to maximize resource allocation, including dynamic traffic control and lane management, the model highlights the significance of knowing how congestion develops across road networks. The strategy, however, has a number of drawbacks, including the need for sophisticated integration into existing infrastructure, the reliance on the amount and quality of available data, and difficulty handling small data sets. Besides, it is also unclear how well the model handles extremely dynamic, real-time environments, especially urban areas with non-uniform traffic patterns and weak sensor coverage.

To enhance the accuracy and scalability of machine learning-based models, authors in [25] suggests a mixed traffic flow prediction method that integrates historical data and road network topology. Since conventional models utilize single-point data and possess limited contextual knowledge regarding traffic behavior over a network, they often cannot identify real-time anomalies. To overcome this, the suggested method involves road interconnectivity structural information, enhancing prediction and making traffic control measures more efficient. The method is not without limitations, however, because the addition of topology expands processing requirements and model complexity, which may constrain real-time implementations. Further, usability in sparsely instrumented environments will be constrained because of a dependence on highly accurate and detailed network data.

While previous works on congestion prediction show some impact, limitations were identified for all of the analysed studies on this topic. Many of the studies that utilized DBSCAN-type methods, along with hybrid techniques, do not have the ability to effectively identify static and dynamic traffic flows from the same analysis. Typically, models that adopted this approach denote both static and dynamic as separate; thus limiting their ability to accurately predict congestion in real-time. Likewise, GNN and deep learning approaches do not take into consideration the significance of spatial relationships between road segments. Other studies also assumed the use of entirely structured and noise-free datasets, leading to a lack of robustness for these models in practice. Regardless of data types, models were seldom subjected to case-specific testing or developed to be scalable. Thus, these limitations supported the rationale for our proposed methodology that enhances the integration of improved DBSCAN clustering methods with GCN-based classification methods by considering both static and dynamic vehicular conditions, resulting in improved accuracy and increased feasibility.

III. Preliminaries

This section introduces the background material required for this work, including the character of the traffic data, graph learning, clustering algorithms, and an introduction to the utilized Graph Convolutional Networks (GCNs) to predict congestion.

A. Problem Definition

In traffic analysis, varied traffic factors (e.g., speed and number of vehicles) are dynamic and change rapidly depending on time, location, or road conditions. On the other hand, unvaried traffic factors (e.g., route dimensions, road type, infrastructure) remain constant over time but still play a crucial role in congestion. By subdividing traffic factors into these two categories, we can better model the traffic system, enabling us to:

1) Differentiate dynamic influences: The varied factors capture real-time fluctuations in traffic, whereas unvaried factors provide a stable foundation for traffic patterns.

2) Increase prediction accuracy: This separation allows more precise models, as varied factors contribute directly to congestion level fluctuations, while unvaried factors help to contextualize these changes.

3) Improve clustering and classification: It aids in selecting the most relevant features for clustering and classification algorithms, reducing noise in data processing by treating dynamic factors differently from static ones.

Clustering is essential in traffic congestion analysis [26] to manage the complexity and variability of traffic data by grouping similar patterns, such as “low”, “medium” and “high” congestion levels. DBSCAN is preferred for this task because it is robust to noise and outliers, which are common in traffic data, and it can discover clusters of varying shapes and densities without requiring the number of clusters to be specified in advance. This makes DBSCAN ideal for identifying natural congestion patterns. Clustering with DBSCAN enhances prediction accuracy, reduces noise, and allows more efficient traffic management and resource allocation.

GNNs can provide a preferred traffic congestion classification because they are well suited for handling the graph-like structure of traffic networks [27], where intersections are nodes and roads are edges. GNNs excel at capturing the varied and unvaried dependencies and relationships between connected traffic points, allowing for a more accurate prediction of congestion levels across the network. They can model how congestion at one point affects surrounding areas, providing a more context-aware classification. By leveraging this structure, GNNs improve the accuracy of congestion predictions and can efficiently classify traffic into “low”, “medium” and “high” congestion levels, even in complex, large-scale networks.

It is usually difficult for traditional models to properly combine both varied and unvaried congestion constraints. In order to overcome these constraints, this paper suggests a hybrid DBSCAN-GNN model that uses graph-based learning for classification and density-based clustering for congestion pattern identification.

B. DBSCAN Algorithm

DBSCAN technique [28], one of the density-based clustering techniques, is used to group data of any type out of a large amount of data when there are noises and outliers. The key benefit of the DBSCAN technique over partition and hierarchical cluster algorithm is as follows, as depicted in Fig. 1:

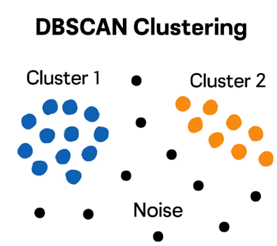

Fig. 1. Visualization of clusters formed by the DBSCAN algorithm.

- The DBSCAN algorithm is not limited to spherical or convex shapes but can be applied to any arbitrary shape. Because of this, it is far more practical than other algorithms like kmeans.
- Because the name itself indicates, the major benefit of this algorithm is that it will work properly even when there is noise and outliers. DBSCAN is a good method because other techniques will not work effectively with noise. (Those points which are different or greater than the original point are called noise and outliers.) With the help of Minpts in the clusters, it does not include noise and outliers.

The role of DBSCAN is the identification of the noise and clusters within a spatial database. Researchers need to know the Eps and MinPts parameters for each cluster as well as at least one point of the cluster. With the appropriate parameters, it can then identify all the point density-reachable from the known location provided [2]. The cluster is formed with the DBSCAN implementing the two significant parameters, minPts and eps:

- eps(ϵ): It is the radius which encloses the point to be used for looking for points near to it. It is utilized to calculate the density of the area. It will create low-density clusters when a huge ϵ value is utilized.
- minPts: It is the number of points that must be found in the surrounding (eps distance from the point) to form a cluster. A good algorithm must not have extremely small minPts.

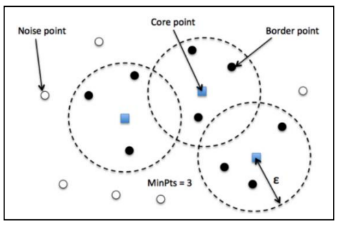

Fig. 2. DBSCAN clustering process showing core, border, and noise points.

Once the DBSCAN algorithm is executed on any data set, researchers essentially obtain three varieties of points (Fig. 2):

- Core: If there are many clusters, this is the spot from where there are at least k other points that fall inside radius r.
- Border: Any area with one or more central points in the distance r may be referred to as a border.
- Noise: Any point that contains at least k points within a distance of r and is neither a core nor a boundary.

IV. Proposed Approach

The predicted traffic congestion architecture is presented in this section. To learn spatial-temporal correlations and effectively predict the level of congestion, the model combines data collection, preprocessing, enhanced DBSCAN clustering, and a GCN-based classification module.

A. Architecture Overview

The architecture of the suggested traffic congestion forecasting framework is depicted in Fig. 3. To make precise predictions of congestion, the model is trained to utilize both dynamic and static traffic information. To capture continuous streams of traffic variables like vehicle speed, vehicle counts, lane counts, and route parameters, the pipeline starts with raw traffic data capture. To provide equivalence among disparate sensors as well as time periods, the information collected is preprocessed through cleaning, normalization, and temporal alignment.

The data is then divided into unvaried factors, constant features that remain the same over time like the number of lanes or route length, and varied factors, dynamic features that change, like traffic flow and vehicle speed. A better DBSCAN module utilizes these features and uses them to divide the road network according to congestion patterns and identify traffic trends. The model can denoise the input data and identify geographical interdependence by grouping similarly occurring traffic conditions.

A Graph Convolutional Network (GCN), mimicking the sophisticated spatial-temporal relationships between various road segments, subsequently receives the cluster data. For a correct prediction of congestion levels for every segment, the GCN leverages the road network topology and the traffic patterns learnt while clustering. The output layer finally offers real-time congestion level prediction, facilitating useful information for traffic control and route optimization.

Every stage of this architecture, from data preprocessing to clustering and GCN classification, improves prediction performance while ensuring computing economy. It is heavily based on modular and interpretable design. The model is amenable to smart city applications and real-time traffic analysis because it efficiently solves the problems caused by dynamic traffic environments through the integration of spatial and temporal information.

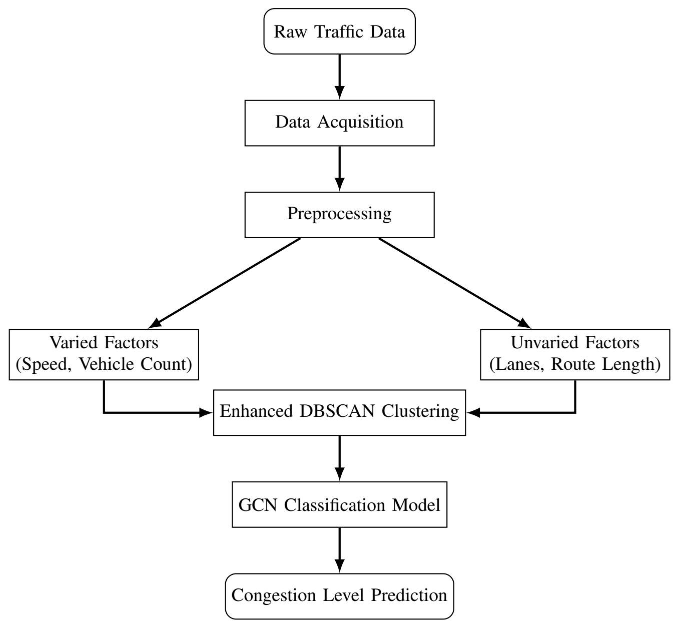

Fig. 3. Compact architecture overview of the proposed traffic congestion prediction model.

B. Traffic Congestion Factors

A variety of key restrictions that fall into two general categories varied and unvaried factors must be evaluated in order to forecast traffic congestion using Internet of Things networks. A few examples of generally fixed (unvaried) factors are the route's length, the average traffic flow, and customary vehicle behavior patterns. These components remain consistent over time and provide a fundamental understanding of traffic patterns. Varied factors present a more difficult challenge, as they are dynamic and change over time. Examples of these varied factors include weather, which can quickly alter road conditions; the route's distinctive and evolving nature (such as whether it is an urban road, highway, or frequently congested area); temporal elements such as rush hours or special events that significantly impact traffic volumes; the behavior of other vehicles on the road; the surrounding environment (such as roadwork and urban development); and the varying speeds of vehicles. Predicting traffic congestion is a very difficult challenge due to the intricate interactions between these two sets of constraints: fixed and dynamic.

Although unvaried factors help establish baseline traffic patterns, varied factors introduce an element of unpredictability, often requiring efficient real-time data management. For example, unforeseen road closures or adverse weather can suddenly turn an otherwise uncongested route into a bottleneck. Similarly, any reliable traffic prediction model must take into account time-dependent varied factors, such as rush hours, special events, or weekend traffic patterns. As shown in Table I, the predicting traffic congestion issue utilizing IoT networks is assessed using two key constraints which are unvaried such as, dimension of route distance, traffic low, pattern current vehicle conditions, and varied like : weather conditions, special nature of route, temporal nature of route, other vehicles pattern conditions, route environment, and vehicle speed. Our main objective is to develop a model that makes this task more efficient, because the related mechanisms particularly consider some constraints to deal with this problem, it is vital to explore all of them for effective congestion prediction.

A comprehensive knowledge of the interplay between these components is necessary for the precise forecasting of traffic congestion. Varied factors introduce layers of complexity that require dynamic handling, whereas unvaried factors set the scene by providing insights on typical traffic flow and bottleneck points. These elements work together to define the traffic environment, and it is crucial to take them into account when forecasting and handling congestion. The complexity of traffic circumstances emphasizes the value of using real-time data, advanced algorithms, and a thorough grasp of the different factors influencing traffic congestion.

Algorithm 1 DBSCAN Clustering with Congestion Estimation

Initialize:
Set core points $\Omega \leftarrow \emptyset$
Set unvisited points $F \leftarrow D$
Set clusters $C \leftarrow \emptyset$
Set cluster index $k \leftarrow 0$
Set congestion levels based on cluster density
Core Point Identification:
for each point x_i in D **do**
    Compute ϵ-neighborhood of x_i
    if number of neighbors $\geq$ MinPts **then**
        Add x_i to core points Ω
    end if
end for
Cluster Formation:
while Ω is not empty **do**
    Pick a core point o from Ω
    Create a new cluster C^k including o
    Expand the cluster with ϵ-neighbors of core points
    Remove processed points from F
    Update clusters C and increment k
    Remove core point o from Ω
end while
Estimate Congestion Levels:
for each cluster C^k **do**
    Compute cluster density
    if density is low **then**
        Assign congestion level to Low
    else if density is medium **then**
        Assign congestion level to Medium
    else
        Assign congestion level to High
    end if
end for
Handle Noise:
Points not included in any cluster are labeled as noise.

C. DBSCAN for Traffic Data Clustering

The Algorithm 1 is intended to cluster data via the DBSCAN technique, and then estimate the congestion level. It first initializes the cluster index and sets up the core points, unvisited points, and clusters. Finding each data point's ϵ-neighborhood and designating those with enough neighbors (more than or equal to MinPts) as core points constitute the core point identification step. The algorithm chooses core points iteratively, forms clusters, enlarges them with nearby core points, and updates the list of unvisited sites during the cluster building process. Lastly, it computes cluster densities and allocates congestion levels (Low, Medium, High) according to these densities in order to estimate congestion levels. Noise is defined as points that do not fit into any cluster.

D. Traffic Congestion Classification Using GNN with Temporal Extensions

Graph-structured data, such social connections or road networks, is processed using GNNs [36], a specific kind of neural network. While graphs have been the subject of applications for traditional neural networks such as Convolutional Neural Networks (CNNs) [37] and Recurrent Neural Networks (RNNs) [38], GNNs are better suited for these kinds of jobs. Since its debut in 2005, supervised, unsupervised, and reinforcement learning have all seen a significant increase in interest in GNNs because of their versatility.

As a robust mechanism For exploring graph structured data, Containing traffic networks, Graph Neural Networks (GNNs) transmit messages among nodes to record spatial associations. GNNs can effectively replicate the time changing character of traffic By integrating temporal frameworks like LSTM and learning Trends out of previous data to predict future traffic congestion. This fusion enables to utilize a dynamic framework in which Adjustments to the traffic network, Comparable to accidents or closures, could be provided in the graph Architecture as the system Is in training. Additionally, by Measuring the Meaning of Diverse nodes, Complex GNN topologies similar to Graph Attention Networks (GATs) [39] Enhance the prediction of accuracy Beyond enabling the model to concentrate on important congestion regions. because of these specifications, GNNs are Particularly well Appropriate for predicting real-time traffic congestion because they could Simultaneously capture Each of the two spatial and temporal dependencies.

TABLE I. THE PERFORMANCE OF DIFFERENT MODELS ON VARIOUS ROUTE AND TRAFFIC FEATURES

Studies	Feature Coverage								
	D	T	P	W	S	Tn	Ov	Re	Vs
[29]	✓	✓					✓		✓
[30]	✓			✓		✓			✓
[31]	✓		✓				✓	✓	✓
[32]	✓				✓			✓	✓
[33]	✓		✓				✓	✓	✓
[34]	✓			✓				✓	✓
[35]	✓		✓						✓
Our Model	✓	✓	✓	✓	✓	✓	✓	✓	✓

D	: **D**imension of Route Distance	T	: **T**raffic Flow	P	: **P**attern of Current Vehicle Conditions
W	: **W**eather Conditions	S	: **S**pecial Nature of Route	Tn	: **T**emporal Nature of Route
Ov	: **O**ther Vehicles Pattern Conditions	Re	: **R**oute Environment	Vs	: **V**ehicle Speed

GNNs are being used in smart cities for practical purposes [40], such as route planning and traffic light optimization to reduce congestion. GNNs are more effective at capturing intricate, non-grid road structures than previous models, which makes them more appropriate for large-scale, irregular road networks. Additionally, the model's ability to prioritize important traffic nodes is made possible by innovations like Graph Attention Networks (GATs), which raise the accuracy of congestion prediction.

The traffic data in a GNN with Temporal Extensions is shown as a graph G = (V, E), where:

- V The total number of nodes that represent traffic locations.
- E The set of edges (i.e., route links) that link each node.

By combining data from nearby nodes, a process known as graph convolution, the model is able to represent the spatial dependencies. Through this procedure, the model is able to understand how surrounding sites affect the traffic conditions at a certain location.

In order to simulate how traffic conditions change over time, the temporal part of the model incorporates temporal graph convolutions or recurrent mechanisms like gated recurrent units (GRUs) or LSTMs [41]. The network can anticipate traffic congestion in upcoming time steps because to the combination of temporal and graph modeling.

There are two primary parts to the temporal Graph Neural Network (GNN) model that was employed in this research. GNNs improve congestion class accuracy by allowing traffic information to spread between nodes, as opposed to traditional machine learning techniques, which treat individual road segments separately. The first layer is the Graph Convolution Layer, which represents the spatial connections between the graph's nodes. The graph convolution for a node $v \in V$ at time t is defined in Eq. (1) [42] as follows:

$$H_t^{(l+1)} = \sigma \left(\sum_{u \in N(v)} \frac{1}{c_{vu}} W^{(l)} H_t^{(l)}(u) \right) \tag{1}$$

where, $sigma$ is a non-linear activation function, $N(v)$ is the set of surrounding nodes of v, c_{vu} is the normalization constant, $W^{(l)}$ is the learnable weight matrix, and $H_t^{(l)}$ is the hidden state of node v at layer l and time t.

The Temporal Extension, a second element, simulates how node properties change over time. It is possible to employ a temporal convolution or a recurrent neural network (such as GRU or LSTM). For example, at time $t+1$, the hidden state of a node is changed as follows in Eq. (2) [43]:

$$H_{t+1} = \text{GRU}(H_t, X_t) \tag{2}$$

where, the traffic data input at time t is denoted by X_t.

The traffic observation network is represented as a graph in this study, with nodes standing in for traffic observation stations and edges for connectivity between places. At each node, traffic data, including vehicle count and speed, is gathered over time to create a time-series of graph-structured data.

For capturing the temporal dynamics of traffic across time, as well as the spatial dependencies between observation stations, a GNN with temporal extensions is used. While recurrent layers (such as GRU or LSTM) or temporal convolutions are utilized to record the time-varying characteristics of traffic at every node, graph convolutions are used to model the influence of surrounding nodes.

Using historic traffic information and interactions with other nodes through time, the model predicts future congestion at each monitored location. Three classifications - low, medium, and high congestion characterize the levels of congestion. The edges of the graph are weighted based on past congestion relationships, reflecting both spatial and temporal dependencies.

V. EXPERIMENTAL RESULTS AND EVALUATION

In this study, we evaluated a number of traffic congestion prediction models using actual traffic data and contrasted their overall effectiveness.

A. Dataset Description

The validation for the proposed methodology was carried out using the CityPulse dataset [44], which includes six months of real traffic data collected from 449 sensor nodes in Aarhus (Denmark). CityPulse was selected as a source of valuable information due to the wide-ranging nature of the type of traffic measured and the city's geographical location; it includes information about the quantity of traffic at different time intervals on several different types of roads (such as freeways and major thoroughfares) and at different locations (within the same metropolitan area).

In terms of traffic data collected over a six month period, the dataset includes a wide-ranging representation of the different types of traffic congestion in cities due to the variety of traffic conditions and types of roads represented within the CityPulse dataset. Also, the dataset provides data for many different types of attributes that may influence traffic congestion patterns, including, for example, the time taken to travel between two locations (travel time), the number of vehicles on the road (road occupancy), and the speed at which traffic is travelling (average speed).

The dataset include the following features: `status`, `avgMeasuredTime`, `avgSpeed`, `extID`, `medianMeasuredTime`, `TIMESTAMP`, `vehicleCount`, `_id`, and `REPORT_ID`.

Nevertheless, these factors do not address the static limitations (route length and capacity, or structural bottlenecks) that play an important role in shaping traffic behaviour.

To enhance the dataset, we incorporated fixed attributes like length of the road, how many lanes there are, and where these roads are located in space. These fixed attributes represent the limitations or constraints created by the road network itself. Conversely, varying attributes such as speeds, elapsed time, and number of vehicles on a road are dynamic attributes that change over time and document traffic's dynamic characteristics.

By merging both fixed and dynamic data to create a dataset, we have created a more complete and realistic framework for modelling congestion behavior. This enhanced dataset gives us the ability to quantify how traffic changes dynamically and how structural bottlenecks impact when and where congestion occurs along a road segment.

B. Enhanced DBSCAN Data Labelling

Once the dataset had been created, the next step in processing the original and enriched variables into effective congestion categories was accomplished using a modified version of the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm. A modified version of DBSCAN was chosen because the noise and variation present in real-world traffic datasets are difficult to handle even with the latest clustering methods. The DBSCAN algorithm has many advantages for use in urban traffic situations, particularly due to its ability to create clusters of differing dimension and to reject outliers, whereas traditional clustering algorithms such as k-means rely on the number of clusters and thus are limited to producing clusters of the same dimension.

Road segments were categorized into three main groupings by the clustering algorithm, which we subsequently designated as congestion levels:

- Low Congestion: Characterized by relatively low traffic volumes, higher average speeds, and short travel times. These conditions usually reflect free-flow traffic.
- Medium Congestion: Represents moderate traffic volumes and reduced average speeds, often corresponding to peak hours or transitional conditions where flow remains manageable but delays are present.
- High Congestion: Defined by high vehicle density, low average speeds, and longer measured times. These conditions reflect severe bottlenecks or critical saturation of the infrastructure.

This labelling step transforms the raw clustering output into interpretable and operational categories, which serve two purposes: 1) they provide a meaningful way to compare different traffic conditions across the city, and 2) they create a labelled dataset that can be used to train supervised learning models such as the GCN classifier.

In this way, the Enhanced DBSCAN not only organizes raw traffic data into coherent groups but also bridges the gap between unsupervised clustering and interpretable, real-world traffic levels.

C. Model Performance Analysis

From pre-clustered with labels for low, medium, and high congestion levels, the Graph Convolutional Network (GCN), being an exemplary model of the class of Graph Neural Networks (GNNs) [45], is used for traffic data classification. The graph-structured representation of traffic flow by the model accurately captures spatial interdependence among neighboring road segments. Each node represents road segments, and geographical correlations among segments are indicated by the edges. The GCN can model the complex non-linear relationships encountered in urban transport networks by considering geographical and temporal dependencies.

The hyperparameters used for GCN model training are listed in Table II. The network topology has three layers with a hidden dimension of 64, learning rate of 0.005, and ReLU activation function. The model was trained by an Adam optimizer for 30 epochs, with a weight decay of 0.0001 to control the model complexity and a dropout of 0.4 to prevent overfitting. 15 epochs of patience were used for early stopping for guaranteeing convergence and preventing unnecessary computation. For a trade-off between computing efficiency and stability, the batch size was chosen to be 128.

TABLE II. HYPERPARAMETERS USED FOR TRAINING THE GCN MODEL

Hyperparameter	Value
Learning Rate	0.005
Number of Layers	3
Hidden Dimension	64
Activation Function	ReLU
Batch Size	128
Dropout Rate	0.4
Weight Decay	0.0001
Number of Epochs	30
Optimizer	Adam
Early Stopping Patience	15

The good convergence and stability of the GCN model are confirmed by the training curves in Fig. 4. As the loss function reduces smoothly before settling to a small value, the accuracy curve rises very sharply in early epochs, saturating to a 93% accuracy. Good generalization and zero overfitting are ensured by the parallel nature of the training and validation curves. Model instability is not the cause of minor differences in validation accuracy, rather it is a result of natural traffic variation.

We evaluated the effectiveness of our suggested method by making comparisons with a number of state-of-the-art techniques, such as a hybrid CNN-LSTM model [9], a GAT-based technique [16], and Mouna's model [17]. A comparison in terms of important performance metrics, i.e., accuracy, precision, recall, F1-score, and area under the ROC curve (AUC), is given in Table III. Our approach based on GCN performs better than these models in all of the evaluation measures. It has well-balanced predictive capability for all types of congestion, with accuracy of 93%, 92% precision and recall, and F1-score of 0.92. The AUC of 0.95 indicates the model's outstanding discriminative capability to distinguish between types of congestion.

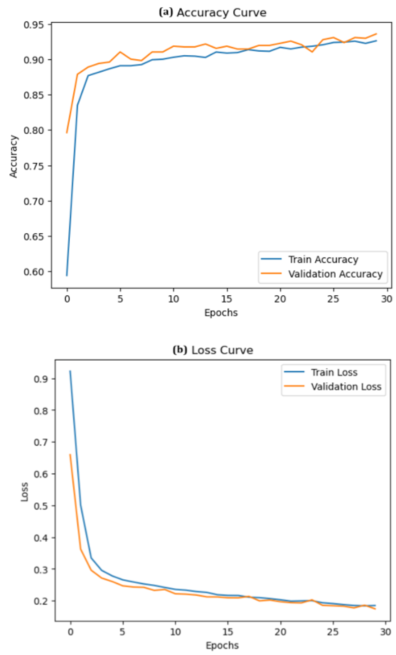

Fig. 4. Accuracy and loss curves for GCN model training, showing rapid accuracy growth and stable convergence.

TABLE III. Comparison of the Proposed GCN Model with Other State-of-the-Art Methods

Metric	GCN	Mouna [17]	CNN-LSTM [9]	GAT [16]
Precision	0.92	0.89	0.88	0.90
Recall	0.92	0.89	0.87	0.89
F1-Score	0.92	0.89	0.87	0.89
Accuracy	0.93	0.90	0.88	0.91
AUC	0.95	0.91	0.90	0.93

1) Computational efficiency and resource utilization: In the process of determining whether or not deep learning models are feasible to be used in large-scale traffic networks, computation efficiency is as vital as prediction accuracy. The experiment average memory (RAM) consumption, training time, and inference time are summarized in Table IV.

Training the proposed GCN model on an NVIDIA RTX 3060 with a RAM of 12 GB and **1.7 GB system RAM** took around **4 minutes and 52 seconds**. At a time of only **11.3 seconds** and an estimated RAM usage of around **0.9 GB**, the inference time (testing) was much faster. Our model has a good balance between accuracy and computational cost compared to the CNN-LSTM and GAT models, whose higher memory cost of over 2 GB was slower to train (6.4 and 5.7 minutes, respectively).

According to IV, the results prove that the proposed GCN-based technique provides superior performance when compared to existing approaches including CNN-LSTM, GAT and Mouna's model based upon the same set of performance metrics. The main advantages offered by our approach are:

- Improved prediction accuracy since the GCN was capable of better capturing spatial dependencies between road segments; therefore, creating an accurate representation of traffic congestion level for each road segment.
- Balanced performance across the different congested levels with high precision, recall and F1-score.
- The GCN model was able to perform computations more efficiently than other models, performing inference at a higher speed (throughput), and with less memory consumption, thus, allowing for its application in real-time traffic situations.

The above advantages indicate that our approach for predicting traffic congestion provides both greater accuracy and greater efficiency than existing state-of-the-art methodologies for predicting traffic flow congestion in dataset utilised in this analysis.

These results confirm that the targeted GCN model retains effective memory usage and speeds up inference while improving prediction performance. This makes it an effective fit in applications where fast and efficient processing is required, such as adaptive traffic control and real-time traffic congestion prediction.

Given everything being equal, the proposed approach attains an appropriate balance among hardware use, training effectiveness, and performance and hence constitutes a viable and competitive option for long-term use in smart transport systems.

D. Discussion

The results of the experiments show that by combining both structural (non-varying) and dynamic (varying) factors, we achieved greater accuracy in classifying traffic congestion levels. The enhanced version of the DBSCAN algorithm enabled the conversion of different types of traffic data into usable traffic congestion levels. This allowed the GCN classifier to distinguish between different traffic behaviours that were not captured by other models. This supports our theory that the use of density-based clustering in conjunction with graph-based classification will provide an improved method of capturing the spatial and temporal characteristics of traffic flow.

When compared with CNN-LSTM, GAT, and Mouna's [17] proposed model, the GCN-based approach achieves better accuracy, precision, recall and Area Under the ROC Curve (AUC). The GCN has an advantage over these sequence-based or attention-based models that treat each segment independently, because of its ability to explicitly model relationships between adjacent road segments. The improved AUC value of 0.95 demonstrates that the model is highly capable of distinguishing between medium and high congestion levels, which tend to be the hardest to differentiate from one another.

An analysis of the efficiency of the proposed model shows that the proposed model achieves competitive training and inference speed, as well as efficiency, while using a much lower amount of GPU Memory in comparison to CNN-LSTM and GAT. Therefore, as a result of this balance between the model's ability to predict accurately and the cost of computation, the proposed model is well-suited to use in the real-time traffic monitoring environment, which requires rapid decisions and limited resources.

VI. Conclusion and Future Work

We introduced a novel approach for predicting traffic congestion in urban environments by utilizing an enhanced DBSCAN clustering technique and a GCN Classifier. Using a 6-Month Traffic Dataset from Aarhus, Denmark, we concluded that combining both static and dynamic (evolving) Traffic Factors together will increase accuracy when predicting the level of congested traffic. Our enhanced DBSCAN method provides a way to convert multiplexed types of traffic data into usable levels of congestion and allows for the capture of spatial relationships between adjacent Roads or Segments of Road via a GCN to further distinguish between Medium and High levels of Traffic Congestion. Our methodology outperforms current leading approaches, including CNN-LSTM, GAT and Mouna's [17], in terms of Accuracy (93%), Precision (92%), Recall (92%) and AUC (0.95), while also achieving a balance between performance and speed which

TABLE IV. COMPUTATIONAL EFFICIENCY AND MEMORY CONSUMPTION COMPARISON BETWEEN MODELS

Model	Train Time (min)	Test Time (s)	RAM (GB)	GPU (GB)
Proposed GCN	4.87	11.3	1.7	2.3
CNN-LSTM [9]	6.42	15.8	2.4	3.1
GAT [16]	5.73	13.5	2.1	2.8
Mouna [17]	5.15	12.7	1.9	2.6

make it appropriate for inclusion in a real-time Traffic Monitoring Application and Intelligent Traffic Management System.

In future research, we will include various contextual elements (like weather patterns, public events, and car wrecks) as input features to create more robust predictive models. We will also expand our temporal modeling technique to forecast traffic congestion further into the future (i.e., hours, days, weeks, etc.). Finally, we are interested in how our traffic prediction method can be integrated into the development of adaptive traffic control systems that respond dynamically to changing traffic conditions in real time. The work outlined above has been developed to make our traffic prediction method robust and able to be used for a wide range of smart traffic management and urban planning applications in rapidly changing and varied urban settings.

REFERENCES

[1] M. Hassan and A. Arabiat, "An evaluation of multiple classifiers for traffic congestion prediction in Jordan," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 36, no. 1, pp. 461–468, 2024. https://doi.org/10.11591/ijeecs.v36.i1.pp461-468.

[2] C. Kanzouai, A. Zannou, E. H. Nfaoui and A. Boulaalam, "Query Processing in IoT Based on Spatial and Temporal Information," in *ICDTA 2022*. doi: 10.1007/978-3-031-02447-4_28

[3] S. Bouarourou, C. Kanzouai, A. Zannou, E. H. Nfaoui and A. Boulaalam, "Optimized Clustering Approach for WSN Using WCLUST Algorithm," *IRASET 2025*, Fez, Morocco, 2025, pp. 1–6. doi: 10.1109/IRASET64571.2025.11008085

[4] A. Zannou, E. H. Nfaoui, A. Boulaalam, M. Fariss, S. Bouarourou and N. El Allali, "Distributed Caching System in a Decentralized Network of IoT," *IRASET 2025*. doi: 10.1109/IRASET64571.2025.11008305

[5] S. Bouarourou, A. Zannou, E. H. Nfaoui, C. Kanzouai and A. Boulaalam, "An Energy-Efficient Pathfinding Model for Wireless Sensor Networks in IoT Using Whale Optimization Algorithm," *Statistics, Optimization & Information Computing*, 2025. doi: 10.19139/soic-2310-5070-2433

[6] C. Kanzouai, S. Bouarourou, A. Zannou, A. Boulaalam and E. H. Nfaoui, "Enhancing IoT Scalability and Interoperability Through Ontology Alignment and FedProx," *Future Internet*, 2025. doi: 10.3390/fi17040140

[7] A. Zannou, E. H. Nfaoui, A. Boulaalam, N. E. Allali and M. Fariss, "Data Gathering from IoT Networks," *CiSt 2023*. doi: 10.1109/CiSt56084.2023.10409956

[8] Author(s) Name, "The Data Stream Distribution and Congestion Control in NoC," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 11, no. 12, pp. 7392–7398, 2013. https://doi.org/10.11591/telkomnika.v11i12.3987.

[9] P. Redhu and K. Kumar, "Short-term traffic flow prediction based on optimized deep learning neural network: PSO-Bi-LSTM," *Physica A: Statistical Mechanics and its Applications*, vol. 625, p. 129001, 2023. https://doi.org/10.1016/j.physa.2023.129001.

[10] I. Katib and S. M. Altowaijri, "Smarter traffic prediction using big data, in-memory computing, deep learning and GPUs," *Sensors*, vol. 19, no. 9, p. 2206, 2019. https://doi.org/10.3390/s19092206.

[11] R. Beed, A. Roy, S. Sarkar, and D. Bhattacharya, "A hybrid multi-objective tour route optimization algorithm based on particle swarm optimization and artificial bee colony optimization," *Computational Intelligence*, vol. 36, no. 3, pp. 884–909, 2020. https://doi.org/10.1111/coin.12276.

[12] N. I. Ramli and M. I. Mohamed Rawi, "An overview of traffic congestion detection and classification techniques in VANET," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 437–444, 2020. https://doi.org/10.11591/ijeecs.v20.i1.pp437-444.

[13] W. Jiang, "Cellular traffic prediction with machine learning: A survey," *Expert Systems with Applications*, vol. 201, p. 117163, 2022. https://doi.org/10.1016/j.eswa.2022.117163.

[14] H. Yuan and G. Li, "A survey of traffic prediction: from spatio-temporal data to intelligent transportation," *Data Science and Engineering*, vol. 6, no. 1, pp. 63–85, 2021. https://doi.org/10.1007/s41019-020-00151-z.

[15] S. A. Sayed, Y. Abdel-Hamid, and H. A. Hefny, "Artificial intelligence-based traffic flow prediction: a comprehensive review," *Journal of Electrical Systems and Information Technology*, vol. 10, no. 1, p. 13, 2023. https://doi.org/10.1186/s43067-023-00081-6.

[16] S. J. Elias, S. M. Hatim, M. Y. Darus, S. Abdullah, J. Jasmis, R. B. Ahmad, and A. W. Y. Khang, "Congestion control in vehicular adhoc network: a survey," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 3, pp. 1280–1285, 2019. https://doi.org/10.11591/ijeecs.v13.i3.pp1280-1285.

[17] M. Z. Mehdi, H. M. Kammoun, N. G. Benayed, D. Sellami, and A. D. Masmoudi, "Entropy-based traffic flow labeling for CNN-based traffic congestion prediction from meta-parameters," *IEEE Access*, vol. 10, pp. 16123–16133, 2022. https://doi.org/10.1109/ACCESS.2022.3180484.

[18] X. Fu, H. Gao, H. Cai, Z. Wang, and W. Chen, "How to Improve Urban Intelligent Traffic? A Case Study Using Traffic Signal Timing Optimization Model Based on Swarm Intelligence Algorithm," *Sensors*, vol. 21, no. 8, p. 2631, 2021. https://doi.org/10.3390/s21082631.

[19] M. Z. Mehdi, H. M. Kammoun, N. G. Benayed, D. Sellami, and A. D. Masmoudi, "Entropy-based traffic flow labeling for CNN-based traffic congestion prediction from meta-parameters," *IEEE Access*, vol. 10, pp. 16123–16133, 2022. https://doi.org/10.1109/ACCESS.2022.3148782.

[20] Y. A. Pan, F. Li, A. Li, Z. Niu, and Z. Liu, "Urban intersection traffic flow prediction: A physics-guided stepwise framework utilizing spatio-temporal graph neural network algorithms," *Multimodal Transportation*, vol. 4, no. 2, p. 100207, 2025. https://doi.org/10.1016/j.multra.2025.100207.

[21] K. Balasubramani and U. Natarajan, "A Cascaded transition recurrent feature network (CTRFN) based Paramount Transfer learning (PTL) model for traffic congestion prediction," *Expert Systems with Applications*, vol. 248, p. 123446, 2024. https://doi.org/10.1016/j.eswa.2024.123446.

[22] W. Fahs, F. Chbib, A. Rammal, R. Khatoun, A. El Attar, I. Zaytoun, and J. Hachem, "Traffic Congestion Prediction Based on Multivariate Modelling and Neural Networks Regressions," *Procedia Computer Science*, vol. 220, pp. 202–209, 2023. (The 14th International Conference on Ambient Systems, Networks and Technologies (ANT) and The 6th International Conference on Emerging Data and Industry 4.0 (EDI40)) https://doi.org/10.1016/j.procs.2023.03.028.

[23] M. Gollapalli, A. Rahman, D. Musleh, N. Ibrahim, M. A. Khan, S. Abbas, A. Atta, M. A. Khan, M. Farooqui, T. Iqbal, M. S. Ahmed, M. I. B. Ahmed, D. Almoqbil, M. Nabeel, and A. Omer, "A Neuro-Fuzzy Approach to Road Traffic Congestion Prediction," *Computers, Materials and Continua*, vol. 73, no. 1, pp. 295–310, 2022. https://doi.org/10.32604/cmc.2022.027925.

[24] A. M. Nagy and V. Simon, "Improving traffic prediction using congestion propagation patterns in smart cities," *Advanced Engineering Informatics*, vol. 50, p. 101343, 2021. https://doi.org/10.1016/j.aei.2021.101343.

[25] Z. Wang, P. Sun, Y. Hu, and A. Boukerche, "A novel hybrid method for achieving accurate and timeliness vehicular traffic flow prediction in road networks," *Computer Communications*, vol. 209, pp. 378–386, 2023. https://doi.org/10.1016/j.comcom.2023.07.019.

[26] H. Yuan and G. Li, "A survey of traffic prediction: from spatio-temporal data to intelligent transportation," *Data Science and Engineering*, vol. 6, no. 1, pp. 63–85, 2021. https://doi.org/10.1007/s41019-020-00143-1.

[27] W. Jiang and J. Luo, ”Graph neural network for traffic forecasting: A survey,” *Expert Systems with Applications*, vol. 207, p. 117921, 2022. https://doi.org/10.1016/j.eswa.2022.117921.

[28] H. V. Singh, A. Girdhar, and S. Dahiya, “A Literature Survey Based on DBSCAN Algorithms,” in *Proceedings of the 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2022, pp. 751–758, doi: 10.1109/ICICCS53718.2022.9788098.

[29] E. Walraven, M. T. Spaan, and B. Bakker, ”Traffic flow optimization: A reinforcement learning approach,” *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 203–212, 2016. https://doi.org/10.1016/j.engappai.2016.02.009.

[30] Y. Liu and H. Wu, ”Prediction of road traffic congestion based on random forest,” in *Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, pp. 361–364, IEEE, 2017. https://doi.org/10.1109/ISCID.2017.23.

[31] S. Devi and T. Neetha, ”Machine learning based traffic congestion prediction in an IoT based smart city,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, pp. 3442–3445, 2017. https://doi.org/10.6084/m9.figshare.4920601.

[32] A. Khanna, R. Goyal, M. Verma, and D. Joshi, ”Intelligent traffic management system for smart cities,” in *International Conference on Futuristic Trends in Network and Communication Technologies*, Springer, 2018, pp. 152–164. https://doi.org/10.1007/978-981-13-3225-5_14.

[33] S. Malik and D. Kim, ”Optimal travel route recommendation mechanism based on neural networks and particle swarm optimization for efficient tourism using tourist vehicular data,” *Sustainability*, vol. 11, no. 12, p. 3357, 2019. https://doi.org/10.3390/su11123357.

[34] R. Beed, A. Roy, S. Sarkar, and D. Bhattacharya, ”A hybrid multi-objective tour route optimization algorithm based on particle swarm optimization and artificial bee colony optimization,” *Computational Intelligence*, vol. 36, no. 3, pp. 884–909, 2020. https://doi.org/10.1111/coin.12234.

[35] X. Fu, H. Gao, H. Cai, Z. Wang, and W. Chen, ”How to improve urban intelligent traffic? A case study using traffic signal timing optimization model based on swarm intelligence algorithm,” *Sensors*, vol. 21, no. 8, p. 2631, 2021. https://doi.org/10.3390/s21082631.

[36] Y. Shao, H. Li, X. Gu, H. Yin, Y. Li, X. Miao, and L. Chen, ”Distributed graph neural network training: A survey,” *ACM Computing Surveys*, vol. 56, no. 8, pp. 1–39, 2024. https://doi.org/10.1145/3443711.

[37] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, ”A survey of convolutional neural networks: analysis, applications, and prospects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2021. https://doi.org/10.1109/TNNLS.2021.3072338.

[38] Y. Yu, X. Si, C. Hu, and J. Zhang, ”A review of recurrent neural networks: LSTM cells and network architectures,” *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019. https://doi.org/10.1162/neco_a_01148

[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, ”Graph attention networks,” in *arXiv preprint arXiv:1710.10903*, 2017. https://doi.org/10.48550/arXiv.1710.10903.

[40] G. Dong, M. Tang, Z. Wang, *et al.*, ”Graph neural networks in IoT: A survey,” *ACM Transactions on Sensor Networks*, vol. 19, no. 2, pp. 1–50, 2023. https://doi.org/10.1145/3467999.

[41] Y. Yu, X. Si, C. Hu, and J. Zhang, ”A review of recurrent neural networks: LSTM cells and network architectures,” *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019. https://doi.org/10.1162/neco_a_01148.

[42] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in *International Conference on Learning Representations (ICLR)*, 2017.

[43] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[44] CityPulse Project, ”Data-Driven Smart City Services,” 2014. Available: http://iot.ee.surrey.ac.uk:8080/datasets.html#traffic.

[45] U. A. Bhatti, H. Tang, G. Wu, S. Marjan, and A. Hussain, ”Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence,” *International Journal of Intelligent Systems*, vol. 2023, no. 1, p. 8342104, 2023. https://doi.org/10.1002/int.8342104.