IJMRSETM

e-ISSN: 2395 - 7639

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING, TECHNOLOGY AND MANAGEMENT

Volume 10, Issue 10, October 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.580

+91 99405 72462 +9163819 07438 ijmrsetm@gmail.com www.ijmrsetm.com

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

Employing Deep Learning for Real-Time Data Collection and Decision-Making for Traffic Management in Smart Cities

Faraj Muftah Mohamed Akash

Faculty of Civil Aviation and Meteorology, Asbeah, Libya, Department of roads and landing.

ABSTRACT: To reduce traffic, improve road safety, and optimize transportation networks, communities that are rapidly urbanizing need effective and sophisticated traffic management systems. A thorough deep learning methodology designed for real-time traffic control in smart cities is presented in this research. The approach uses Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for data collecting, pre-processing, feature extraction, and decision-making. The primary contributions of this study are its comprehensive strategy for handling the geographical and temporal components of traffic data, which eventually improves traffic flow and decision-making. Real-time traffic camera images are acquired throughout the data gathering phase. Standardization and min-max normalization are two pre-processing approaches which ensure data compatibility and homogeneity across various image features. In the key phase of feature extraction, CNNs are used to recognize distinguishing visual characteristics. CNNs are renowned for their proficiency in successfully capturing spatial representations, which is essential for comprehending traffic situations. Shift invariance and local feature extraction are accomplished using the CNN architecture, which consists of convolution and pooling layers. The application of RNNs, which are excellent at handling sequential data, is used to handle the temporal elements of traffic data. RNNs use data from earlier time steps to simulate traffic patterns, taking into consideration the constantly changing nature of traffic circumstances. This makes it possible to make decisions in real time based on past traffic data, improving traffic management. The implementation of the proposed method is done using python software. The accuracy (99.2%), precision, recall, and F1-score of the proposed CNN-RNN hybrid model are greater than those of existing techniques like CNN, NN, ENN and ANFIS, demonstrating its better performance. This study provides a comprehensive and practical approach to improving traffic management in smart cities, which will ultimately lead to less traffic congestion, more traffic safety, and better transit networks.

KEYWORDS: Traffic Management, Smart Cities, Convolutional Neural Networks, Recurrent Neural Networks, Transportation Networks

I. INTRODUCTION

In order to handle the problems brought on by rising urbanization and population expansion, traffic management is a crucial aspect of smart cities. Congestion reduction is one of the main goals of traffic management in smart cities. Traffic congestion constitutes a serious problem when metropolitan areas grow more densely inhabited, which results in lost time, elevated anxiety, and lower productivity for locals [1]. By maximizing traffic flow and minimizing bottlenecks, effective traffic management techniques, such as sophisticated traffic signal systems and continuous traffic monitoring, assist reduce congestion and eventually increase the mobility of a city's residents [2]. Pollution reduction in smart cities is greatly aided by effective traffic management. Traffic congestion not only aggravates drivers but also deteriorates the quality of the air. Elevated levels of dangerous pollutants are the outcome of stationary cars and numerous stoppings and starting. Smart cities may lower air pollution and provide a cleaner and more environmentally friendly urban environment by deploying intelligent traffic management technologies, such as traffic light synchronization and traffic redirecting according to real-time information. Another important justification for traffic control in smart cities is energy conservation [3]. In heavy traffic, continuous braking and acceleration waste energy and consume a lot of petrol. By enhancing traffic flow, lowering vehicle energy usage generally, and lowering the environmental impact of urban transportation, smart traffic management technologies can reduce these shortcomings to a minimum.

In smart cities, traffic management increases safety. A less hazardous driving environment is made possible by innovative technologies which includes traffic monitoring recording devices, collision detection technologies, and automated evacuations in emergencies. Traffic management aims to lower the number of fatalities and serious injuries, keeping city streets easier for all users by swiftly detecting and reacting to accidents and emergencies. Incorporation of public transportation as an environmentally friendly method of mobility is another need for smart cities [4]. Efficient traffic management techniques can increase the accessibility and dependability of public transportation. Smart cities encourage inhabitants to utilize public transit through the installation of designated routes and infrastructures for trolleys, buses, trams, and railroads which lowers the number of private automobiles on roadways and promotes sustainable mobility alternatives. Decision-making based on information is crucial to traffic management. Information regarding traffic patterns and congestion is gathered via camera footage, sensors, and other sources of information [5]. Planners for cities can make intelligent decisions regarding traffic flow, road maintaining, and improvements to infrastructure through the examination of this information. By optimizing transportation networks, cities may increase their effectiveness and responsiveness to the requirements of their citizens. Smart cities must include traffic management since it can solve many urgent urban problems. Smart cities can greatly enhance the quality of life for their citizens while promoting economic development and environmental sustainability. They can do this by reducing traffic, reducing pollution, enhancing energy utilization, improving security, supporting environmentally friendly transportation, and using data-driven approaches.

An effective respond to the numerous issues with urban mobility is to use deep learning for traffic control in smart cities. Deep learning, a branch of artificial intelligence, is ideally suited for this task since it can analyze enormous amounts of information and identify complex traffic patterns. Traffic forecasting and optimization is a common use. Deep learning models can estimate traffic trends by examining historical traffic information, real-time information from sensors, and even social media data. This enables preventive changes to traffic lights and movement to reduce congestion and improve overall traffic effectiveness. Intelligent traffic signal creation is made easier by deep learning [6]. In order to relieve congestion and shorten commuter wait times, these adaptive signals can prioritize particular lanes or directions in real-time in response to dynamic traffic circumstances. Deep learning algorithms may also be used for managing and detecting accidents. These algorithms can quickly locate accidents and events through the evaluation of images captured from traffic cameras, resulting in immediate warnings to responding agencies and traffic rerouting to reduce interruptions and improve road safety. Overall, deep learning enables smart cities to transform traffic management by enhancing its effectiveness, responsiveness, and resident safety [7].

The foundation of efficient traffic management in smart cities is real-time data collecting and decision-making, transforming how metropolitan areas address the issues of contemporary mobility. The installation of a wide range of sensors, including as traffic cameras, automobile detection devices, and environment monitoring, which continually collect real-time information on many elements of traffic and their surrounding circumstances, is a significant component of this method [8]. In order to make well-informed decisions, these sensors build a complete and dynamic representation of the flow of traffic, congestion and weather conditions, and other pertinent aspects. One essential component of this procedure is the combination of real-time information from many sources. A centralized system that integrates data from GPS devices, mobile apps, and intelligent traffic signals enables traffic officials to acquire a comprehensive view of the city's present traffic condition. This combined information provides the foundation for quickly recognizing problems and swiftly resolving them to improve traffic flow [9]. In real-time traffic management, sophisticated analytics often powered by machine learning and artificial intelligence are essential. These analytical algorithms analyse the flood of information to identify trends in traffic flow, forecast areas of high congestion, and suggest the best routes to commuters as needed. As a consequence, decision-making by authorities may be quick and data-driven, improving travel times, traffic effectiveness, and resident dissatisfaction from congestion.

The dynamic regulation of traffic lights is one of the most obvious advantages of real-time information collecting and decision-making in smart cities. Actual time information is used by these intelligent traffic lights to adjust timing and synchronization. These signals may dynamically change to give preference to the route with the most traffic when congestion is identified, significantly cutting wait times and enhancing traffic flow. Real-time information is essential to systems for responding to emergencies and also helps to improve regular traffic conditions [10]. When accidents or events happen, the system can instantly organize closures of roads, immediately alert emergency services, and reroute traffic to reduce interruptions and improve security for everyone using the roads [11]. Digital displays and smartphone applications may give travellers current updates on traffic circumstances, closures of roads, and other routes thanks to real-time information, facilitating more informed choices and lowering irritation during trips. Real-time information affects the planning of transportation in smart cities in ways which extend outside the immediate. This information is used to inform long-term planning initiatives, helping decision-makers choose infrastructure upgrades, public transportation system expansions, and urban planning approaches that can support expanding populations while ensuring sustainability and effective traffic flow. RNNs can help determine the most effective routes for moving cars. This may be used to optimize public transit timetables, develop dynamic routes for individual drivers, or even manage traffic in actual time by redirecting vehicles to fewer crowded roadways [12]. RNNs can forecast upcoming transportation needs by examining trends and previous information. Development for public transportation benefits from this knowledge because it enables resource allocation that is effective and lessens traffic congestion. RNNs are additionally permitted to instantly adjust traffic signal timings according to the flow of traffic. RNNs aid in minimizing traffic congestion and enhancing traffic flow by taking into account the movement of cars and adjusting signal timings correspondingly [13].

A complex strategy is required when using deep learning for actual-time information collecting and making decisions in traffic management inside smart cities. Convolutional neural networks are employed at the beginning of the procedure for feature extraction, with a primary concentration on processing visual input from resources like traffic cameras and drone footage. CNNs are used for recognizing objects, identify moving objects, and determine important road features including lane markers and traffic signs. In the first step of the process, useful data may be extracted from the visual components laying the foundation for further research. Recurrent Neural Networks, which are experts at processing sequential data, are used. RNNs are skilled at digesting past traffic data, taking into account elements such as patterns of traffic, vehicle directions, and external influences like weather. They make route optimization, real-time traffic prediction, and adaptive traffic light management possible. This temporal evaluation enables proactive congestion management by enabling traffic management platforms to anticipate changing traffic circumstances and make educated decisions based on them. These two structures of neural networks are used to provide a thorough traffic control framework in smart cities. RNNs use the characteristics retrieved from the visual data analyzed by CNNs as input for additional evaluation and predictive modelling. The system can respond to both visual indications acquired by cameras and temporal traffic patterns seen over time thanks to this integrated methodology. The final outcome is a dynamic and adaptable traffic management system that, by utilizing deep learning in actual-time information collecting and decision-making, may improve traffic flow, increase road safety, and guarantee effective urban mobility. CNNs extraction of characteristics processes visual input from resources like surveillance footage and satellite imaging, and this processing is crucial for traffic management. The superior object recognition, detection of vehicles, and roadway marking and indicator recognition capabilities of CNNs allow for real-time surveillance and identification of anomalies. They provide traffic management systems the ability to evaluate traffic flow, spot delays, and quickly address emergencies. RNNs for traffic management, on the contrary hand, utilize the effectiveness of sequential processing of information to forecast traffic patterns, optimize directions, and enhance traffic signalling. RNNs take into account past traffic information, vehicle directions, and outside variables like the weather, enabling preventive actions and effective resource allocation. A comprehensive solution is provided by the combination of CNNs and RNNs, in which visual input is pre-processed by CNNs, and the derived characteristics are subsequently input into RNNs for predictive evaluation and real-time decision-making. By responding to temporal traffic patterns and visual signals, this combination improves the flow of traffic and security while also enhancing urban mobility in smart cities.

The Key Contribution of the paper is given as follows:

- By compiling a sizable collection of real-time traffic camera images taken at four signal points at significant intersections in Vijayawada and Guntur, capturing information on vehicle flow. The sizeable collection, weighing in at 1.6 GB, consists of low-resolution images shot in a variety of lighting, perspectives, and environments. Given the wide diversity of image properties, the research underlines the need of methods of pre-processing such standardization and min-max normalization to assure data compatibility and consistency.

- The research uses CNNs to extract features from the images captured by traffic cameras. The study describes how CNNs may be used to successfully capture spatial representations and attributes. CNNs are notable for their capacity to recognize distinctive visual aspects. One of the most important steps in comprehending and interpreting the traffic data is the feature extraction procedure.
- The utilization of RNNs for predicting and modelling traffic patterns over time is described in the study. RNNs are excellent at processing sequential data, which qualifies them to capture temporal elements of traffic data. By taking into account temporal correlations and trends, this method recognizes the dynamic character of traffic circumstances and aims to enhance traffic management.
- The paper's combination of deep learning methods including CNNs and RNNs presents a thorough method for managing traffic and making decisions. The article discusses both geographical and temporal elements of traffic data by merging various neural network designs, providing a more comprehensive approach to traffic management in smart cities.
- The demonstrated performance improvement attained by the suggested technique is one of the major contributions. The suggested CNN-RNN hybrid model surpasses other current approaches in terms of a variety of performance measures, including accuracy, precision, recall, and F1-score, according to the results presented in the study. For improving traffic flow, lowering congestion, and raising road safety in metropolitan locations, this performance improvement is essential.

The rest of the section is organised as shown below. Section 2 illustrates literature works on traffic management. Section 3 gives the Problem Statement. Section 4 covers the proposed technique for traffic management. Section 5 illustrates the performance measures and summarises the findings and compares the method's performance to previous techniques. Section 6 summarises the conclusion.

II. RELATED WORKS

Technology for communication and information integration has led to the standardization of smart linked vehicles. By incorporating various analytical and interaction methods, automated vehicles are used for traffic control, navigational help, etc. The Social Network of Things enables connected cars to make specific to an application decision by utilizing interoperability and distributed computing frameworks. The paper offers an integrated adaptive computing technique for enhancing the dependability of automotive control and managing traffic by taking into account the requirement for computation algorithms in effective connected vehicle networking. This computational approach takes into account a variety of in-range vehicle characteristics while identifying traffic and offering directed recommendations for dependable navigation in a smart city setting. The conditioned support vector machine assists this computer system in separating the complicated nature of multiflow information processing from the nearby vehicles. The physical as well as connectivity-based parameters from the SVM categorization learning-based smart automobile increase the selection's efficiency and decrease its level of complexity and processing time [14].

Modern times are characterized by quickening development and technological developments. The traffic signal, which forms the central location of the traffic structure, is one of among the most crucial places that needs development. With the growth of smart cities, this need gets stricter [10]. Despite the unrelenting efforts made to increase and improve traffic flow, vehicular traffic is now governed by highly outdated traffic lights. These conventional traffic signals have a number of issues, such as poor time management at junctions, vulnerability to weather conditions like rain, and no way to give emergency vehicles precedence. Vehicles may connect with adjacent and distant entities due to revolutionary innovations like Vehicular Ad-hoc Network and the Internet of Vehicles. The study suggests a brand-new traffic management system depending on the VANET and IoV that is appropriate for Smart Cities and next traffic networks. The structure of the suggested Intelligent Traffic Management System and Smart Traffic Signal controller is presented in the present investigation. The study demonstrates localized transportation planning at an intersection that satisfies the needs for potential smart cities, including equality, a shorter travel time, an acceptable flow of traffic, a decrease in congestion, as well as providing precedence to responding vehicles. The suggested system works better than the conventional management system, according to simulation data, and may be an ideal option for the transportation management system in upcoming Smart Cities. The average delay is greatly decreased by the suggested adaptive method, which also results in an upsurge in the total number of automobiles being serviced. In addition, the study demonstrates the STS hardware prototype that was put into use.

Nowadays, traffic forecast is quite difficult because of the rapid rate of population growth and the constant traffic density. A higher level of communication in automobiles eliminates unnecessary delay and fuel waste, environmental harm, and even deaths brought on by people becoming stuck in traffic [15]. Systems for predicting and controlling travel delays are only recently being studied, however they may not be as accurate. In order to create a smart city, the research suggested an effective IoT-based traffic forecasting system utilizing the OWENN algorithm and a traffic signal management system. The five stages of the suggested framework include IoT information collecting, feature extraction, categorization, optimal traffic IoT values, and controlling the traffic signal system. The dataset is first used to gather IoT traffic statistics. The OWENN classifiers then determine which location has greater traffic by classifying the retrieved aspects of the roadways, the weather, and trajectory data. Assuming there is greater traffic in one way, the IoT values are optimized employing IBSO, and the flow is then managed by an Intel 80286 microprocessor. The findings of the experiments demonstrate that the suggested system works better than cutting-edge techniques.

The most important component of every traffic management strategy in a smart city is the forecast of traffic flow. It can assist an individual in selecting the best route to their intended destinations. There is a lot of study on the relationship among pollutants in the atmosphere and traffic congestion employing various machine learning techniques since information regarding air pollution as well as congestion are frequently linked. There is currently no method for effectively estimating traffic flow utilizing ensemble approaches like bagging and air pollution. As a result, smart cities require an infrastructure for more precise traffic flow forecast. The purpose of the study is to anticipate traffic flow employing information on pollution [16]. There are two contributions: To start, the most effective system has been evaluated employing a variety of simple regression approaches. The most precise simulation amongst the two assessments has been identified using bagging and stacked ensemble approaches. The findings demonstrate that compared with every one of the existing regression approaches utilized in the present investigation, the K-Nearest Neighbours bagged ensemble delivers much superior outcomes. According to the empirical findings, the KNN bagged combination approach decreases the error percentage for congestion-related prediction by over thirty per cent. The consequences of several seasons, such as winter as well as summer, have not yet been studied.

Vehicle traffic management is one of the numerous difficulties that urbanization is creating. It disrupts efficient circulation of traffic, consumes time, and endangers the safety of the road. Additionally, it has an effect on other crucial services, the financial system, the environment, and one's well-being. The constantly changing characteristics of traffic on roadways and the inability of traditional technologies to perceive these patterns in real-time are the main contributors to inadequate traffic management. In technologically advanced urban environments, analysis of information services built on Edge Cloud architectures offer quick, effective answers to the problems that urbanization is posing [17]. Thus, real-time control of urban vehicle traffic using Edge Cloud technology is possible. The study develops a connected device Edge Cloud-centric automated traffic management framework for time-optimized traffic influx forecasting. The traffic inflow forecast adjusts the traffic movement stages time as needed and prevents junction congestion and lengthy waiting lines. The efficient allocation of traffic to viable routes is made achievable by smart navigation, which also increases junction road safety. The traffic influx is predicted using baseline classification algorithms, and statistical analysis recognize the J48 decision tree's superior estimation performance contrasted with additional commonly employed classifiers. Edge computing is utilized for real-time traffic load management and time-optimized smart vehicle guidance. The best traffic load balance also benefits cars at junctions from a perspective of roadway security. The outcomes show the effectiveness of the suggested method for smart navigating, ideal balanced distribution of traffic, and enhanced safety on the roadways at intersections.

The purpose of the study is to explore the viability and effectiveness of using deep learning for artificial intelligence in the context of smart cities. Depending on the Deep Belief Network method, a circulation of traffic forecasting framework is created. In Tianjin, statistics on the intended road section's previous circulation are gathered and prepared. Then, a DBN that has been learned as a model that generates information is created by stacking numerous Reduced Boltzmann Machinery [18]. The simulation study then analyses how well it performed. The suggested algorithm framework is contrasted with a deep learning Design, CNN approach, and Neuro Fuzzy C-Means framework. The findings demonstrate that the recommended algorithm system's RMSE, MAE, and MAPE is respectively high. Compared to the remaining three algorithms, it makes predictions with a substantially better degree of accuracy. Additionally, the algorithm may effectively prevent traffic congested in the connected city from spreading, allowing for prompt delays in evacuation. The developed Deep Learning-based traffic circulation forecasting approach, in brief, has a high-precision forecasting impact and traffic congestion discharge efficiency, which may serve to be experimental benchmarks for the development of smart cities in the future.

In order to build transportation systems and optimize traffic in immediate terms in big cities, effective traffic forecasting is a crucial responsibility. Forecasts produced by the mathematical techniques employed in scientific research are typically legitimate, although they can be seriously off when there is congestion. Complicated city-wide road network dynamics that computerized techniques are infrequently equipped to forecast are at the core of these scenarios. The evolution of traffic patterns is significantly impacted by the phenomenon of congestion that is spreading throughout the road system that connects large cities [19]. The Congested-based Traffic Forecasting System, which improves earlier estimates by employing congested transmission structures, is a novel traffic forecasting framework that the study introduces in the current article. The study wants to demonstrate how integrating congestion information may significantly enhance the projections. There does not exist the requirement for substitute effective procedures because the produced strategy may be utilized in combination with any prior approach. To the greatest extent of current understanding, no technique is currently in development that utilizes traffic data for predictions in this manner. According to recent effectiveness investigations, the study was capable of to improve traffic estimates by a median of by employing CTPM.

The standard of life is impacted by traffic congestion since it wastes time and frustrates people. Ambulances and police cars are two examples of vehicles that have substantial emergency needs for which the congestion is crucial. Consequently, there are more carbon dioxide emissions. Congestion patterns must be accurately modelled for traffic management. Automobile speed and the density are the two primary measurable requirements that characterize a city's level of connectedness [20]. An experiential explanation for congestion, as opposed to a quantitative one, is the chaos and unpredictability resulting from changes in traffic characteristics. In order to simulate such conditions effectively and naturally, statistical evaluation provides a structure. In the present study, a differential-entropy-based labelling method was suggested. Convolutional neural network-based supervised traffic forecasting using traffic meta-parameters was next developed. Node locality, the time, the particular day of the week, the hour, unique highway conditions, and vacations are some examples of traffic characteristics. The City Pulse information set, which consists of a collection of automobile traffic recordings from four hundred and fifty-nine observational points during six months of operation in the Danish city of Aarhus, serves as the basis for validating the proposed framework. The City Pulse datasets simulation findings show that the suggested technique produces precise forecasting levels for the various nodes taken into consideration. By redirecting automobiles to use alternative routes, the suggested method can reduce traffic congestion.

III. PROBLEM STATEMENT

From the above literature review, it is given that over the past ten years, the growing issue of traffic congestion has elevated to a significant concern in the creation of smart cities. Intelligent Transportation devices must address a number of issues, including as reducing the time it takes for communications between automobiles in motion and stationary devices, assuring smooth traffic flow, and improving road safety. Traffic congestion and the seriousness of traffic accidents have been made worse by the quick growth in the number of automobiles on the road. To address these issues, novel ideas including transportation networks, vehicle communication, navigating, and managing traffic have arisen. Intelligent transportation systems' inherent abilities may be unlocked despite specific programming due to machine learning, especially deep learning [21]. The study suggests a traffic congestion management system that uses deep learning algorithms to collect real-time traffic information and direct automobiles along feasible paths in order to handle traffic congestion in smart cities. With the long-term objective of reducing traffic difficulties, this system supplies drivers with modern technology for remotely tracking the movement of traffic and the quantity of automobiles on the road.

IV. TRAFFIC MANAGEMENT USING DEEP LEARNING

Data collection from real-time traffic camera images taken at four signal points at significant intersections in Vijayawada and Guntur, capturing information on vehicle flow. The sizeable collection, weighing in at 1.6 GB, consists of low-resolution images shot in a variety of lighting, perspectives, and environments. The data must be pre-processed, using standardization and min-max normalization for optimal uniformity across a variety of images with different lighting, angles, and atmospheric conditions. Utilizing Convolutional Neural Networks' (CNNs') capacity to identify distinctive image characteristics, extraction of characteristics is carried out utilizing these networks. For local extraction of characteristics and shift invariance, CNNs use convolutional and pooling layers, which improve the data's usefulness. In order to control traffic and make decisions, Recurrent Neural Networks (RNNs) are used, especially when managing the temporal elements of traffic data. RNNs make it possible to simulate traffic patterns over time, taking into consideration how dynamic traffic circumstances are. This helps to enhance real-time traffic flow and relieve congestion. To improve traffic management in smart cities, this complete technique combines data collecting, pre-processing, feature extraction, and RNN-based decision-making. Figure 1 gives the entire structure of the proposed approach.

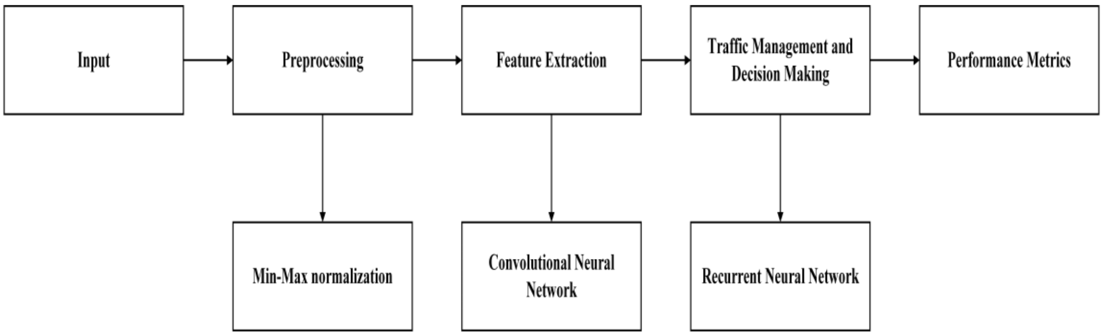


Figure 1: **Entire Structure of the Proposed Approach**

4.1 Data Collection

Real-time traffic camera images were collected to construct a dataset of CCTV images. The study has gathered information on traffic flow in the two significant towns of Vijayawada and Guntur. Guntur Market and Brundhavan Gardens are two of the intersections where the study gathered data in Guntur City, whereas Benz Circle, Seetharamapuram, Guru Nanak Colony, and Ramavarappadu Junction are four of the junctions where data was obtained in Vijayawada. Four separate signal points—signal1, signal2, signal3, and signal4—were taken into account at each junction, along with the time it takes a vehicle to get to the opposite intersection. In the cities, 83 cameras are in use [22]. The total amount of the data collection the study used for the experiments is 1.6 GB. The low-resolution images were taken under various lighting circumstances, from various angles, and in various settings. Each picture has a width of 800 pixels and a height of 600 pixels.

4.2 Pre-processing using Min-Max Normalization

Pre-processing is necessary to improve the usability of the CCTV image collection before using it for any analytical or deep learning activities. The collection, collected by traffic cameras, is made up of 1.6 GB of low-resolution images taken in various scenarios. Each image has a width and height of 800 pixels and 600 pixels, respectively. Pre-processing is essential to maintain consistency and compatibility for following activities given the variety of the images regarding the variety of angles, light levels, and atmospheric conditions. To get the pixel values into a common range for efficient evaluation, standardization, scaling, and min-max normalization are required.

The min-max normalization procedure, sometimes referred to as feature scaling, is an important stage in the pre-processing of images. Using this method, the image's pixel values are rescaled to ensure that they lie inside a predetermined range, often [0, 1]. This reduces the impact of variances in lighting and pixel distribution of values by standardizing the intensity levels throughout all images. Identifying the dataset's maximum and minimum values for pixels and then employing a linear transformation to convert every pixel's initial number to an alternative value that belongs within the defined range are the two primary processes in the min-max normalization process. Through this procedure, image consistency is improved, and it also helps with subsequent evaluation and machine learning positions to provide findings that are accurate and consistent. The dataset's lowest and maximum pixel values for each image is initially determined before applying min-max normalization to the dataset. The dataset's lowest pixel is represented by the minimum significance, and the most illuminated by the maximum value. Following the establishment of these values, the initial intensity value of each pixel is linearly changed to an alternative value falling within the [0, 1] range. Equation (1) provides the formula utilized for this transformation.

$$S_{out} = (S_{in} - Min)\frac{newMax - newMin}{Max - Min} + newMin \tag{1}$$

The image after min-max normalization is designated asS_{out}, and the novel the minimal and maximum intensities are designated as $NewMin$ and $newMax$. The original real-time traffic image is indicated asS_{in}, the minimum and maximum intensity standards, which range from 0 to 255, are denoted as Min and Max, respectively. All of the images in the dataset were subjected to this modification, which ensured that the pixel values were uniform and appropriate for further processing. By removing biases brought out by changes in illuminating and pixel value variations, this normalizing technique contributes to making the dataset more conducive to precise and consistent evaluation.

4.3 Feature Extraction Employing Convolutional Neural Network

Convolutional neural networks (CNNs) are used for feature extraction, which is a key step in computer vision applications because it takes advantage of the network's special properties including local connection and weight sharing. CNNs are made up of pooling layers that incorporate shift consistency and layers of convolution that adapt to convey distinctive characteristics within input images. The CNN exhibits exceptional qualities such as local connection to the neurons and weight sharing in its interpretation of a characteristic of an input image. The convolutional layer, which develops in order to represent the distinctive characteristics of the input image, and the pooling layer, which achieves shift invariance, are the layers of CNN. The general architecture of CNN is given in Figure 2.

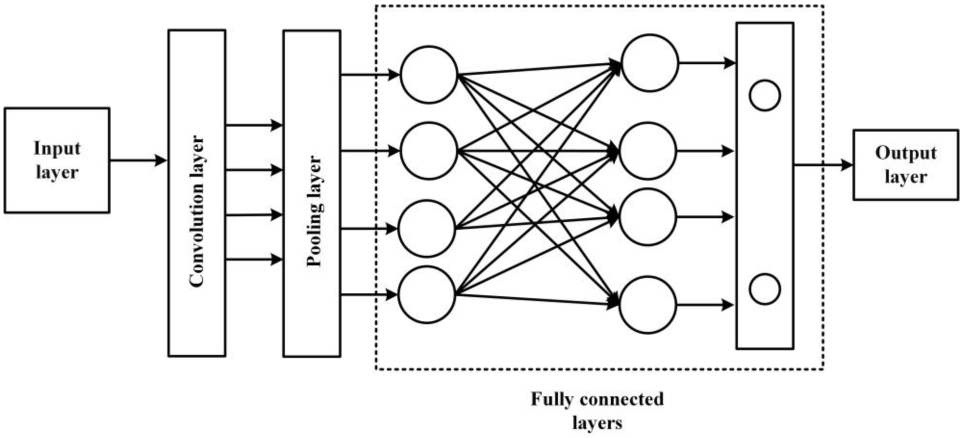


Figure 2: **General Architecture of CNN**

The nearest group of neurons from the resultant layer's preceding layer will provide input to the convolutional layer's neurons. By combining a number of kernels from the preceding layer, the different characteristic representations were built. Equation (2) results in the creation of the convolution layer.

$$u_f^j = \sigma\left(\sum_{i=1}^{f_{j-1}} u_l^{j-1}, Z1_{lf}^j + a1_f^j\right), f\,\epsilon[1, f_1] \tag{2}$$

The l^{th} activation map of the $(j-1)^{th}$ layer is indicated byu_l^{j-1}, the f^{th} activation mapping of the j^{th} convolution layer is implied byu_f^j, and $Z1_{lf}^j$ and $a1_f^j$ indicate the weight that links the lth activation map of the f^{th} layer at position. The elementwise exponential activation function and l_1 can both describe the various filters in the f^{th} layer.

The pooling processes have the necessary data but can reduce the spatial dimension of the activation map. When the outcome of the preceding layer is coiled with the dimensions *(p, q)* in the convolution filter's and handled bitwise nonlinear activation, the result is $u_f^j(x, y)$ in Equation (3). The kernel locations are b1 and a1.

$$u_f^j(x,y) = \sigma\left(\sum_{l=1}^{f_{j=1}} \sum_{b1=0}^{p=1} \sum_{a1=0}^{p=1} \left(Z1_{lf}^j(b1,a1) \otimes u_l^{j-1}(x+b1,k+a1) + a1_f^j\right)\right), f \,\epsilon [1, f_1] \quad (3)$$

By getting the results of the preceding layer with a filtrate of size *(2, 2),* the convolution layer was supervened by the position of the f^{th} activation map of the $j+1^{th}$ pooling layer; $u_l^{j+1}(x, y)$ was then produced, and the bitwise nonlinear activation was utilized employing Equation (4).

$$u_f^{j+1}(x,y) = \sigma\left(\sum_{l=1}^{f_{j=1}} \sum_{b1=0}^{p=1} \sum_{a1=0}^{p=1} \left(Z1_{lf}^{j+1}(b1,a1) \otimes u_l^{j}(2x+b1,k+a1) + a1_f^{j+1}\right)\right), f \,\epsilon \left[1, f_{j+1}\right] \quad (4)$$

4.4 Traffic Management and Decision Making Employing Recurrent Neural Network

Using deep learning techniques in traffic management and decision-making has shown to be a potential way to increase effectiveness and safety. RNN are then used to analyze the temporal elements of traffic information after the key stage of extraction of characteristics employing Convolutional Neural Networks. RNNs are an effective alternative for modelling and forecasting traffic patterns over time since they excel at jobs requiring sequential data. The RNN receives its input from the CNN's output, which generally comprises of high-level characteristics and spatial representations retrieved from static traffic images. This input contains important details on the flow of traffic at the moment, such as vehicle placements, concentrations, and patterns of movement. However, traffic situations are by nature dynamic and constantly change. It is essential to take into account the temporal relationships and sequential nature of traffic information in order to manage traffic efficiently and make informed decisions. The general architecture of RNN is given in Figure 3.

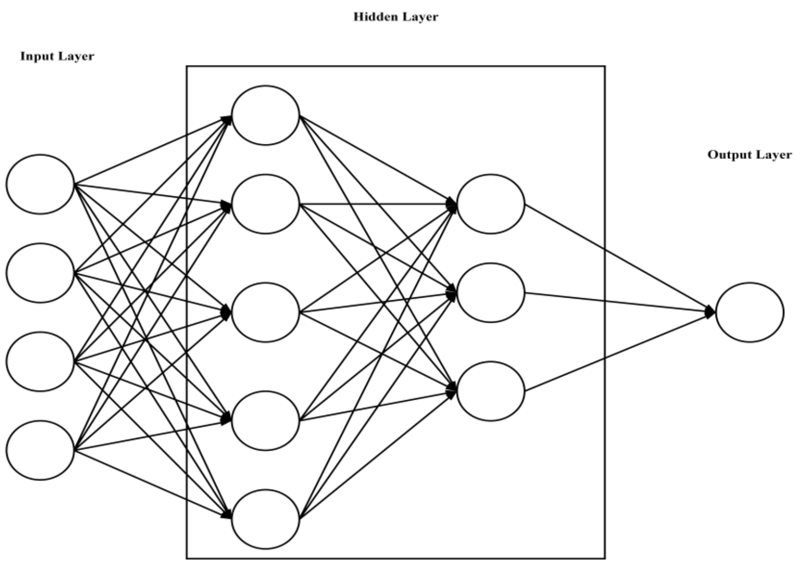


Figure 3: **General Architecture of RNN**

Because of their recurrent interactions, which enable them to preserve a hidden state that gathers input from earlier time phases, RNNs excel at processing sequential information. The network is able to store and make use of information from previous traffic observations because to this concealed state, which serves as a memory. At each time step, the RNN repeatedly updates its hidden state and generates output predictions as it analyses the incoming data sequentially. With the help of this technique, RNNs may detect intricate temporal correlations and patterns in the traffic data, including changes in traffic flow and congestion as well as repeated patterns at particular times of the day. In Equation (5), the previously computed hidden state s_{t-1} and the current input y_t are used to determine the hidden state at time step *t*, represented ass_t.

$$s_t = f(Z_{is}y_t + Z_{ss}s_{t-1} + a_s) \tag{5}$$

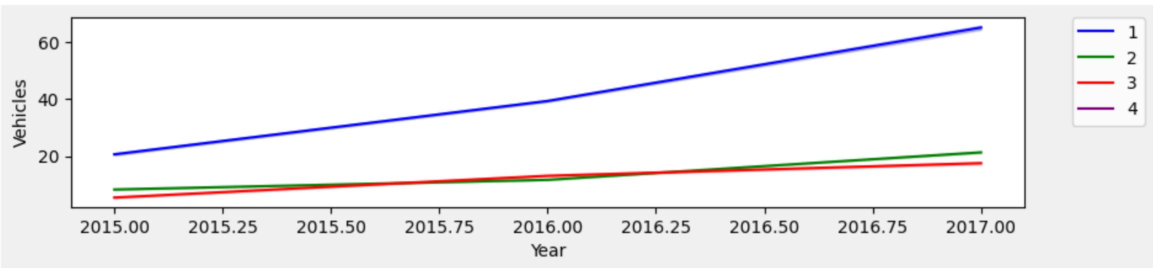
In this case, the weight matrices Z_{ss}andZ_{is}, the bias terma_s, and the activation function f typically, a hyperbolic tangential (tanh) or reconditioned linear unit (ReLU) function are all weight matrices. Equation (6) provides the output at time step t, abbreviated asy_t, which is produced depending on the current hidden state.

$$x_t = h(Z_{s0}s_t + a_0) \tag{6}$$

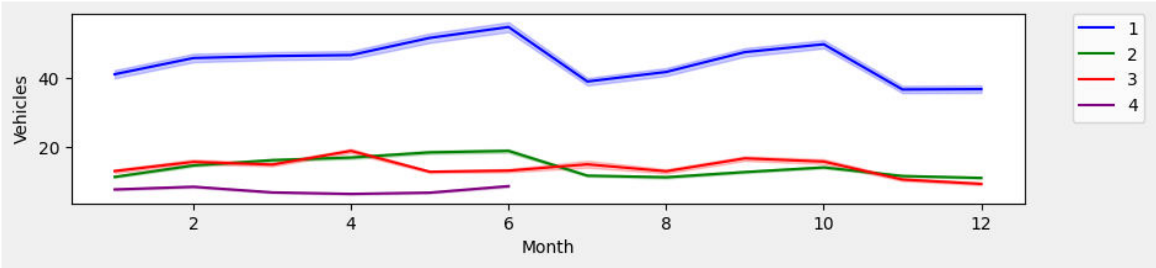
Where, a_0 is the bias terms for the resultant layer and Z_{s0} is the weight matrix. Recurrent linkages are used by RNNs to keep track of information from earlier time phases. The present input y_t and the preceding hidden states s_{t-1} are used to determine the hidden statess_t. Backpropagation across time may be employed to teach the network and modify the weights and biases, allowing it to pick up on and adjust to the temporal trends seen in the traffic data. Traffic management systems may use the capabilities of RNNs to make real-time decisions on the basis of past traffic data, resulting in improved traffic flow, decreased congestion, and increased road safety.

5. Results and Discussion

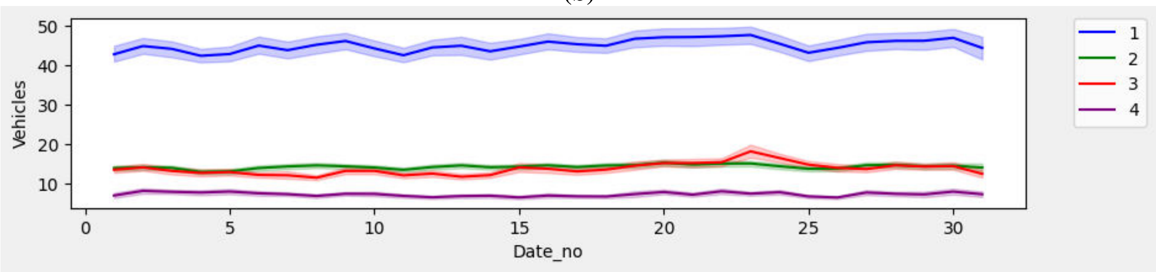
This section provides the study's findings and a discussion of them. The study's first data source was live traffic camera footage. The data's were taken from four signal points at significant intersections in Vijayawada and Guntur, capturing information on vehicle flow. For the best consistency across a range of images with various lighting, angles, and atmospheric conditions, the data must be pre-processed employing standardization and min-max normalization. These networks are used to extract attributes by using CNN's ability to recognize distinctive visual properties. CNNs employ convolutional and pooling layers for local extraction of attributes and shift invariance, which increases the value of the data. RNN are employed, especially when controlling the temporal components of traffic data, to manage traffic and make decisions. RNNs enable the simulation of traffic patterns over time while taking into account the dynamic nature of the traffic environment. Congestion is reduced and real-time traffic flow is improved as a result. This comprehensive approach includes data collection, pre-processing, feature extraction, and RNN-based decision-making to enhance traffic management in smart cities. The performance measures listed below are used to calculate the effectiveness of the suggested method.



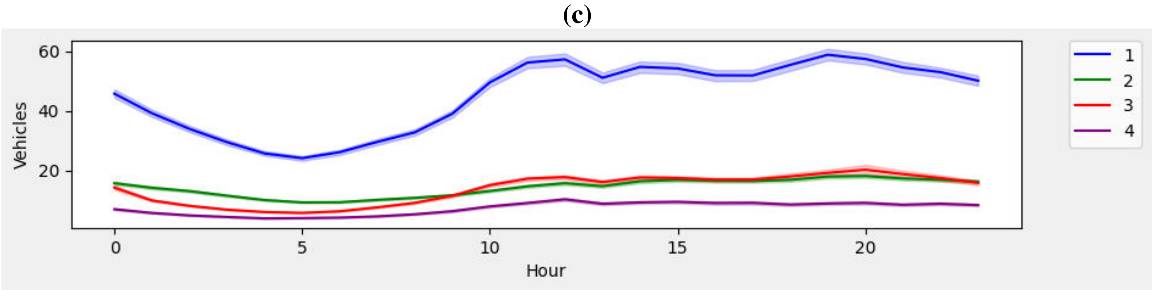
(a)



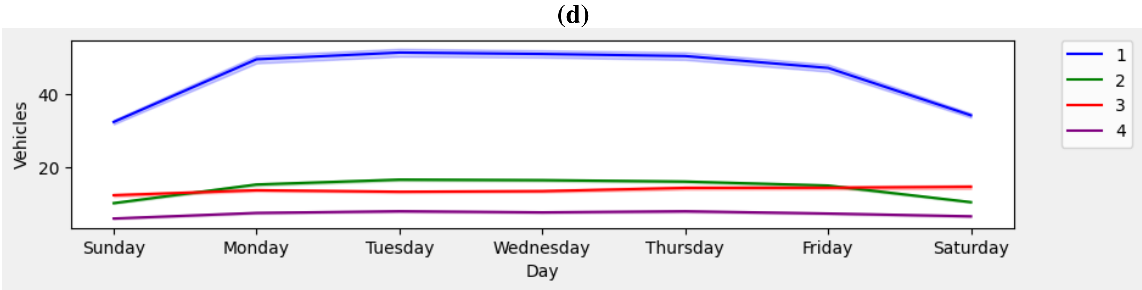
(b)



(c)



(d)



(e)

Figure 4: **Vehicle Count by Time Features and Smart Cities (a) Year (b) Month (c) Date_no (d) Hour (e) Day**

Figure 4 demonstrates how the number of vehicles and other temporal aspects relate to one another in the smart cities. There are five distinct temporal features represented by the subplots: "Year," "Month," "Date_no," "Hour," and "Day." Each subplot's x-axis reflects the corresponding time characteristic, while the y-axis displays the number of vehicles. The "Junction" variable is used to color-code the line plots, making the ability to compare the traffic patterns at various junctions visually. The separate coloration allocated to each junction makes it easier to distinguish between the various trends. These charts are useful resources for comprehending how vehicle counts change over time and at various intersections in smart cities. Using the chosen time characteristics, the line plots reveal traffic patterns and trends. These graphs may be used to identify changes in the number of vehicles over time, including years, months, days, and hours, as well as the various days of the week affect traffic. Urban planners, traffic management authorities, and researchers involved with smart city initiatives can all benefit from these visuals. They provide a simple and understandable illustration of how traffic volumes change over time, which may help with resource allocation, congestion control, and urban planning choices. Furthermore, the varied colors for each junction enable comparisons across other traffic intersections, assisting in the identification of regions that may need special attention or interventions in the framework of smart city traffic management.

Table 1: **Average Monthly Traffic**

Months	Vehicles			
	Junction 1	Junction 2	Junction 3	Junction 4
1	41.24933	11.16801	12.8797	7.497312
2	46.00073	14.53289	15.56579	8.279762
3	46.59005	16.08266	14.77016	6.627688
4	46.86528	16.83819	18.79375	6.186111
5	51.89987	18.36425	12.6754	6.555108
6	55.07917	18.79444	13.00486	8.468056
7	39.12769	11.53495	14.85081	-
8	41.91532	11.07124	12.82258	-
9	47.74167	12.59028	16.60278	-
10	50.03226	13.95565	15.66532	-
11	36.77639	11.43472	10.39167	-
12	36.89449	10.87567	9.12164	-

Figure 5: **Average Monthly Traffic**

The average monthly flow of traffic at various intersections inside a Smart City are detailed in Table 1 and Figure 5. The table displays the average number of vehicles seen each month at four separate intersections (Junctions 1, 2, 3, and 4). The values displayed in the table show the typical annual traffic volume for each individual junction. It is clear that traffic numbers vary across intersections and during the months. For instance, Junction 1, particularly in the middle of the year, has the largest average traffic flow, but Junction 2 often has lower traffic numbers. The table gives an accurate representation of the monthly variations in traffic patterns as well as how they vary at different Smart City intersections. This data is graphically represented in the accompanying Figure 6. It connects data points with lines, where each line corresponds to a particular junction. The average number of vehicles is shown on the y-axis, while the months are shown on the x-axis. It is simple to compare the traffic patterns of different junctions since each color-coded line refers to a distinct junction. By allowing data-driven decisions for traffic management and infrastructure design, this visualization aids users and planners for cities in understanding the traffic patterns in various Smart City locations.

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

Table 2: **Average Traffic by Day of the Week**

Days of the Week	Vehicles			
	Junction 1	**Junction 2**	**Junction 3**	**Junction 4**
Friday	47.04215	14.94253	14.35105	7.307692
Monday	49.40421	15.22893	13.65757	7.469551
Saturday	34.11192	10.41279	14.5969	6.546667
Sunday	32.29837	10.14416	12.30556	5.945513
Thursday	50.28448	16.02634	14.29933	7.915064
Tuesday	51.2409	16.55699	13.25192	7.921474
Wednesday	50.86255	16.41667	13.40613	7.628205

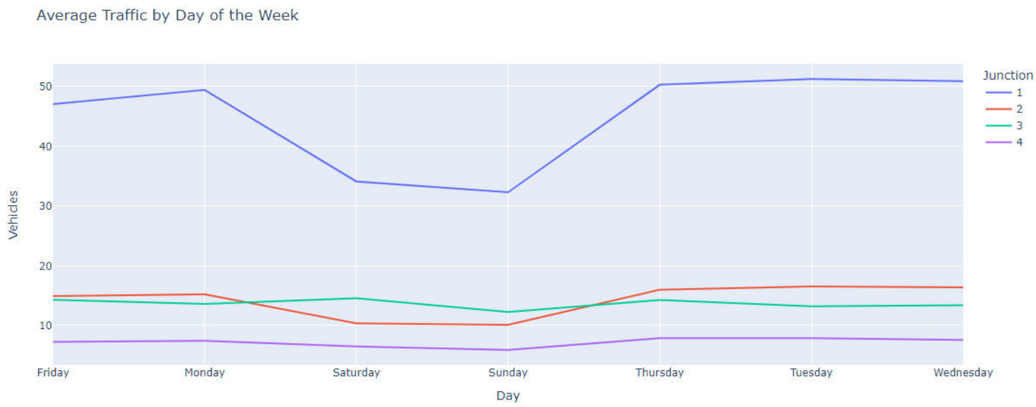


Figure 6: **Average Traffic by Day of the Week**

Table 2 and Figure 6 offer helpful data about the typical traffic patterns at different Smart City intersections on various days of the week. The average number of cars seen at four different junctions (Junctions 1, 2, 3, and 4) on each day of the week is shown in the table. These numbers show fluctuations in traffic patterns since they indicate the weekly average traffic volume. For example, it is apparent that traffic volumes are often higher on Fridays and Tuesdays at all intersections, especially Junction 1. Conversely, Sundays consistently have lower traffic levels than other days. The table provides an in-depth evaluation of traffic variations in the Smart City on different days of the week and at numerous intersections. The average number of vehicles on various days of the week is shown on the Figure 7 by a number of lines, each of which corresponds to a particular junction. The average number of vehicles is shown on the y-axis, while the x-axis reflects the days of the week. It is convenient to contrast traffic patterns at several crossroads when the lines are coloured in a specific way. The patterns seen in the table are supported by the graph. For example, the most crowded days are regularly Fridays and Tuesdays, while Sundays have the least amount of traffic. City planners and decision-makers may use this visual depiction to better comprehend the weekly fluctuations in traffic and to make well-informed choices about traffic management and infrastructure planning in the Smart City.

Table 3: **Average Traffic by Day of the Month**

Days of the Month	Vehicles			
	Junction 1	**Junction 2**	**Junction 3**	**Junction 4**
1	42.78125	13.88958	13.4875	6.944444
2	44.86458	14.08542	14.06667	8.118056
3	44.15	13.94792	13.18125	7.881944
4	42.39167	12.91667	12.66667	7.701389
5	42.84792	13.05417	12.89375	7.951389
6	44.98333	13.90208	12.15417	7.5
7	43.82292	14.3125	12.07708	7.277778
8	45.19792	14.5875	11.45833	6.826389

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

9	46.15417	14.34375	13.18542	7.381944
10	44.2625	14.03333	13.19375	7.347222
11	42.5375	13.45	12.04792	6.847222
12	44.51042	14.15208	12.51042	6.5
13	44.94375	14.57708	11.70625	6.805556
14	43.52917	14.10417	12.11042	6.881944
15	44.74167	14.26458	14.06875	6.465278
16	46	14.59167	13.72708	6.930556
17	45.3375	14.16042	13.051042	6.736111
18	44.94167	14.5625	13.51042	6.6875
19	46.72917	14.63333	14.51458	7.298611
20	47.09792	15.25625	15.5833	7.833333
21	47.15833	14.78542	15.17083	7.145833
22	47.34375	15.03958	15.38125	8.013889
23	47.675	15.07083	18.14792	7.409722
24	45.45833	14.36042	16.45833	7.784722
25	43.16667	13.72083	14.77083	6.708333
26	44.4	13.76875	13.95417	6.444444
27	45.82292	14.63333	13.65	7.701389
28	46.17917	14.7	14.6	7.375
29	46.20395	14.34868	14.28947	7.216667
30	46.96065	14.49074	14.41667	7.933333
31	44.40152	14.03788	12.43939	7.263889

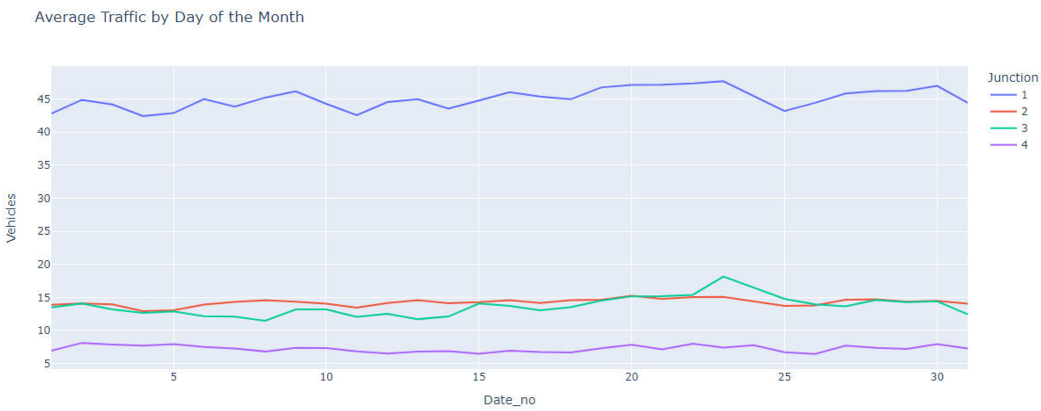


Figure 7: **Average Traffic by Day of the Month**

A full description of the typical traffic patterns for each day of the month at four different intersections inside a smart city is provided by Table 3 and Figure 7. The rows in the table's structure correspond to the days of the month, which are listed from 1 to 31. The four intersections are represented by the columns, and the numbers in the table show the typical number of vehicles at each junction on particular days of the month. The traffic management authorities and city planners are able to identify monthly trends and variations due to this tabular representation, which provides useful data on how traffic volumes change throughout the month. It would be helpful to use the accompanying graph to clearly visualize this data. This visual awareness is crucial for comprehending potential traffic cycle patterns in smart cities, which may help with resource allocation, road maintaining, and urban planning decisions.

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

Table 4: **Average Hourly Traffic**

Hourly Traffic	Vehicles			
	Junction 1	**Junction 2**	**Junction 3**	**Junction 4**
0	45.73849	15.65625	14.17434	6.922652
1	39.15625	14.11513	9.85691	5.668508
2	33.90789	12.99507	8.055921	4.320442
3	29.43092	11.46875	6.776316	3.856354
4	25.65461	10	5.978616	3.944751
5	24.06743	9.217105	5.685855	4.110497
6	26.08059	9.263158	6.236842	4.519337
7	29.52632	10.06414	7.550987	5.220994
8	32.7352	10.74507	9.057566	6.259669
9	39.00329	11.61184	11.42928	7.861878
10	49.52796	13.02961	15.00493	9.005525
11	56.1875	14.61513	17.1875	10.23757
12	57.25493	15.65789	17.70724	8.740331
13	51.07566	14.65625	16.04934	9.209945
14	54.73355	16.30757	17.57237	9.392265
15	54.21875	16.79605	17.37171	8.994475
16	51.91118	16.49342	16.88158	9.060773
17	51.85197	16.42763	16.90132	8.486188
18	55.41118	16.78289	17.92928	8.850829
19	58.80428	17.88322	19.12993	9.038674
20	57.40132	18.06086	20.20066	8.425414
21	54.53783	17.23684	18.72204	8.762431
22	52.96382	16.81908	17.39474	8.77431
23	50.08882	16.17434	15.80099	8.309392

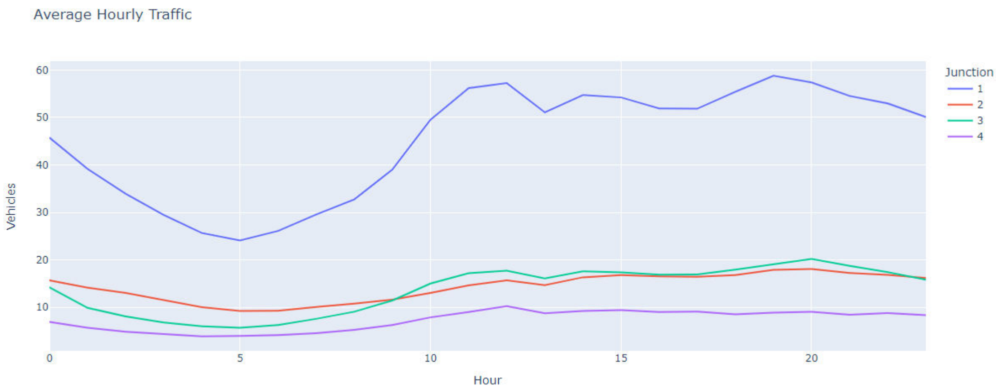


Figure 8: **Average Hourly Traffic**

An interesting summary of how traffic patterns change throughout the day for four different intersections in a smart city is provided by Table 4 and Figure 8. It is simple to see patterns and fluctuations in traffic volume during the course of a day because to the table's systematic organization, which lists hours (from 0 to 23) as rows. The numbers in the table show the typical vehicle counts at each junction for each hour of the day, while the columns reflect the four separate junctions. This tabular depiction provides helpful information into hourly traffic patterns, enabling decision-makers to allocate resources, time traffic signals, and maintain roads based on data. The graph would show how the volume of traffic changes for each junction during the day. In order for users to understand daily traffic trends, and peak periods, which may guide traffic management policies and support effective urban planning, these visual

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

representations are crucial. In a smart city, the graph would be a useful tool for making decisions about managing traffic flow and congestion at various times of the day.

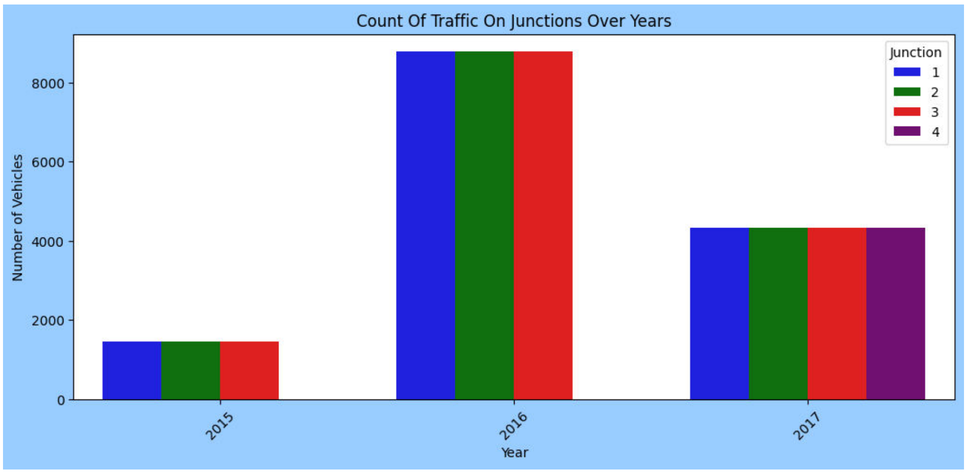


Figure 9: **Count of Traffic on Junctions over Years**

A detailed overview of traffic counts in smart cities throughout several years is shown in Figure 9. This visualisation makes use of a count plot, which is especially helpful for categorical data presentation, such as traffic counts over time. The years are shown by the x-axis, with labels identifying the particular years being taken into account. The y-axis displays the number of cars, providing a clear picture of the traffic volume. As indicated by the annotation on the upper right, this graph's distinctive feature is the inclusion of colour-coded data for four separate intersections inside the smart city. The graph enables users to discern traffic counts for various intersections and track their development over time by using separate colours for each junction. It offers a quick visual picture of any shifts or patterns in the volume of traffic at each intersection. As it helps to spot changes in patterns of traffic over time and enables data-driven decision-making, this is particularly helpful for transportation organisations and urban planners. Additionally, by reducing label overlap as well as rendering it simpler for viewers to identify certain years, the rotation of the x-axis labels improves the readability of the graph.

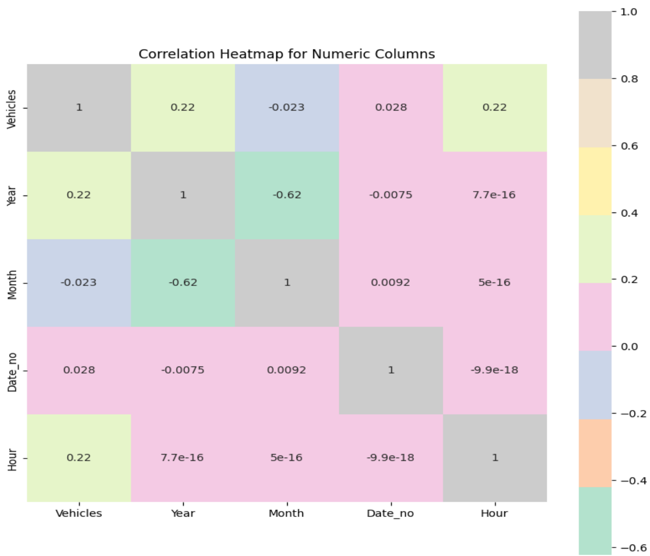


Figure 10: **Correlation Heatmap for Numeric Columns**

With a focus on the connections between "Vehicles," "Year," "Month," "Date_no," and "Hour," Figure 10 shows a correlation connecting various numerical columns in the dataset. This kind of heatmap is a useful tool for data

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

analysis since it makes it easier to spot and comprehend trends and relationships between numerical variables. The heat map's coloured squares and numerical comments shed light on the relationships' strength and direction. The correlation strength is represented by a colour scale in the heatmap; warm hues, such as dark red and bright orange, denote positive correlations, whereas cold hues, such as blue and green, denote negative correlations. In the heatmap, a white square denotes a weak or non-existent link. This knowledge makes it easier to identify which factors affect one another more strongly than others. For instance, a high positive correlation between "Month" and "Vehicles" indicates that the month of the year has an impact on the number of cars. It is simpler to determine the associations since the graphical annotations inside each square show the precise correlation coefficients. The formation of hypotheses and focused investigations are made easier with the help of this heatmap, which is especially beneficial for data investigators and analysts looking for probable patterns or relationships within the information.

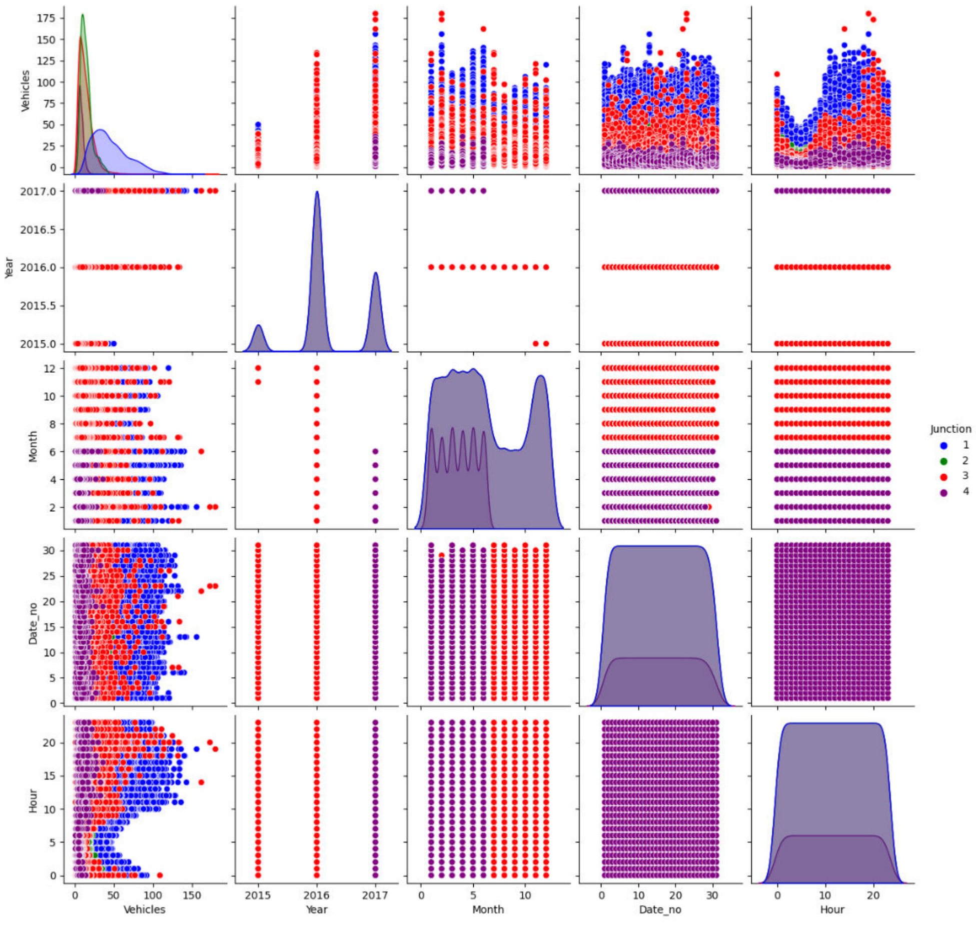


Figure 11: **Pairplot of Numeric Columns with Hue by Junction**

Figure 11 is a thorough visualization tool that enables us to investigate correlations and distributions among several numerical variables in the dataset, including "Vehicles," "Year," "Month," "Date_no," and "Hour." By using the 'Junction' variables as the colour, which creates unique measurements for each connection on the pairplot, this pairplot is enhanced with more information. Understanding how these numerical parameters interact and if these interactions vary throughout the various junctions is made easier with the use of this form of visualisation. Each scatterplot in the graph depicts the connection between two numerical parameters, with every location on the plot standing in for a dataset data point. The data points are distinguished for each intersection using colour coding with the hue Junction. This makes it simple to see how the relationships throughout variables vary between the junctions. The division of the data by "Junction" sheds light on whether these associations hold consistently across multiple regions or whether there are changes in patterns by allowing one to examine correlations, spread, and any possible anomalies within the data.

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

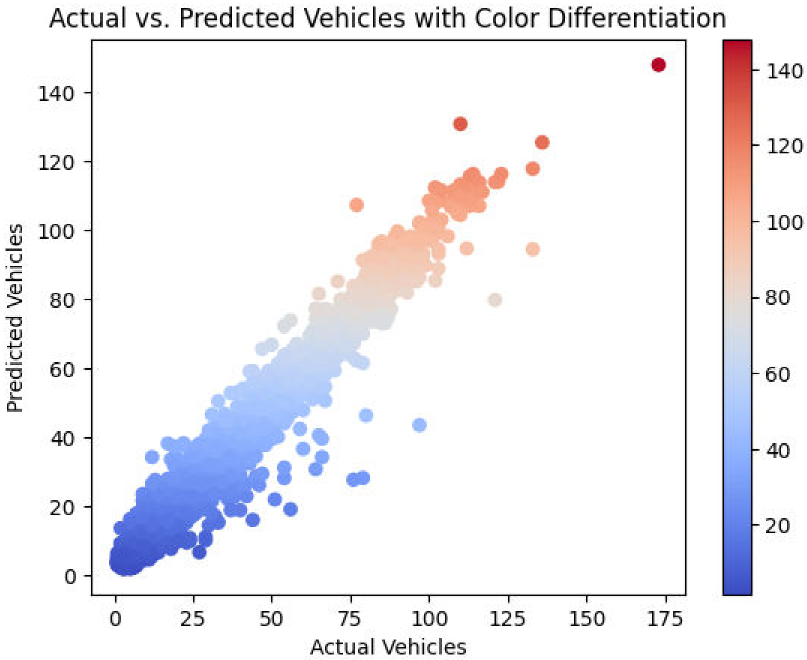


Figure 12: **Actual vs. Predicted Vehicles with Color Differentiation**

The useful visualisation for predictive forecasting, generally connected to regression analysis, is Figure 12. Each point in this graphic corresponds to a dataset observation. The 'Vehicles' variable's real or ground true values are shown on the x-axis as 'real Vehicles,' while a regression model's projected values are shown on the y-axis as 'projected Vehicles. The location of each point on the plot, which represents a single instance from the dataset, illustrates where the model's predictions stack up against the actual values that were observed. Perfect forecasts are shown by points along the 45-degree diagonal line, while the actual and expected values exactly match. With points above the line suggesting over expectations and points below the lines suggesting under predictions, departures from this diagonal line represent disparities between actual and projected values. The colour coding gives the visualisation a new level by enabling viewers to see both the relative size of the model's predictions as well as how well they match the actual data. The colour bar on the right side of the plot acts as a visual cue by illuminating the color-to-value mapping and improving the graph's readability.

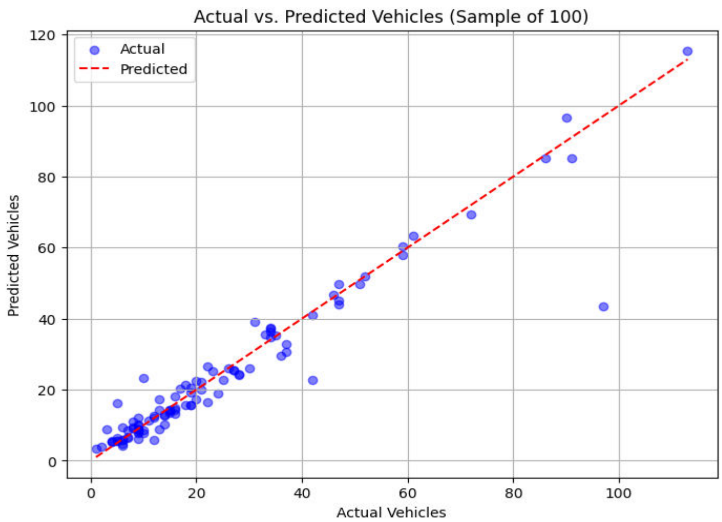


Figure 13: **Actual vs. Predicted Vehicles (Sample of 100)**

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

Figure 13 is a narrowly focused visualisation created to evaluate the effectiveness of a prediction model on a 100-point sample of data. This kind of sample is typically used to get information about how a model works on a sample that is representative of the entire dataset. Each point on this graph represents a distinct observation made from the sample. The 'Vehicles' variable's real or ground true values are shown on the x-axis as 'real Vehicles,' while the values predicted by the model are shown on the y-axis as 'Predicted Vehicles. The straight line extending from the bottom-left edge to the top-right corner is a crucial element of this layout. The ideal situation, when the anticipated and actual values coincide precisely, is represented by this diagonal line, which is red and dashed. Differences between the actual and projected values are shown by deviations from this line. Over predictions are shown by spots above the red line, while under predictions are indicated by points below it. A reasonable subset for visual evaluation is made possible by using a sample size of 100, allowing for a more in-depth analysis of the model's efficiency on a more condensed but still representative fraction of the data.

5.1 Performance Evaluation

To assess the effectiveness of detection, assessment metrics are necessary. The method most frequently used for this purpose is a measurement of precision. The classifier's effectiveness for a given testing dataset is measured by the proportion of those information that it properly recognizes. Because using the accuracy measure alone will prevent you from making the best judgments. Other criteria were used by researchers to assess the classifier's efficiency. Accuracy, recall, precision, and F1-score assessments were used to evaluate the success of the suggested strategy. The following is a description of each metric's definitions:

- T_{pos} (True Positive) refers to the amount of information that has been discovered correctly.
- The term $Fpos$ (False Positive) represents the amount of reliable information that was incorrectly identified.
- False negatives (F_{neg}) are circumstances in which inaccurate information has been identified as authentic.
- Identification of erroneous information values is known as T_{neg} (True Negative).

5.1.1 Accuracy

The accuracy of the classifier shows the frequency that it establishes the correct assumption. The ratio of correct predictions to all alternative possible theories serves as an indication of accuracy. It is demonstrated by Equation (7).

$$Accuracy = \frac{T_{pos}+T_{neg}}{T_{pos}+T_{neg}+F_{pos}+F_{neg}} \quad (7)$$

5.1.2 Precision

The precision, or degree of accuracy, of a classifier is calculated to estimate the number of properly classified outcomes. More accurate results in fewer false positives, whilst less precise results in many more false positives. Precision is defined as the proportion of examples that accurately match all instances. It is defined by Equation (8).

$$P = \frac{Tpos}{Tpos+Fpos} \quad (8)$$

5.1.3 Recall

Recall determines the sensitivity of an identification and the amount of pertinent data it generates. With better recollection, the overall quantity of F_{neg} decreases. Recall is the proportion of properly classified instances to all expected instances. This is demonstrable by Equation (9).

$$R = \frac{Tpos}{Tpos+Fneg} \quad (9)$$

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

5.1.4 F1-Score

Precision and recall are combined to create the F1-Score, a set of measures that represents the weighted mean of recall and accuracy. It is characterised by Equation (10).

$$F1 - Score = \frac{2 \times precision \times recall}{precision \times recall} \tag{10}$$

Table 5: **Comparison of Performance Metrics of the Proposed Approach with other Existing Methods**

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	95	92	92	94
NN	92	91	91	89
ENN	96	96.4	95.2	95
ANFIS	90	92.3	91	85
Proposed CNN-RNN	99.2	99	99	99

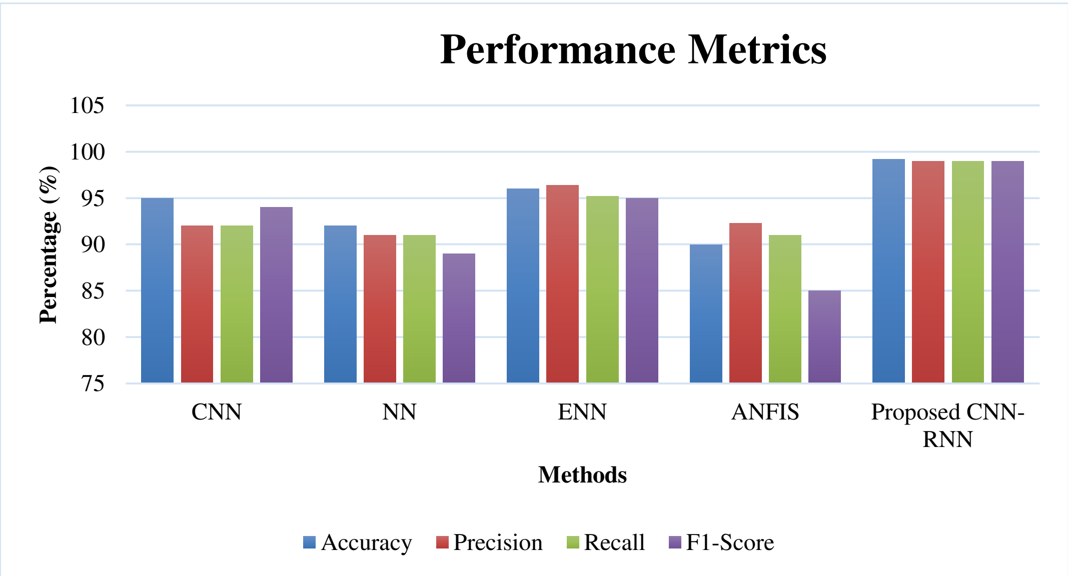


Figure 14: **Comparison of Performance Metrics of the Proposed Approach with other Existing Methods**

A detailed comparison of performance metrics for the proposed CNN-RNN methodology compared to a number of currently used traffic management techniques is shown in Table 5 and Figure 14. Accuracy, Precision, Recall, and F1-Score are among the metrics assessed; they are all presented as percentage values. The Proposed CNN-RNN technique stands out with an outstanding accuracy of 99.2%, exhibiting its exceptional traffic management. Accuracy is the percentage of cases that are properly identified. Positive prediction accuracy is measured by precision, and the proposed CNN-RNN exhibits high precision at 99%. Recall, which measures the capacity to recognize all pertinent occurrences, is 99% for the Proposed CNN-RNN, highlighting the effectiveness of its thorough monitoring. The Proposed CNN-RNN also performs outstandingly, scoring 99% on the F1-Score, which measures performance in terms of precision and recall. The proposed CNN-RNN method consistently exhibits the highest accuracy and precision, as well as excellent recall and F1-Score, making it the most outstanding option for this traffic management.

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

In contrast, the other methods, including CNN, NN, ENN, and ANFIS, demonstrate different levels of effectiveness across these metrics.

5.1.5 Mean Squared Error (MSE)

MSE, or mean squared error, is a standard statistic employed in the field of machine learning to assess how well a regression model performs. The average squared variance between the anticipated and actual values in the dataset serves as how the MSE calculates. This is demonstrable by Equation (11).

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(U_i - \widehat{U_\iota}\right)^2 \tag{11}$$

Where, m is the total amount of information points, U_i is the original values and $\widehat{U_\iota}$ is the anticipated values.

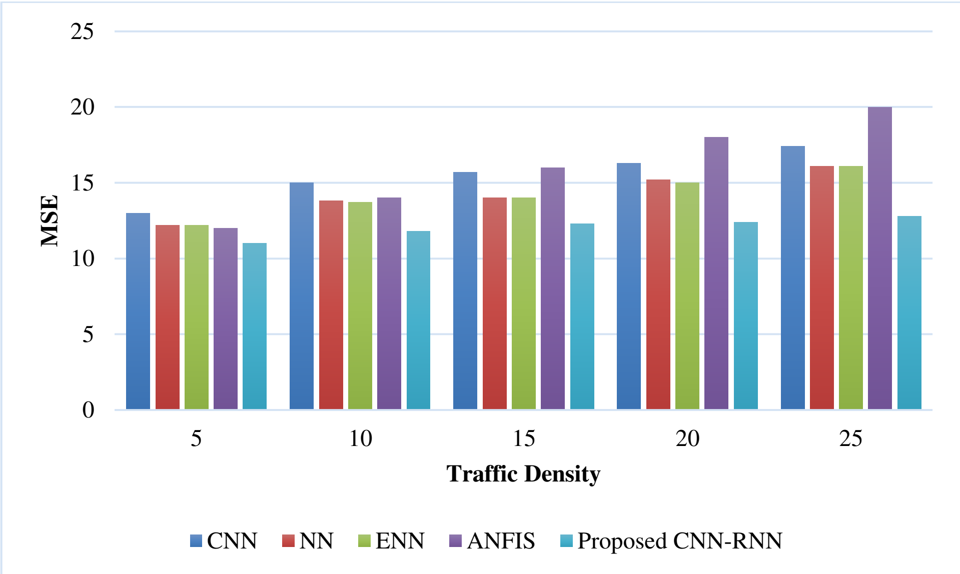


Figure 15: **Mean Squared Error**

Figure 15 presents a comparison of the Mean Squared Error (MSE) values for several models over five separate situations, with the data labeled as 5, 10, 15, 20, and 25. The average squared discrepancies between anticipated and actual values are typically measured using the MSE metric in machine learning and statistics. A Convolutional Neural Network (CNN), a Neural Network (NN), an Ensemble Neural Network (ENN), an Adaptive Neuro-Fuzzy Inference System (ANFIS), and a proposed CNN-RNN hybrid model are among the models that are assessed in this illustration. The MSE values show how effective these models are in reducing prediction mistakes. The Proposed CNN-RNN model constantly surpasses the other models across all circumstances, as seen by the fact that it consistently has the lowest MSE values. Lower MSE values suggest better predictive accuracy since the algorithm's predictions are more in line with the actual values. The ANFIS model, on the other hand, consistently generates the highest MSE values of all models, indicating less precise predictions. The proposed CNN-RNN model exhibits the maximum predictive accuracy and efficiency among the models taken into consideration, according to this graph, which provides a concise assessment of the models' effectiveness in minimizing prediction errors across different data situations.

5.1.6 Mean Absolute Error (MAE)

The average amount of errors between anticipated and real outcomes is measured using a metric called mean absolute error in statistical and machine learning. It estimates the average of these absolute variations after measuring the percentage difference between every projected value and its matching real value. A predictive model's accuracy may be easily evaluated using MAE, with reduced MAE values representing greater predictive effectiveness. It is characterised by Equation (12).

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|U_i - \widehat{U_i}\right| \tag{12}$$

Where, m is the total amount of information points, U_i is the real values and $\widehat{U_i}$ is the anticipated values.

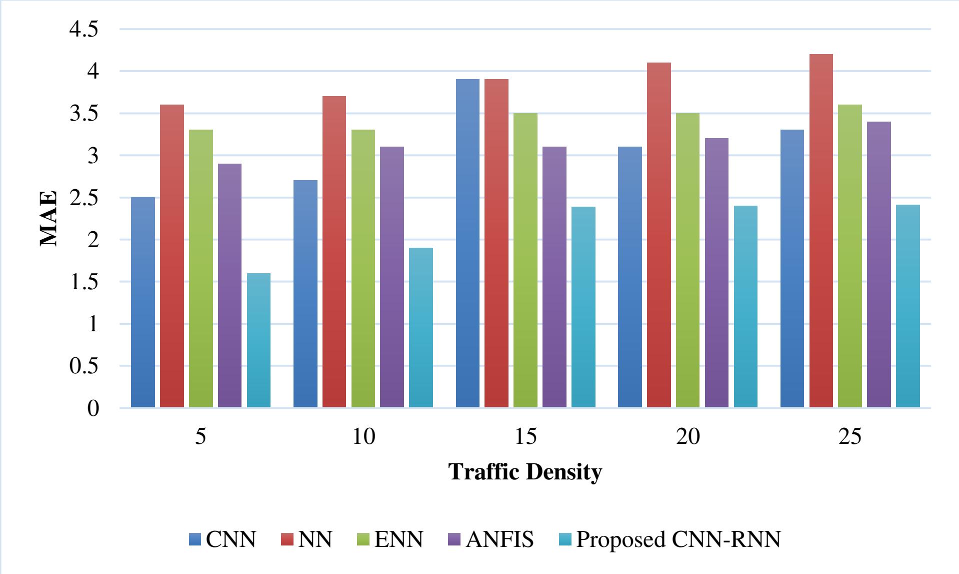


Figure 16: **Mean Absolute Error**

Figure 16 offers a comparison of the performance of several models using the MAE measure for five distinct situations or data sets, represented by the numbers 5, 10, 15, 20, and 25. The average absolute discrepancies between anticipated and actual values are measured by the MAE, a commonly employed metric in statistics and machine learning. Various models, comprising a CNN, NN, ENN, ANFIS, and a suggested CNN-RNN hybrid model, are evaluated in this figure. The MAE values for every approach are provided for each of the five information scenarios, allowing for a clear evaluation of how well each model predicts outcomes in various circumstances. It appears from an in-depth investigation of the figures values that the suggested CNN-RNN model consistently beats the other models in terms of MAE for each of the five cases. Increased prediction accuracy may be inferred from smaller MAE values, which show that the assumptions made by the model are similar to the actual values. The ANFIS model, on the other hand, regularly produces the highest MAE values between all models, indicating that its forecasts are the least reliable. Researchers and practitioners can use these MAE values as a useful tool to evaluate how well various models handle the particular data circumstances shown here. The suggested CNN-RNN model shows the greatest effectiveness among the models taken into consideration, according to the graph, which highlights how these models execute with regard to of prediction accuracy across different data sets.

5.1.7 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error, or MAPE, is a frequently employed statistic for assessing how well regression algorithms operate in machine learning. It calculates the typical absolute percentage difference between the actual and projected values of a target variable. Equation (13) contains the MAPE equation.

$$MAPE = \frac{1}{m}\sum_{t=1}^{m}\left|\frac{u_t - \widehat{u_t}}{u_t}\right| \times 100\% \tag{13}$$

Where u_t is the target variable's actual value, m is the sample size, and $\widehat{u_t}$ is the target variable's predicted value.

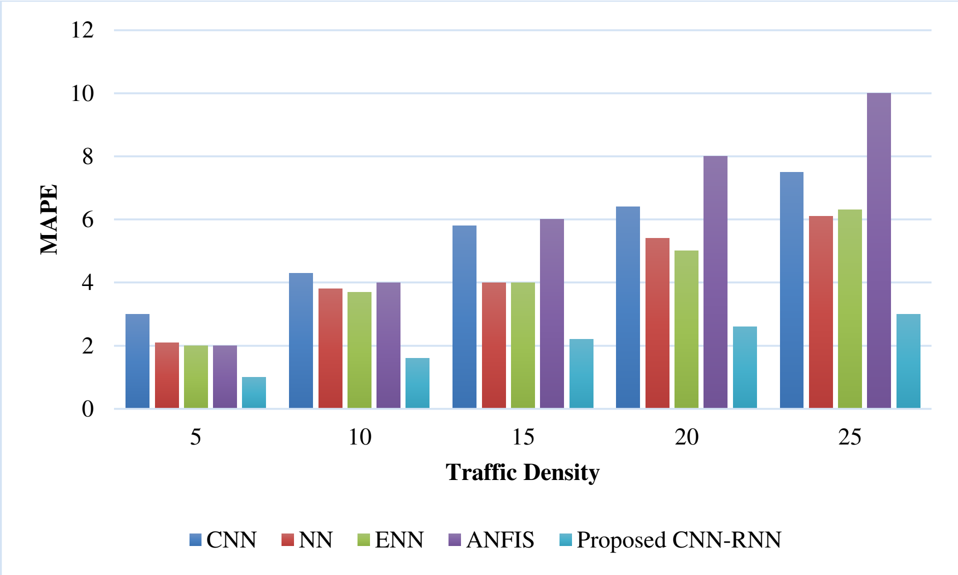


Figure 17: **Mean Absolute Percentage Error**

The MAPE values for several models are compared over five different situations, which are represented by the data values 5, 10, 15, 20, and 25 in the Figure 8. The percentage difference between expected and actual values is measured using the commonly used metric known as MAPE in forecasting and predictive modelling. Various models, including CNN, NN, ENN, ANFIS, and a proposed CNN-RNN hybrid model, are evaluated in this figure. The accuracy of the forecasts produced by these models is measured by the MAPE values. The Proposed CNN-RNN model consistently surpasses the remaining models across every circumstance, as seen by its demonstrated that it consistently displays the lowest MAPE values. Lower MAPE values represent better predicting accuracy since they demonstrate how closely the model's predictions match the actual data. In contrast, the ANFIS model regularly generates the greatest MAPE values, indicating less precise predictions. The proposed CNN-RNN model shows the maximum predictive precision and effectiveness among the models taken into consideration in this graph, which gives a succinct assessment of the models' effectiveness in predicting accuracy across multiple data situations.

5.1.8 Root Mean Square Error (RMSE)

RMSE is a common statistic for evaluating the efficacy of regression approaches. It figures out the average variance between the anticipated and actual results through taking into consideration the squared variances. RMSE is particularly useful when larger mistakes are more serious since it highlights larger disparities. In (14) below, its equation is provided.

$$RMSE = \sqrt{\sum_{j=1}^{M} \frac{\|u(j) - \hat{u}(j)\|}{N}} \tag{14}$$

Here, the variable j is shown as well as the non-missing information points M, the real-life observation time series$u(j)$, and the predicted observation time series$\hat{u}(j)$.

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 10, October 2023 |

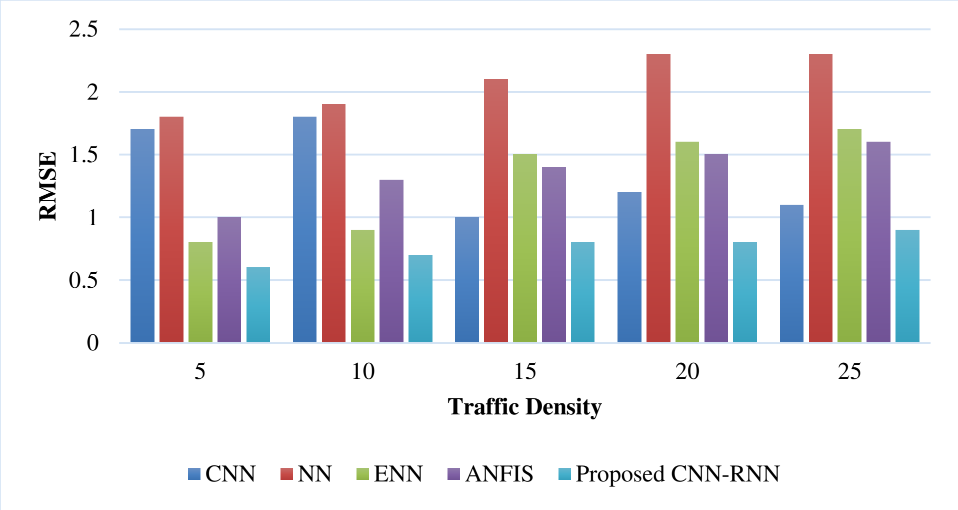


Figure 18: **Root Mean Square Error**

Figure 18 displays a comparison of the performance of several models employing the RMSE measure for five distinct situations or data sets, indicated by the values 5, 10, 15, 20, and 25. By computing the square root of the average of the squared variances between projected and actual values, the RMSE metric is a frequently employed statistic to assess the precision of forecasting techniques. CNN, NN, ENN, ANFIS, and a suggested CNN-RNN hybrid model are among the models that are evaluated in this illustration. The RMSE values for all models are shown for each of the five information situations, making it easy to compare how well each one predicts in various circumstances. When values are examined in the graph, it is clear that the suggested CNN-RNN model consistently exceeds the remaining methods with regard to of RMSE throughout all five circumstances. Lower RMSE values show that the model's predictions are more accurate since they are more in accordance with the measured data. On the other hand, the ANFIS model regularly produces the highest RMSE values across all models, indicating that its forecasts are the least reliable. Practitioners as well as researchers can evaluate the efficacy of various models in managing the information supplied here using these RMSE values. The suggested CNN-RNN model exhibits the greatest efficiency among the models taken into consideration, according to the graph, which provides an indication of how these algorithms execute with regard to of accuracy in prediction across different data sets.

5.2 Computational Time

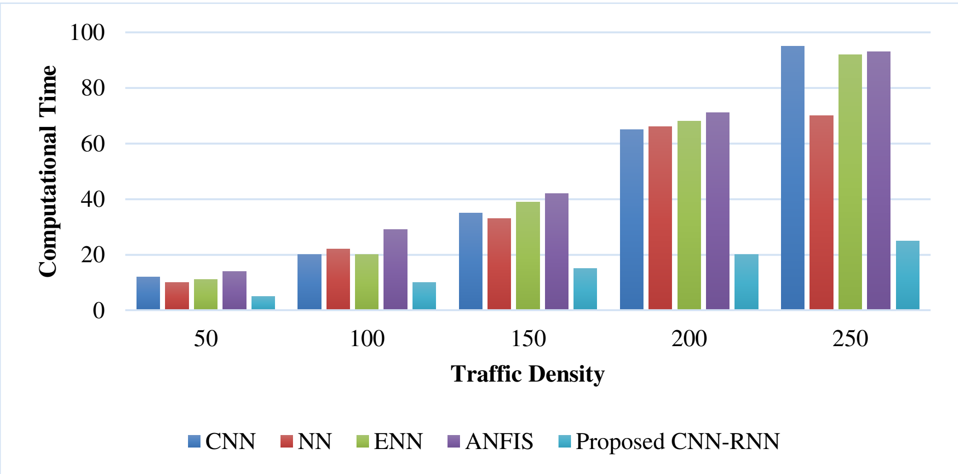


Figure 19: **Computational Time**

Using five different scenarios with computing loads of 50, 100, 150, 200, and 250, Figure 19 compares the computational times, expressed in seconds, for various models. The amount of time it requires for each model to run and finish its duties is referred to as computational time. Several models, including CNN, NN, ENN, ANFIS, and a proposed CNN-RNN hybrid model, are assessed in this image. The figure illustrates how well each model manages a range of computational loads. Upon closer inspection, it becomes apparent that the Proposed CNN-RNN model continuously surpasses the other models by exhibiting the quickest computing times in all circumstances. Lower calculation times suggest quicker and more effective task completion, indicating more computational efficiency. However, across all models, the ANFIS model consistently has the longest computing durations, indicating less effective processing. The proposed CNN-RNN model consistently achieves the quickest execution times, making it the most effective option among the models taken into consideration. This graph provides a simple assessment of the models' efficacy in terms of computing effectiveness across different computational load situations.

VI. CONCLUSION AND FUTURE SCOPE

In conclusion, this study offers an adequate and comprehensive deep learning strategy designed specifically for real-time traffic management in smart cities. This work covers the complex geographical and temporal characteristics of traffic data with a well-structured technique that includes data collecting, pre-processing, feature extraction using CNN, and decision-making using RNN. In order to enhance traffic flow and decision-making in urban contexts, this research's initial achievements establish in its capacity to accurately record and simulate the dynamics of traffic situations. When compared to existing techniques, the suggested CNN-RNN hybrid model performs better, as evidenced by improved accuracy, precision, recall, and F1-score. Traffic management systems in smart cities are able to make intelligent decisions in real-time based on previous traffic data due to the combination of CNNs for spatial feature extraction and RNNs for temporal pattern modelling. This strategy has a great deal of potential for reducing traffic, boosting road safety, and improving transit systems, which will lead to more effective and sustainable urban life. Adoption of intelligent traffic management systems, such as the one described in this study, is essential for assuring a smoother, safer, and more environmentally friendly future for urban mobility as cities continue to expand and change. Future work for this paper could consist of creating a real-time traffic management system based on the suggested methodology, integrating it with the current traffic infrastructure in smart cities, and investigating more sophisticated deep learning models and predictive algorithms to further improve traffic volume predictions and decision-making in dynamic traffic situations.

REFERENCES

[1] S. Khatri *et al.*, "Machine learning models and techniques for VANET based traffic management: Implementation issues and challenges," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1778–1805, 2021.

[2] M. Humayun, F. Ashfaq, N. Z. Jhanjhi, and M. K. Alsadun, "Traffic Management: Multi-Scale Vehicle Detection in Varying Weather Conditions Using YOLOv4 and Spatial Pyramid Pooling Network," *Electronics*, vol. 11, no. 17, p. 2748, Sep. 2022, doi: 10.3390/electronics11172748.

[3] I. Gokasar, D. Pamucar, M. Deveci, and W. Ding, "A novel rough numbers based extended MACBETH method for the prioritization of the connected autonomous vehicles in real-time traffic management," *Expert Systems with Applications*, vol. 211, p. 118445, Jan. 2023, doi: 10.1016/j.eswa.2022.118445.

[4] M. A. A. Al-qaness, A. A. Abbasi, H. Fan, R. A. Ibrahim, S. H. Alsamhi, and A. Hawbani, "An improved YOLO-based road traffic monitoring system," *Computing*, vol. 103, no. 2, pp. 211–230, Feb. 2021, doi: 10.1007/s00607-020-00869-8.

[5] A. Kambur, L. Kushchenko, and I. Novikov, "Improving traffic management through the use of intelligent transport systems," *MATEC Web Conf.*, vol. 341, p. 00044, 2021, doi: 10.1051/matecconf/202134100044.

[6] S. Kannan *et al.*, "Ubiquitous Vehicular Ad-Hoc Network Computing Using Deep Neural Network with IoT-Based Bat Agents for Traffic Management," *Electronics*, vol. 10, no. 7, p. 785, Mar. 2021, doi: 10.3390/electronics10070785.

[7] X. Mu, "Public Security Road Traffic Management Strategy based on Big Data and Intelligent Dispatching System," in *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, IEEE, 2022, pp. 589–592.

[8] M. Masuduzzaman, A. Islam, K. Sadia, and S. Y. Shin, "UAV-based MEC-assisted automated traffic management scheme using blockchain," *Future Generation Computer Systems*, vol. 134, pp. 256–270, 2022.

[9] S. K. Sood and others, "Smart vehicular traffic management: An edge cloud centric IoT based framework," *Internet of Things*, vol. 14, p. 100140, 2021.

[10] S. A. Elsagheer Mohamed and K. A. AlShalfan, "Intelligent Traffic Management System Based on the Internet of Vehicles (IoV)," *Journal of Advanced Transportation*, vol. 2021, pp. 1–23, May 2021, doi: 10.1155/2021/4037533.

[11] M. Saleem, S. Abbas, T. M. Ghazal, M. Adnan Khan, N. Sahawneh, and M. Ahmad, "Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques," *Egyptian Informatics Journal*, vol. 23, no. 3, pp. 417–426, Sep. 2022, doi: 10.1016/j.eij.2022.03.003.

[12] D. D. Nguyen, J. Rohacs, and D. Rohacs, "Autonomous flight trajectory control system for drones in smart city traffic management," *ISPRS International Journal of Geo-Information*, vol. 10, no. 5, p. 338, 2021.

[13] D. S. Praveen and D. P. Raj, "RETRACTED ARTICLE: Smart traffic management system in metropolitan cities," *J Ambient Intell Human Comput*, vol. 12, no. 7, pp. 7529–7541, Jul. 2021, doi: 10.1007/s12652-020-02453-6.

[14] G. Manogaran, J. J. Rodrigues, S. A. Kozlov, and K. Manokaran, "Conditional support-vector-machine-based shared adaptive computing model for smart city traffic management," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 174–183, 2021.

[15] N. S, B. M. A, S. Tripathi, B. D. V, I. Bhardwaj, and A. N, "IoT Based Traffic Prediction and Traffic Signal Control System for Smart City," In Review, preprint, Apr. 2021. doi: 10.21203/rs.3.rs-306500/v1.

[16] 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, no. 7, p. 4164, Mar. 2022, doi: 10.3390/su14074164.

[17] Sahil and S. K. Sood, "Smart vehicular traffic management: An edge cloud centric IoT based framework," *Internet of Things*, vol. 14, p. 100140, Jun. 2021, doi: 10.1016/j.iot.2019.100140.

[18] G. Chen and J. Zhang, "Applying Artificial Intelligence and Deep Belief Network to predict traffic congestion evacuation performance in smart cities," *Applied Soft Computing*, vol. 121, p. 108692, May 2022, doi: 10.1016/j.asoc.2022.108692.

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

[20] 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.

[21] 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.

[22] N. Algiriyage, R. Prasanna, E. H Doyle, K. Stock, and D. Johnston, "Traffic Flow Estimation based on Deep Learning for Emergency Traffic Management using CCTV Images," Jun. 2020.