International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Navigating Urban Gridlock: Deep Learning for Predicting City-Wide Traffic Congestion in Smart Cities

S. Jana[1*], Arun Aram[2], B.Yuvaraj[3], K. Jaya Deepthi[4], K. Manikandan[5], Ch Veera Kiranmayi[6]

Submitted: 13/05/2024 **Revised**: 26/06/2024 **Accepted**: 06/07/2024

Abstract: Traffic congestion prediction has a significant part in managing smart city networks, helping authorities to reduce traffic jams and make transportation systems function well. Even with progress made in deep learning methods, it is common for current ways to experience difficulty when trying to predict congestion patterns accurately. The paper tackles these shortcomings by suggesting a fresh Dynamic Traffic Prediction Network (DTPN) model that blends MobileNet Recurrent Convolutional Network (MRCN) for spatial-temporal traits extraction and Sonar Sweep Optimization (SoSO) as parameter adjustment technique. The newness of our method is in how it can accurately catch the spatial and time-based changes in traffic flow, which results better prediction. When we evaluate using Kaggle datasets, we see that DTPN model does better than other methods for precision, recall, F1-score as well as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The suggested model shows excellent precision of 99%, recall at 98.9% while F1-score is up to 99; these figures are past what Random Forest (RF), XGBoost (XGB), Light GBM (LGBM) and Improved May Fly Optimization with Light GBM (IMFO-LGBM). Moreover, when we compare MAE and RMSE results, it becomes clear that the DTPN model outperforms other methods in various time ranges. This comparison provides more evidence to support the effectiveness of this method for improving prediction accuracy in traffic congestion. It shows promise for a future where urban transport systems are smarter and more efficient.

Keywords: *Traffic Congestion Prediction, Smart City, Deep Learning, Artificial Intelligence (AI), Traffic Management, and Optimization.*

1.Introduction

The economy is growing, cities are developing fast and people want to travel privately [1]. This has led to a big increase in traffic congestion level for many large or rapidly expanding cities all over the world. The problem of traffic jam directly impacts the growth rate, development speed and environment quality in these places. As economic growth takes place and cities become more attractive for people to live there, it also brings an increasing demand on transportation infrastructure—needed both by those who work in these areas as well as those wanting goods moved or travelling for leisure purposes—to keep up with this urban expansion [2, 3]. But often times building new infrastructures does not happen at same speed like how quickly urban areas grow causing problems such as congestion which can affect city life greatly. The growing wish to have personal cars, pushed by more money for spending and cheaper vehicles, is also making the problem worse. As a result, when there are more private cars on the street it causes increased traffic volume especially at busy times of day. The unbalanced increase in population and lack of growth in infrastructure significantly adds to congestion that results in many negative impacts. Traffic congestion [4] not only slows down transportation but also causes serious environmental problems. The fumes from vehicles that are not moving add to air pollution and greenhouse gases, which makes air worse and increases global warming [5, 6]. Also, when the streets are crowded with cars and trucks, it will take more time to travel. This impacts every person and different parts of our economy. For workers, an increase in travel duration means they have less time for actual work or enjoying life beyond their job's official hours. This also reduces their ability to produce output and impacts the overall quality of life. The irritation of being caught in traffic may result in road fury and intense driving manners, increasing the chances for accidents. Moreover, crowded traffic situations might lead to disorderly driving circumstances that enhance probabilities of accidents [7]. The stopping and starting of vehicles in heavy traffic, together with the high number of cars on the road, can

[1]Professor, Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India, ***Email: seljana.1995@gmail.com*(Corresponding Author)***

[2]Post-graduate Resident, Department of Radio-Diagnosis, Saveetha Medical College and Hospital, Saveetha Institute of Medical and Technical Sciences (SIMATS), Saveetha University, Chennai, Tamil Nadu - 602105, India. ***Email: drarunaram007@gmail.com***

[3]Associate Professor, Department of Computer Science and Engineering, Kings Engineering College, Sriperumbudur, Tamil Nadu 602117. ***Email: byuvarajb@gmail.com***

[4]Assistant professor, Department of Artificial Intelligence and Machine Learning
School of Computing, Mohan Babu University, Tirupati-517102 Andhra Pradesh. ***Email: deepthi.kaluva@gmail.com***

[5]Assistant professor, Department of Computer Science and Engineering, P.S.R Engineering College, Sivakasi -626140, Tamil Nadu. ***Email: manikandan.k@psr.edu.in***

[6]Associate Professor, Department of Electronics & Communication Engineering, Aditya Engineering College, Surampalem, India, ***Email: kiranmaich@aec.edu.in***

create a situation where unpredicted accidents happen. The total effect of traffic congestion is a big problem that needs systematic solutions like putting money into public transport; enhancing infrastructure quality such as roads and bridges; planning land use properly along with promoting other means for traveling apart from using private automobiles. Working on these aspects can assist cities in lessening issues related to excessive traffic, as well as render living arrangements inside the cities more sustainable and agreeable for all parties involved.

Hence, traffic management study [8] is very important for researchers in present times. We can reduce high congestion by two ways: one way is to add more transportation infrastructure which costs a lot of money; second way includes using possible traffic strategies like analyzing congestion pattern or making short-term traffic information prediction that can be applied quickly on existing road networks and it's just a small part of the total cost. When compare it to pattern analysis, which finds out about road networks that have repeated problems with too much traffic, predicting exact short-term information related to traffic like speed, volume and level of congestion becomes more useful for people who are traveling as well as those managing the flow of vehicles [9]. From these metrics, one parameter that is especially wanted and beneficial for short-term traffic forecasting has to be the traffic congestion level. This shows the condition of road network (like Jam, Slow or Free) which allows drivers to select routes better by avoiding congested roads. It also aids in enhancing efficiency of traffic managers who can react systematically towards variations in transport network's supply-demand balance. In this way, precise short-term predictions about traffic serve as a useful instrument for improving flow of cars on roads and lessening issues related with congestion in city environments.

The beginning forecasting models [10, 11] mainly concentrated on estimating traffic characteristics like speed, volume and flow of vehicles in solitary roads, sets of roads or minor road networks. These models had a restriction because the complete data necessary was not easily accessible, so it limited their extent. Thus, these initial models only gave incomplete prediction abilities that were not very helpful for people who travel to work or organizations managing traffic looking for more detailed and useful understandings. Hence, these models did not receive extensive use and could not successfully tackle the larger requirements of city traffic management. In this first type of model, the data usually came from sensors that are fixed on roads [12]. These could be road sensors, inductive loops or traffic cameras. Some models also used data from networks of vehicles like Vehicular Ad Hoc Networks (VANETs) and Floating Car Data. In these cases, the cars themselves become a source for information as they move along various routes. The provided data was useful but it was not easy to gather consistently and required considerable effort to process into useful form for modeling purposes [13, 14]. These were some difficulties: Fixed sensors are costly in terms of installation, running and keeping them well-maintained. This kind of expense could make it hard to set up many fixed sensors throughout an area. Additionally, the continuing costs for maintaining these sensors so that they continue giving precise and dependable data over time are also taken into account. Getting data from these sources is not easy because of privacy and regulation issues. For example, to gather data from traffic cameras and other fixed sensors often needs special permissions that can be quite bureaucratic - it takes time to get these licenses. It make complexity to acquire the complete data needed for accurate and prompt traffic forecasts. Moreover, the difference in data quality and uniformity among sensors contributes to a fluctuation in accuracy of traffic predictions. These limitations showed that there was a demand for better and broader ways of gathering data, along with more complex predictive models which could offer insights about traffic throughout the whole city. The dependence on fixed sensor information had scaling problems, pushing researchers to look at other sources of data and new ways to enhance forecasting. This change aimed at handling the flaws in initial models and providing more efficient methods for controlling urban traffic jam situations, benefiting both people who travel daily as well as those who manage traffic from their work perspective.

The issue of traffic jam prediction has become very important as cities grow and face complex transportation problems. The old ways to forecast traffic, using fixed sensors along with limited data suppliers, cannot give total real-time understanding throughout the whole city. Web services like Google Traffic, Bing, Seoul Transportation Operation and Information Service (TOPIS) [15] are now bringing new perspectives in this field by giving precise city-wide real-time traffic information. Data from many places is gathered and studied. For instance, data from GPS that comes out of people's smartphones, information coming in from road sensors or traffic cameras - all these are combined to give an instant picture of how the traffic situation looks like. These kind of web services can provide abundant data and high-quality standards, factors that can enhance precision and dependability in traffic forecasts. The up-to-date traffic particulars from these services assist in developing better predictive models for foreseeing congestions without much lag time. This active information about traffic aids the creation of flexible management plans before congestion turns into a problem [16]. Also, these web services are simpler to use and less costly compared to the old ways of collecting data. This lowers the demand for expensive building and upkeep of infrastructure.

However, even with their potential advantages, deep learning models for traffic congestion prediction also come with some significant limitations [17]. These models need a lot of good quality data sets to train on. It might be hard to gather and keep these data sets, especially in real-time and across big city areas. Also, deep learning models are very demanding on computer power and memory resources. This makes live prediction complicated and requires a big investment in high-performance computing infrastructure. So, the proposed work could be related to developing a particular and intelligent system for recognizing traffic jams in networks of smart cities by utilizing images from traffic [18]. This approach takes advantage of the common existence of traffic cameras and image information to offer immediate accurate forecast about crowded regions. The structure uses sophisticated methods for image processing and deep learning [19, 20]. It can examine the traffic images, locate congested patterns and provide prompt information to people who travel as well as traffic management agencies. The addition of traffic images also improves model interpretability because visual data is easier to understand and can be verified by human operators. The idea of this creative system is to enhance city movement, reduce crowding, and contribute in constructing more intelligent city transport systems. The main research objectives of this work are given below:

- This develop a new structure that uses sophisticated methods to improve the precision of forecasting traffic jams on smart city networks.
- To implement the MobileNet Recurrent Convolutional Network (MRCN) as an important part of the framework, created to adeptly capture spatial-temporal characteristics from traffic data.
- To incorporate the Sonar Sweep Optimization (SoSO) model, a fresh optimization method for tuning the framework's parameters and enhance its performance.
- Also, enhance the framework more effective in predicting traffic congestion by smoothly combining MRCN and SoSO techniques. This will improve the accuracy of predictions for smart city networks, offering a stronger solution.

The paper is organized as follows: Section 2, a thorough review of literature about traffic jam prediction models. This area deeply examines different methods ranging from old statistical ways to complex deep learning techniques and their advantages as well as disadvantages. Section 3, an understandable explanation of the suggested model. This part gives a detailed path for the process and algorithmic explanations which show how traffic images are used to detect congestion. In section 4, we discuss about performance outcomes of this proposed work that include results compared with existing models to show its effectiveness and enhancements. To end, Section 5 gives a wrap-up of the paper's results and offers ideas for upcoming studies to improve traffic congestion prediction in smart city networks.

2. Related Works

In this section, we examine the current research on anticipating traffic congestion in smart city settings. There is a special emphasis placed on typical machine learning and deep learning methods. Through study of appropriate literature, we intend to give an all-encompassing view of the ways, algorithms and approaches used for predicting traffic jams. This review acts as groundwork for comprehending how models for prediction have changed over time within this area and recognizing important patterns and obstacles that influence present study tasks.

Bai, et al [21] introduced a new method for predicting traffic jam using Relative Position Congestion Tensor and Predictor for Position Congestion Tensor. The goal of the authors is to solve the difficulty in accurately guessing traffic congestion on city road networks by using spatio-temporal data and deep learning methods. This technique works with relative places of road nodes, which is different from usual ways that often concentrate only on absolute locations or basic traffic details such as speed and amount. These matrices get changed to three-dimensional spatio-temporal tensors, which offer an extensive representation of traffic data across time and area. The model's need for large and good traffic data is a big difficulty, because it can take much effort to get and keep these sets of information. Also, the complexity of computational work in ConvLSTM networks could restrict how easy it is to increase the size of model and use it in real time. This situation may be more difficult for traffic management groups that do not have much access to strong computing facilities. Lastly, not being able to interpret the predictions from this model might slow down its acceptance by professionals who manage traffic situations. Akhtar, et al [22] orderly wraps up previous research that has been done by using different methods of artificial intelligence, especially various machine learning models. It neatly classifies these models into branches of AI and gives a complete view on their strong points and drawbacks. This way of arranging information helps readers to get a straightforward grasp on the wide-ranging field of AI-centered solutions for certain problems. In this way, the paper combines what literature already exists to give important understandings about the present situation of AI research and to show where more study is needed or enhancements are possible. The mention of artificial intelligence development and availability of big data points towards a shift in traffic management, where researchers are using modern modeling methods to deal with this complex matter. In general, the sentence gives a clear summary about why predicting traffic jams is so important now in today's transport control system along with how new technologies can help push forward researches related to this area.

Khan, et al [23] talks about a very important missing part in the field. It suggests an effective scheme for predicting traffic flow using group techniques, like bagging, combined with air pollution data. This method is especially useful because it helps to make precise traffic flow prediction systems in smart cities more accurate. These systems need advanced methods to handle growing congestion problems efficiently. The goal of the research is to predict traffic flow and it has two parts: first compare various simple regression techniques to find best model performing; second use bagging and stacking ensemble methods for improving prediction accuracy afterwards. However, the model's usefulness could be restricted by the necessity for pollution data, which might not be easy to get in some places. Also, the computational difficulty of ensemble techniques such as bagging might make it difficult to apply them immediately in environments with limited resources. Furthermore, this study mainly concentrates on regression models and could overlook other sophisticated machine learning methods that could boost prediction precision. Devi, et al [24] examined the advantages of traffic and safety, tackling a big current problem. The dataset's classification is done with the help of a Support Vector Machine (SVM), and regression techniques - consisting of linear regression along with logistic regression. The method is successful because it combines high-level data analysis with machine learning, creating one system that gives useful comprehension about traffic jams. This understanding of information might be helpful in making improved procedures for controlling and keeping traffic safe in cities. But, this approach could not have easy generalizability because it relies on particular tools and methods for classifying data. The conclusion might need more confirmation in different datasets and conditions to improve its applicability.

Campbell, et al [25] suggested an economical method to create custom-made street sign computer vision datasets using Google's Street View API. The study included teaching a personalized object finding model to identify and sort Stop as well as Give Way signs in pictures taken at intersection approaches. This shows a practical solution for traffic sign detection that may save money, possibly helping to enhance the effectiveness of managing traffic assets. But the paper does not talk about how this model would handle detecting more kinds of traffic signs or its performance in different environmental situations. Also, depending on GSV images could restrict its use in real-time because there might be some delay when updating data. AI-qudah, et al [26] suggested a model that uses technology to help with the suggested structure for predicting traffic congestion. The authors show how the overall framework and their technology model are related. They also set up an intelligent image handling system made specifically for people not skilled in technical matters, bringing attention to importance and usefulness of this suggested structure. Yet, the paper does take note of some possible restrictions in terms of expense, security worries and functional difficulty which could slow down cities' acceptance of these solutions. To deal with these issues, it is recommended that a general image processing model be created and put into use by using deep learning techniques. In summary, this paper gives useful understanding about how technology interacts with managing cities and transport methods. It presents not just ideas but also real ways to reduce traffic congestion problems in urban areas through both theory-based concepts as well as feasible applications.

Ismaeel, et al [27] explained how deep recurrent neural networks (RNNs) can be used for sorting out different kinds of traffic patterns in smart cities. A new way to classify traffic patterns based on deep recurrent neural networks is introduced. These types of networks are good at understanding the changing and sequential features of traffic movements. The suggested model combines layers with convolutional and recurrent features for extracting characteristics from data about traffic patterns, along with a SoftMax layer to classify these kinds of movements. Henceforth, forthcoming investigation could delve into techniques for improving the intricacy of models and computational efficiency without affecting classification precision much – thus making these deep RNNs more applicable within smart city settings where classifying patterns related with traffic becomes crucial. Jenifer, et al [28] put forward two important enhancements for feature engineering and training datasets in smart city applications. Initially, it offers an approach to feature engineering that involves both extraction of characteristics as well as using a nature-inspired optimization algorithm for picking the most appropriate features. The Mayfly optimization algorithm gets improved with the addition of a mode-based ranking method; this makes it easier to identify which are the best features. Next, the article uses LightGBM which is a type of lightweight boosting method to train datasets and enhance accuracy. The suggested Improved MayFly Optimization with LightGBM (IMFO-LGBM) method is tested on well-known smart city datasets that can be accessed from Kaggle site. The paper shows new ways to make features and train the dataset, but it should explain more about how experiments were done and results were understood. It could also talk more in-depth about what this method means for real-life situations and its possible limits. Oyewola, et al [29] compared the performance of five machine learning models that include Bagging (BAG), K-Nearest Neighbors (KNN), Multivariate Adaptive Regression Spline (MARS), Bayesian Generalized Linear Model (BGLM) and Generalized Linear Model (GLM). The goal is to predict traffic patterns in a smart city. Deekshitha, et al [30] intended to predict traffic for developing an intelligent transportation systems, where they use past data from previous years to predict what will happen with traffic this year. By studying the information every hour and looking at how close their predictions, it helps people understand real-time traffic states. The statistics that come up from these forecasts give useful ideas for drivers who want to take good choices while on a journey. Furthermore, the system performs comparative evaluations of every road so it can single out the most jam-packed paths within city limits. This characteristic gives extra usefulness to urban commuters. Even though the paper shows a hopeful way for traffic forecast, more information about prediction method, where data comes from and checking methods could make study more complete.

The examination of current literature about forecasting traffic congestion in intelligent city situations highlights some areas needing more study. The first one is that although traditional ways of machine learning and deep learning have been thoroughly investigated, there exists a shortage in comparative researches which systematically assess how well various algorithms perform within wide-ranging urban settings. Secondly, numerous studies concentrate on immediate traffic forecasts, leaving out the chance to predict long-term situations and strategic handling of traffic issues. In addition, the expansion ability along with transferring of forecast models across various cities and regions are somewhat unexamined aspects. This shows a lack of comprehension about how well suggested solutions can be applied elsewhere. Handling these research holes will not only push forward our knowledge on predicting traffic congestion but also help create stronger and efficient smart city transport systems.

3. Proposed Methodology

In this part, the clear explanation for the proposed Dynamic Traffic Prediction Network (DTPN) model is provided in detail. We will explain how it works and what methods it uses. This work's main goal is to present an original framework called DTPN that is designed for predicting traffic congestion in smart city networks. For reaching this aim, we use advanced image processing methods - MobileNet Recurrent Convolutional Network (MRCN), and also a new technique named Sonar Sweep Optimization (SoSO). The DTPN methodology is an approach to examining and foreseeing traffic jams in city areas, using input data that comes as pictures. These pictures are like quick views of different parts of the city's traffic situation, showing important details such as how many vehicles there are on a road or what kind of condition it's in along with patterns for how much movement is happening at any given time. The power of the MobileNet structure, which includes separable convolution layers based on depthwise convolutions assists DTPN in efficiently extracting spatial features from these images without excessive computational requirements. Next, the spatial features are taken and put into the recurrent layers of MRCN. This lets the model understand time-based relationships and moving patterns found in traffic data. The inclusion of recurrent connections helps DTPN to examine data sequences across time, thus boosting its prediction abilities. Moreover, the Sonar Sweep Optimization (SoSO) technique is used in this model for fine-tuning the model parameters to get high predictive precision. By combining these advanced methods together in harmony, DTPN provides a special and strong answer for predicting traffic jams within smart city networks. This prediction is very helpful for those who plan cities and manage traffic because it allows them to take action before congestion starts. They can put in place measures that lessen congestion or enhance transport efficiency when they know about it beforehand from reliable predictions made by DTPN. In the end, the creation of DTPN makes a big step forward in traffic management area which could change how we deal with difficult issues related to urban movement during current time period. The proposed DTPN is bringing a fresh framework that changes the way we predict traffic jams in smart city networks. It essentially uses advanced methods for processing images, especially MRCN, to get complex spatial traits from traffic pictures. In contrast with old ways of looking at images as fixed things, DTPN uses the recurrent connections in MRCN to understand times-based relationships and moving patterns present in traffic data. This merging creates the ability for DTPN to inspect sequential details across time, providing better understanding about how traffic moves and congestions growing.

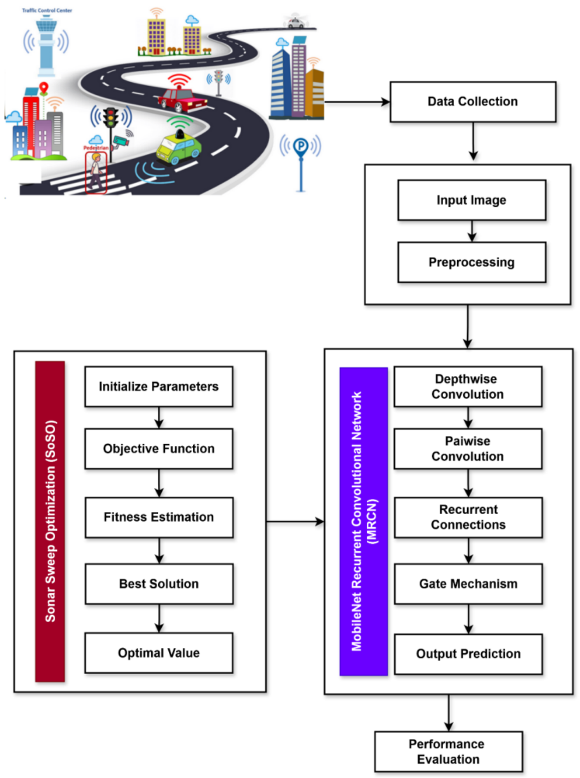


Fig 1. Flow of the DTCN Model

The flow of the DTCN model can be seen in Fig 1, which gives a graphical display of how the DTCN structure works to anticipate traffic congestion amounts within smart city networks.

- Acquiring Input Data: The first step is to acquire input data. This usually involves gathering different kinds of traffic-related information from smart city sensors, cameras or other monitoring devices. The input data contains images that show the present situation of traffic like vehicle quantity, condition of roads and patterns in how traffic is flowing.
- Preprocessing: The input data that we have collected goes through preprocessing steps to make it ready for more analysis. This can include actions like cleaning the data, normalizing it, and adding more to improve quality and uniformity of input data.
- Feature Extraction: The data that has been preprocessed is given to the module for feature extraction. Here, important spatial and temporal features are taken out. This part uses complex methods like convolutional neural networks (CNNs) or recurrent neural networks (RNNs) to get useful attributes from the input data.
- Training and validation: After feature extraction, the DTCN model is trained. In this phase, the model learns to match input data with appropriate levels of traffic congestion. It does so by using a mix of supervised learning methods and optimization techniques on the features gathered from previous stage. After the model is trained, it can be used to make predictions on fresh data that hasn't been seen before. The DTCN model which has been trained takes in input as the features extracted from incoming data and gives out predictions on traffic jam levels for a specific smart city network.
- Evaluation: At last, the performance of DTCN model is assessed by using suitable standard and yardstick to measure how precise, dependable and successful it can be in predicting traffic jam. This evaluation step helps in understanding the strong points as well as restrictions of this model and directs possible enhancements or adjustments.

3.1 MobileNet Recurrent Convolutional Network (MRCN) for Traffic Congestion Prediction

The MRCN, a design of deep neural network for traffic congestion prediction tasks, is developed in this work for smart city settings. The structure combines aspects from MobileNet –seen as small-sized CNNs or Convolutional Neural Networks and RNN parts used to understand time-related connections within traffic data. The choice of MobileNet as the base framework was made because it is adept at dealing with image data and can scale better, making it appropriate for application in situations where resources might be scarce. It uses depthwise separable convolutions. This way is beneficial in reducing the calculation expense of regular convolutions and allows for quicker inference on devices with lower computational power. MobileNet functions as the primary characteristic extraction model in traffic congestion prediction, taking input images that depict various road traffic scenes. In MobileNet architecture, we add recurrent connections to capture temporal dependencies and dynamic patterns. Then, we integrated it with recurrent neural network (RNN) layers, so that information can stay and move forward through time steps. This lets the model learn from sequence data - such as past traffic records - and make predictions considering how traffic jams change over time. In MRCN architecture, MobileNet part does feature extraction from given traffic images. It gets spatial details like vehicle amount, road state and traffic movement. These kinds of spatial features get into the recurrent layers where they do temporal modeling. The connections that are recurrent help the model to study sequential data about traffic across a span of time, learning about patterns and changes which impact how congested it is with vehicles there. When the MRCN model is trained, we can use it to make predictions about traffic congestion in smart city networks. After giving input traffic images, the model uses MobileNet backbone for processing these images. It then takes out spatial features and sends them into recurrent layers so as to model time. In the end, this gives an output that predicts how much traffic will be congested at one particular time and place within a smart city network. The MRCN model's performance is measured using prediction accuracy, precision, recall and F1-score. These metrics show how well the model can predict traffic congestion.

The MobileNet convolution operation is a depthwise separable convolution followed by pointwise convolution. This operation effectively gathers spatial characteristics from input feature maps. The depthwise separable convolution operation in MobileNet is shown as:

$$\mathrm{O_F} = \text{Pointwise Convolution}(\text{Depthwise Conv}(\mathrm{F_M})) \tag{1}$$

Where, $\mathrm{F_M}$ indicates the input image feature map, and $\mathrm{O_F}$ is the output feature map. Recurrent connections, which are made by using Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks, can catch time-related relationships in traffic data. For every step in time, the hidden state of the recurrent layer gets an update depending on output feature map from MobileNet and previous hidden state. The computation for hidden state at time step with RNN operation is mathematically represented in below:

$$\mathfrak{h}_k = \mathrm{RNN}(\mathfrak{h}_{k-1}, \mathrm{O_F}) \tag{2}$$

Where, $\mathfrak{h}_{k-1}$ represents the hidden state of previous time step k, and $\mathfrak{h}_k$ is the hidden state information. Mechanisms that act like gates, for instance input, forget and output gates in LSTM networks, direct the information flow within recurrent layer. They handle how cell state gets updated and calculation of output at every time step by deciding what information model keeps or discards. The equations for the input gate, forget gate and output gate are given below:

$$\mathscr{i}_k = \varphi(\omega_{\mathscr{i}x}\mathrm{F_{Mk}} + \omega_{\mathscr{i}\mathfrak{h}}\mathfrak{h}_{k-1} + \beta_{\mathscr{i}}) \tag{3}$$

$$\mathscr{f}_k = \varphi(\omega_{\mathscr{f}x}\mathrm{F_{Mk}} + \omega_{\mathscr{f}\mathfrak{h}}\mathfrak{h}_{k-1} + \beta_{\mathscr{f}}) \tag{4}$$

$$\mathscr{o}_k = \varphi(\omega_{\mathscr{o}x}\mathrm{F_{Mk}} + \omega_{\mathscr{o}\mathfrak{h}}\mathfrak{h}_{k-1} + \beta_{\mathscr{o}}) \tag{5}$$

Where, φ indicates the sigmoid activation function, which is optimally computed with the use of SoSO algorithm, $\omega_{\mathscr{i}x}$, $\omega_{\mathscr{f}x}$, $\omega_{\mathscr{o}x}$ are the input weight matrices, $\omega_{\mathscr{i}\mathfrak{h}}\mathfrak{h}_{k-1}$, $\omega_{\mathscr{f}\mathfrak{h}}\mathfrak{h}_{k-1}$, $\omega_{\mathscr{o}\mathfrak{h}}\mathfrak{h}_{k-1}$ the recurrent weight matrices, and $\beta_{\mathscr{i}}$, $\beta_{\mathscr{f}}$, $\beta_{\mathscr{o}}$ are the bias values. For LSTM networks, the cell state is updated with input gate, forget gate and candidate cell state. Element-wise multiplication is used to combine the influence of forget gate on previous cell state with input gate's effect from candidate cell state for updating current one. In LSTM networks, we update the cell state using the input gate, forget gate and current input:

$$\tilde{\mathcal{C}}_k = \tanh(\omega_{\mathcal{C}x}\mathrm{F_{Mk}} + \omega_{\mathcal{C}\mathfrak{h}}\mathfrak{h}_{k-1} + \beta_{\mathscr{o}}) \tag{6}$$

$$\mathcal{C} = \mathscr{f}_k \odot \mathcal{C}_{k-1} + \mathscr{i}_k \odot \tilde{\mathcal{C}}_k \tag{7}$$

Where, $\tilde{\mathcal{C}}_k$ represents the candidate cell state and $\odot$ indicates the element wise operation. The last output in every step of time is calculated using the new cell state and the output gate. This gate controls how much information gets sent to the output, it's found as a point-wise multiplication of value from output gate with hyperbolic tangent of updated cell state. The MRCN model, with the merging of these functions, is able to understand spatial characteristics from traffic pictures through MobileNet and temporal relationships in traffic information using recurrent connections. This makes it perfect for predicting congestions accurately within smart city surroundings.

3.2 Sonar Sweep Optimization (SoSO) for Parameter Tuning

The Sonar Sweep Optimization (SoSO) is a new optimization method that draws from the concept of sonar. The aim of this technique is to fine-tune the hyperparameters in machine learning models for getting improved results. In our study on forecasting traffic congestion, we used SoSO to optimize parameters in MobileNet Recurrent Convolutional Network (MRCN) model. Approximation and alteration of the parameters for sigmoid activation function constitute a significant aspect in this optimization.
The sigmoid activation function is widely used in neural networks, especially for models dealing with outputs similar to probability. It is computed as follows:

$$\varphi(x) = \frac{1}{1+e^{-x}} \tag{8}$$

Where, $\varphi(x)$ indicates the sigmoid activation function for input x. For the SoSO algorithm, when starting, the number of ships (agents) is set beforehand. This method is used often in algorithms inspired by nature to control computational resources efficiently. Normally, a larger quantity of agents enhances chances for locating an optimized solution as it permits broader investigation throughout the solution area. However, SoSO has an interesting characteristic that lessens the requirement for a lot of agents by increasing the exploration skills of every agent. When we begin the algorithm, we place these agents in different positions inside solution space. The starting process is very important because it determines initial search locations. The simplest and most frequent way to start these positions is with a random procedure, typically employing a normal distribution function. This randomness guarantees that the agents commence at different points in the solution area, thereby encouraging varied exploration right from initiation.

In the start of algorithm, we randomly set up the positions for agents within pre-determined boundaries of solution space. This step makes sure there's a variety in beginning point for searching process and covers wide range area within solution space. Every agent also gets given an effective radius and intensity. The area in which the agent can look for solutions is determined by the effective radius, and its ability to find best solutions within this area is represented by intensity. These two parameters are very important because they give direction to how agents search. After setting these values, we start main loop of algorithm that repeatedly improves positions of all our agents. The loop goes on until a stop condition is met, like reaching the maximum number of iterations or getting to a good enough fitness level. The algorithm examines if a counter has arrived at an earlier set checkpoint. If this situation is true, then the agent will be moved to another position. This movement assists with discovering fresh regions within solution space and halts search process from becoming trapped in local optima. The effective radius for every agent is now changed as per the state of search. This change allows agents to either increase or decrease their search area, depending on the circumstances. So, the algorithm maintains a balance between exploring and utilizing solution space. The recalculating of intensity for each agent is similar to determining how successful they have been in finding better solutions. Those who find promising solutions can increase their intensity, making them better at detecting even more improved solutions during next iterations.

4. Results and Discussion

This section shows that the suggested model, the DTCN, has been validated and is considered strong. Validation is an important phase in creating any forecasting model. It makes sure the model works well on data it hasn't seen before, not just on the training set. We are using open source datasets, which provide traffic information from different smart city locations and cover a wide range of traffic scenarios. The chosen open source datasets for validation, Kaggle and GoogleMap, are quite varied in their traffic scenarios. They include city traffic congestion, movement on highways and even seasonal patterns within traffic conditions. These datasets provide extensive and data to test the model's ability to handle different kinds of traffic situations. Once these sets have been chosen, we do preprocessing steps before they go into our model. Standard preprocessing steps involve cleaning the data, normalizing it or feature extraction as required by our input pipeline setup stage. This aids in ensuring that the data is well-structured which enhances learning of model by assisting it to comprehend and utilize information efficiently. To understand more clearly how well the DTCN model performs, we use a variety of evaluation measures. These measures give us number-based results that can be used for comparison with other methods available now.

Estimation of accuracy is the ratio of correctly predicted instances to the total instances. A higher value in this metric signifies better performance of model. Precision is a calculation that uses the total number of correct positive predictions made by the model and divides it by all positive predictions. Recall (Sensitivity) estimates the ability of our model to find and recognize all correct instances in the dataset. The F1 Score, which is a mix of precision and recall into one value, gives a single measure for comparing models. The Mean Absolute Error (MAE) is calculated using the average absolute difference between the predicted and actual values. The performance measures are estimated as shown in below:

$$\text{Accuracy} = \frac{\text{No of correct predictions}}{\text{Total no of predictions}} \quad (9)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (10)$$

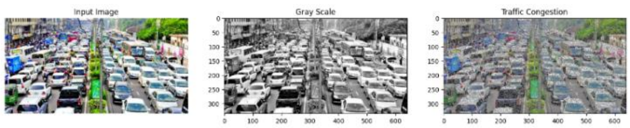
$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (11)$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}(g_i - \hat{g}_i)^2 \quad (13)$$

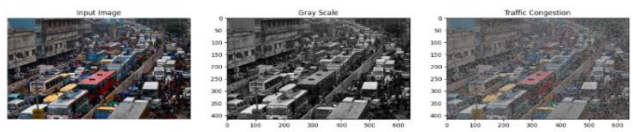
$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(g_i - \hat{g}_i)^2} \quad (14)$$

Where, TP – true positives, TN – true negatives, FP – false positives, FN – false negatives, N indicates the total number of observations, g_i denotes the actual value, and $\hat{g}_i$ represents the predicted value. Fig 2 shows the steps included in handling an input image through DTPN model. Three important images are shown in this figure: original input image, grayscale version of input image and predicted output image for traffic congestion. The progression of these visual elements provides a clear understanding about how data is processed and analyzed by this model to make predictions on traffic congestion.

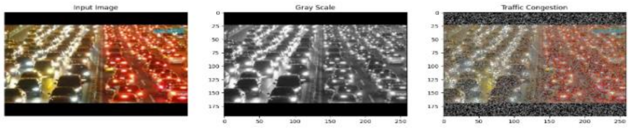


(a)

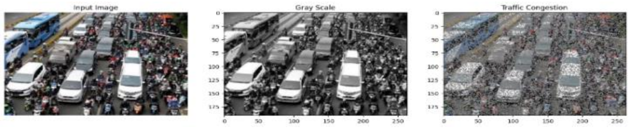
(b)



(c)

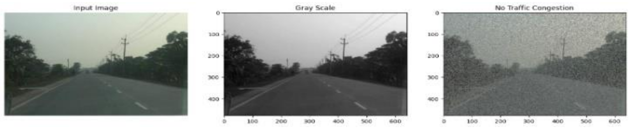


(d)



(e)

(f)



(g)

Fig 2. Sample input, grayscale and traffic congestion predicted output images

The input image, as the name suggests, is where we begin with for estimating traffic congestion. It contains all necessary details that model will examine to identify and forecast levels of congestion. The reason behind converting it into gray scale is to simplify the image by reducing its data complexity. This part is very important because it lessens the computational burden and concentrates the analysis on changes in intensity instead of color details. Usually, colors are not as useful for identifying traffic congestion. The anticipated image output shows visually where the traffic jam areas are; they might be marked with separate colors or shades like gray (for instance). This gives a visual understanding of road conditions quickly, which helps in making decisions related to managing traffic.

Fig 3 and Fig 4, together with Table 1 and Table 2, give a comparative study of different traffic congestion prediction models using the metrics: precision, recall, F1-score and accuracy on a Kaggle dataset. In Fig 3 it is shown how well models perform in terms of precision (vertical axis), recall (horizontal axis) as well as their F1-scores (size/color). The suggested Dynamic Traffic Prediction Network model shows much better results compared to traditional methods like Random Forests (RF), XGBoosts (XGB), LightGBMs (LGBM) and Improved May Fly Optimization with Light GBMs (IMFO-LGBM). The exact values for these metrics are provided in Table 1 which displays that the suggested model achieves almost perfect scores (Precision 99%, Recall 98.9%, F1-Score 99%). This underlines its superior ability to accurately predict traffic congestion. In Fig 4, we can see a comparison of the models according to accuracy. It confirms that DTPN model is very powerful because it has topmost accuracy among all models. This thorough comparison shows how strong and effective the DTPN model is for handling and forecasting traffic congestion in smart city settings, making it a hopeful solution for practical use.

Table 1. Precision, recall and f1-score values for existing and proposed models

Methods	Precision	Recall	F1-score
RF	81	80	81
XGB	82	81	82.1
LGBM	85	82.5	84.3
IMFO-LGBM	88	88.2	88.2
Proposed	99	98.9	99

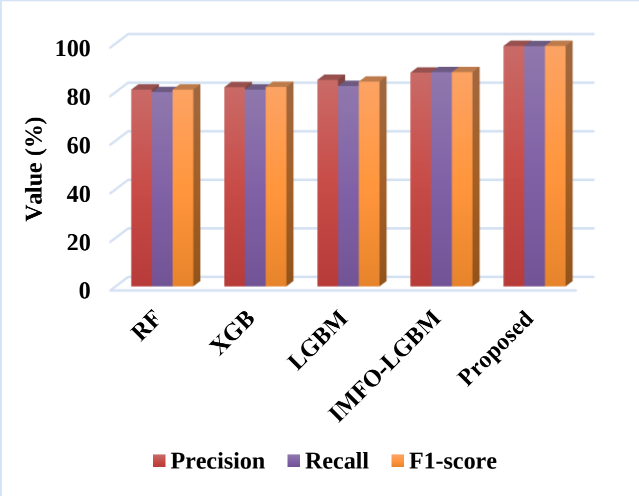


Fig 3. Comparison based on precision, recall and f1-score with existing models using Kaggle dataset

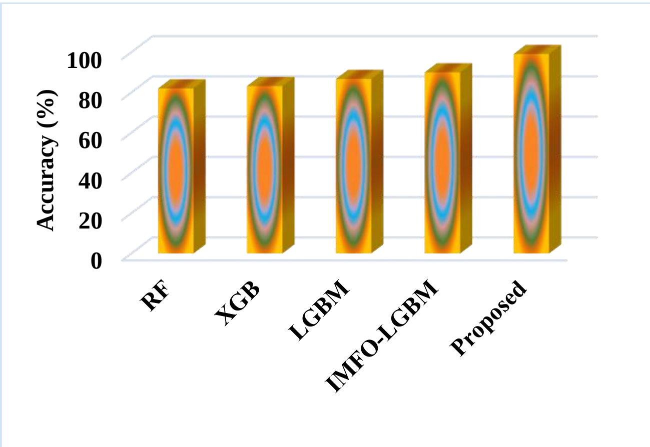


Fig 4. Comparison based on accuracy with existing models using Kaggle dataset

Table 2. Comparison based on accuracy

Methods	Accuracy (%)
RF	82
XGB	83.2
LGBM	86.7
IMFO-LGBM	90
Proposed	99.1

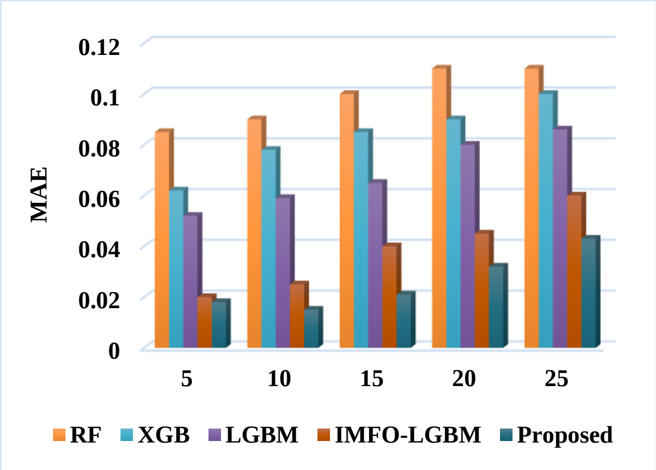


Fig 5. Comparison based on MAE with existing models using Kaggle dataset

Table 3. Comparison based on MAE

Time scale	RF	XGB	LGBM	IMFO-LGBM	Proposed
5	0.085	0.062	0.052	0.02	0.018
10	0.09	0.078	0.059	0.025	0.015
15	0.1	0.085	0.065	0.04	0.021
20	0.11	0.09	0.08	0.045	0.032
25	0.11	0.1	0.086	0.06	0.043

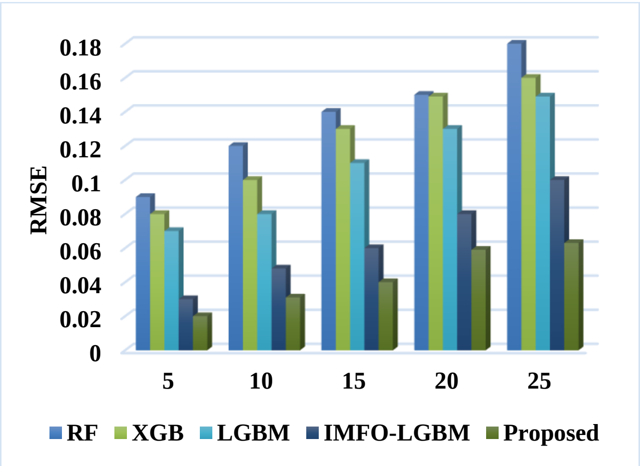


Figure 6. Comparison based on RMSE with existing models using Kaggle dataset

The MAE values for various models over time periods of 5, 10, 15, 20 and 25 minutes are shown in Figure 5 and Table 3. Mean Absolute Error (MAE) is a measure used to understand prediction accuracy: it tells us how big mistakes on average are in a set of predictions without considering which way they miss the mark. When the MAE value is smaller, this indicates better model performance. The DTPN model is showing the least values of MAE on all scales of time, which means it has better accuracy in forecasting traffic congestion. In particular, when we consider a 5-minute time scale, this proposed method gives an MAE value as low as 0.018. This result is exceptionally good when compared to other models like Random Forest (RF), XGBoost (XGB), LightGBM (LGBM) and Improved MayFly Optimization with LightGBM (IMFO-LGBM). When the time scale increases to 25 minutes, the DTPN model still performs best with an MAE of only 0.043 - a significant improvement over RF's 0.11, XGB's 0.1, LGBM's 0.086 and IMFO-LGBM's value of just at about .06 for this larger timeframe bulkier prediction task. This steady performance in various time scales shows the DTPN model is strong enough for handling predictions of traffic in both short-term and long-term periods.

In the same way, Figure 6 and Table 4 show the RMSE values for matching models and time spans. The RMSE is about measuring how much variation there is between predicted and observed values, with a stronger reaction to bigger errors because of its quadratic character. Smaller RMSE numbers signify forecasts that are more precise and have less big errors. The DTPN model we suggest gives the smallest RMSE values for all time scales, showing its efficiency in making accurate predictions about traffic congestion. At a 5-minute time scale, this proposed model gets an RMSE of 0.02 that is much lower than the RF (0.09), XGB (0.08), LGBM (0.07) and IMFO-LGBM (0.03). Even on a bigger 25-minute time scale, DTPN still has the lowest RMSE of 0.063 compared to other models: RF (0.18), XGB(0.16), LGBM(149) and IMFO-LGBM(1). This similarity in lower RMSE values throughout different time frames shows how the DTPN model is good at reducing big prediction mistakes. This makes it dependable for predicting traffic jams in real time within smart city networks.

Table 4. Comparison based RMSE

Time scale	RF	XGB	LGBM	IMFO-LGBM	Proposed
5	0.09	0.08	0.07	0.03	0.02
10	0.12	0.1	0.08	0.048	0.031
15	0.14	0.13	0.11	0.06	0.04
20	0.15	0.149	0.13	0.08	0.059
25	0.18	0.16	0.149	0.1	0.063

The DTPN model demonstrates strong potential to alter the functioning of traffic control in smart cities, displaying the lowest MAE and RMSE values over all time scales examined. Its accuracy and reliability ensure that traffic authorities are able to make informed decisions for reducing congestion, improving flow of traffic and enhancing urban mobility. By outperforming other conventional or recent models, the DTPN model establishes a fresh benchmark in traffic prediction technology. This reveals its potential to bring significant progress in the area of city transport control, guiding towards smarter methods for handling urban traffic that promise better overall results. To sum up, the detailed comparisons with MAE and RMSE show that DTPN model can predict traffic congestion very accurately and with low error. This is a significant improvement compared to current methods. The results confirm how suitable the suggested model is for use in smart city settings. Here, precise and timely traffic forecasts are crucial to enhance transport effectiveness and reduce issues triggered by high levels of congestion.

5. Conclusion

This paper proposes a new framework, DTCN to forecast traffic congestion in smart city surroundings. The MRCN and SoSO techniques provide strong and effective solutions for managing traffic instantly. We have presented the DTPN model, which is a new and innovative framework designed to predict traffic congestion in smart city environments. The advanced methodologies used by this model include the MRCN for feature extraction and temporal modeling as well as SoSO technique that helps with hyperparameter tuning. This high-level technology solution shows promise in providing efficient management of traffic. The MRCN has excellent ability to extract features of space and time from traffic images, making it suitable for prediction purposes. Simultaneously, the SoSO is beneficial because it enhances the effectiveness of this model. It does so by optimizing its parameters using a method inspired by nature that ensures equilibrium between exploration and exploitation in searching for solutions. The DTPN model shows excellent results in our thorough evaluations with open-source datasets, especially from Kaggle. The superiority of the proposed model is evident across multiple metrics like precision, recall, F1-score, MAE and RMSE. It consistently outperforms other approaches such as Random Forest (RF), XGBoost (XGB), LightGBM (LGBM) and Improved MayFly Optimization with LightGBM (IMFO-LGBM). The findings of the study indicate that the DTPN model is not only effective in predicting traffic congestion, but it can also be adjusted and scaled to fit various urban situations.

The application of this model might result in significant improvements for managing traffic movement. It could potentially reduce congestion along with travel time and related environmental impacts. The DTPN model is a big advancement in intelligent transportation systems. It utilizes high-level techniques of machine learning and optimization algorithms, providing city planners as well as traffic controlling bodies with a powerful tool to enhance urban mobility. The future tasks will focus on refining this model more, incorporating real-time data flows and expanding its utilization within smart city frameworks to other domains. The model hopes to provide a smarter, more efficient and enduring solution for managing traffic in cities using DTPN deployment. Ultimately, this could result in improved global living standards.

Declaration Statement

Ethical Statement

I will conduct myself with integrity, fidelity, and honesty. I will openly take responsibility for my actions, and only make agreements, which I intend to keep. I will not intentionally engage in or participate in any form of malicious harm to another person or animal.

Informed Consent for data Used

All subjects gave their informed consent for inclusion before they participated in the study. The study was conducted in accordance with the Declaration of Helsinki.

I consent to participate in the research project and the following has been explained to me: the research may not be of direct benefit to me. my participation is completely voluntary. my right to withdraw from the study at any time without any implications to me.

Data Availability

- Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.
- The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.
- All data generated or analysed during this study are included in this published article

Conflict of Interest

The authors declare that they have no conflict of interest.

Competing Interests

The authors have no competing interests to declare that are relevant to the content of this article.

Funding Details

No funding was received to assist with the preparation of this manuscript.

Acknowledgments

I am grateful to all of those with whom I have had the pleasure to work during this and other related Research Work. Each of the members of my Dissertation Committee has provided me extensive personal and professional guidance and taught me a great deal about both scientific research and life in general.

References

[1] B. Vijayalakshmi, K. Ramar, N. Jhanjhi, S. Verma, M. Kaliappan, K. Vijayalakshmi, et al., "An attention-based deep learning model for traffic flow prediction using spatiotemporal features towards sustainable smart city," International Journal of Communication Systems, vol. 34, p. e4609, 2021.

[2] A. Hameed, J. Violos, and A. Leivadeas, "A deep learning approach for IoT traffic multi-classification in a smart-city scenario," IEEe Access, vol. 10, pp. 21193-21210, 2022.

[3] F. Wang, J. Xu, C. Liu, R. Zhou, and P. Zhao, "On prediction of traffic flows in smart cities: a multitask deep learning based approach," World Wide Web, vol. 24, pp. 805-823, 2021.

[4] S. Bhattacharya, S. R. K. Somayaji, T. R. Gadekallu, M. Alazab, and P. K. R. Maddikunta, "A review on deep learning for future smart cities," Internet Technology Letters, vol. 5, p. e187, 2022.

[5] Q. Chen, W. Wang, F. Wu, S. De, R. Wang, B. Zhang, et al., "A survey on an emerging area: Deep learning for smart city data," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 3, pp. 392-410, 2019.

[6] B. Liu, C.-T. Lam, B. K. Ng, X. Yuan, and S. K. Im, "A Graph-based Framework for Traffic Forecasting and Congestion Detection using Online Images from Multiple Cameras," IEEE Access, 2024.

[7] S. Neelakandan, M. Berlin, S. Tripathi, V. B. Devi, I. Bhardwaj, and N. Arulkumar, "IoT-based traffic prediction and traffic signal control system for smart city," Soft Computing, vol. 25, pp. 12241-12248, 2021.

[8] S. Khan, S. Nazir, I. García-Magariño, and A. Hussain, "Deep learning-based urban big data fusion in smart cities: Towards traffic monitoring and flow-preserving fusion," Computers & Electrical Engineering, vol. 89, p. 106906, 2021.

[9] R. M. AlZoman and M. J. Alenazi, "A comparative study of traffic classification techniques for smart city networks," Sensors, vol. 21, p. 4677, 2021.

[10] D. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik, and P. K. Singh, "Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities," Transactions on Emerging Telecommunications Technologies, vol. 32, p. e4121, 2021.

[11] A. N. Muhammad, A. M. Aseere, H. Chiroma, H. Shah, A. Y. Gital, and I. A. T. Hashem, "Deep learning application in smart cities: recent development, taxonomy, challenges and research prospects," Neural computing and applications, vol. 33, pp. 2973-3009, 2021.

[12] S. Majumdar, M. M. Subhani, B. Roullier, A. Anjum, and R. Zhu, "Congestion prediction for smart sustainable cities using IoT and machine learning approaches," Sustainable Cities and Society, vol. 64, p. 102500, 2021.

[13] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," IEEE Transactions on Intelligent Transportation Systems, vol. 23, pp. 4927-4943, 2021.

[14] G. Perumal, G. Subburayalu, Q. Abbas, S. M. Naqi, and I. Qureshi, "VBQ-Net: A Novel Vectorization-Based Boost Quantized Network Model for Maximizing the Security Level of IoT System to Prevent Intrusions," Systems, vol. 11, p. 436, 2023.

[15] A. Navarro-Espinoza, O. R. López-Bonilla, E. E. García-Guerrero, E. Tlelo-Cuautle, D. López-Mancilla, C. Hernández-Mejía, et al., "Traffic flow prediction for smart traffic lights using machine learning algorithms," Technologies, vol. 10, p. 5, 2022.

[16] J. James, "Citywide traffic speed prediction: A geometric deep learning approach," Knowledge-Based Systems, vol. 212, p. 106592, 2021.

[17] A. N. Aledaily and K. Yadav, "Deep learning techniques for the prediction of traffic jam management for smart city infrastructure," in Artificial Intelligence and Blockchain in Industry 4.0, ed: CRC Press, 2024, pp. 163-183.

[18] K. L.-M. Ang, J. K. P. Seng, E. Ngharamike, and G. K. Ijemaru, "Emerging technologies for smart cities' transportation: geo-information, data analytics and machine learning approaches," ISPRS International Journal of Geo-Information, vol. 11, p. 85, 2022.

[19] X. Li, H. Liu, W. Wang, Y. Zheng, H. Lv, and Z. Lv, "Big data analysis of the internet of things in the digital twins of smart city based on deep learning," Future Generation Computer Systems, vol. 128, pp. 167-177, 2022.

[20] D. Hemanand, G. V. Reddy, S. S. Babu, K. R. Balmuri, T. Chitra, and S. Gopalakrishnan, "An intelligent intrusion detection and classification system using CSGO-LSVM model for wireless sensor networks (WSNs)," International Journal of Intelligent Systems and Applications in Engineering, vol. 10, pp. 285–293-285–293, 2022.

[21] M. Bai, Y. Lin, M. Ma, P. Wang, and L. Duan, "PrePCT: Traffic congestion prediction in smart cities with relative position congestion tensor," Neurocomputing, vol. 444, pp. 147-157, 2021/07/15/ 2021.

[22] M. Akhtar and S. Moridpour, "A review of traffic congestion prediction using artificial intelligence," Journal of Advanced Transportation, vol. 2021, p. 8878011, 2021.

[23] N. U. Khan, M. A. Shah, C. Maple, E. Ahmed, and N. Asghar, "Traffic flow prediction: an intelligent scheme for forecasting traffic flow using air pollution data in smart cities with bagging ensemble," Sustainability, vol. 14, p. 4164, 2022.

[24] T. Devi, K. Alice, and N. Deepa, "Traffic management in smart cities using support vector machine for predicting the accuracy during peak traffic conditions," Materials Today: Proceedings, vol. 62, pp. 4980-4984, 2022.

[25] A. Campbell, A. Both, and Q. C. Sun, "Detecting and mapping traffic signs from Google Street View images using deep learning and GIS," Computers, Environment and Urban Systems, vol. 77, p. 101350, 2019.

[26] R. Al-qudah, Y. Khamayseh, M. Aldwairi, and S. Khan, "The Smart in Smart Cities: A Framework for Image Classification Using Deep Learning," Sensors, vol. 22, p. 4390, 2022.

[27] A. G. Ismaeel, K. Janardhanan, M. Sankar, Y. Natarajan, S. N. Mahmood, S. Alani, et al., "Traffic pattern classification in smart cities using deep recurrent neural network," Sustainability, vol. 15, p. 14522, 2023.

[28] J. Jenifer and R. Priyadarsini, "Improved mayfly optimization and LightGBM classifier for smart city traffic prediction," Indian Journal of Science and Technology, vol. 15, pp. 2085-92, 2022.

[29] D. O. Oyewola, E. G. Dada, and M. B. Jibrin, "Smart City Traffic Patterns Prediction Using Machine Learning," in Machine Learning Techniques for Smart City Applications: Trends and Solutions, ed: Springer, 2022, pp. 123-133.

[30] H. Deekshetha, A. Shreyas Madhav, and A. K. Tyagi, "Traffic prediction using machine learning," in Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2021, ed: Springer, 2022, pp. 969-983.