International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Feb-24)

http://dx.doi.org/10.12785/ijcds/150167

Improving Detection and Prediction of Traffic Congestion in VANETs: An Examination of Machine Learning

Mohammed S Jasim [1,2], Nizar Zaghden [1] and Mohamed Salim Bouhlel [1]

[1] *SETIT-Smart Systems for Engineering Research Lab, University of Sfax*
[2] *National School of Electronics and Telecommunications University of Sfax*

Received 19 Aug. 2023 Revised 29 Jan. 2024 Accepted 31 Jan. 2024 Published 5 Feb. 2024

Abstract: Traffic congestion remains a pressing challenge in urban areas, causing significant economic and environmental repercussions. To address this issue, accurate detection and prediction of traffic congestion are imperative for effective traffic management and planning. This research study investigates the efficacy of Support Vector Machines (SVM) and various other machine learning algorithms in augmenting traffic congestion detection and prediction for Vehicular Ad hoc Networks (VANETs). Leveraging historical congestion patterns, we train and evaluate the performance of the algorithms. Our results demonstrate the potential of SVM, coupled with advanced feature engineering techniques, to outperform other methods in accurately identifying and forecasting traffic congestion. The SVM classifier achieved an impressive classification accuracy of 0.99, showcasing its effectiveness in handling diverse traffic scenarios. Additionally, the K-Nearest Neighbors (KNN) and Ensemble Learning classifiers also yielded commendable accuracies of 0.99. Notably, the Decision Tree (DT) classifier attained a perfect accuracy score of 1.00, indicating its robustness in handling congestion patterns. The proposed approach not only achieves high detection accuracy but also exhibits remarkable robustness and scalability, enabling its application across various traffic scenarios. These findings contribute significantly to the development of intelligent traffic management systems, providing valuable insights into optimizing transportation networks. Ultimately, implementing our approach holds the potential to alleviate congestion, enhance travel efficiency, and foster urban sustainability.
Keywords: Traffic congestion, (SVM), random forest, logistic regression, decision trees, VANETs.

1. Introduction

Urban areas worldwide grapple with the persistent challenge of traffic congestion, a phenomenon that not only exerts a considerable economic toll but also poses environmental concerns. The detrimental impact of congestion on efficiency, productivity, and air quality underscores the urgency of effective traffic management and planning. In this context, the present research delves into the realm of Vehicular Ad hoc Networks (VANETs), aiming to enhance the detection and prediction of traffic congestion through the utilization of machine learning algorithms.

The phenomenon of urbanization has led to a significant increase in traffic congestion, posing considerable challenges to urban transportation systems worldwide. The growing prevalence of Vehicular Ad-Hoc Networks (VANETs) on roadways has resulted in extended travel times, fuel wastage, and adverse environmental consequences. Traditional traffic management methods often rely on manual observations or fixed control systems, which may struggle to adapt to constantly changing traffic conditions. As a result, there is an urgent need to devise effective techniques for detecting and predicting traffic congestion to alleviate its economic and environmental impact.

To address these concerns, the integration of machine learning algorithms, such as SVM, DT, KNN and Ensemble Learning, holds significant promise in creating intelligent traffic management systems. SVM is well-regarded for its ability to handle high-dimensional data and complex, non-linear traffic patterns, making it a suitable choice for analyzing urban traffic dynamics. Similarly, Decision Trees offer interpretable and understandable decision-making processes, while KNN provides a simple yet effective approach to classification tasks. Ensemble Learning techniques like Random Forest and Gradient Boosting combine the predictions of multiple models to enhance accuracy and robustness.

Leveraging these machine learning methods can lead to the development of real-time, data-driven solutions for accurate congestion detection and prediction. By conducting thorough research in this domain, the potential of SVM, DT, KNN, and Ensemble Learning for traffic congestion detection and prediction can be explored, shedding light on their effectiveness in addressing the complexities of urban traffic. This research endeavor aims to significantly contribute to the advancement of intelligent transportation systems and to provide insights to policymakers for effective

Mohamed.jasim.doc@enetcom.usf.tn, nizar.zaghden@gmail.com

traffic management strategies.

Ultimately, the outcomes of this study have the potential to optimize transportation networks, improve traffic flow, reduce travel times, and enhance overall transportation efficiency, benefiting both commuters and the environment. This research effort seeks to gain a deeper understanding of the application of SVM, Decision Trees, KNN, and Ensemble Learning in traffic congestion detection and prediction, thus advancing the field of intelligent transportation systems and offering valuable insights for the development of proactive, data-driven approaches to traffic management.

A. *Research Objectives*

This paper is centered on the advancement of traffic congestion detection and prediction within Vehicular Ad Hoc Networks (VANETs) through a comprehensive exploration of machine learning algorithms. The challenges associated with dynamic and unpredictable traffic conditions are aimed to be addressed by leveraging the capabilities of machine learning models. Specifically, the performance of various algorithms in accurately detecting and predicting traffic congestion in real-time VANET scenarios will be evaluated. A thorough analysis of algorithmic approaches will be undertaken, considering factors such as data collection, feature engineering, and model training. Valuable insights that can enhance the reliability and efficiency of traffic management systems in VANETs are sought to be contributed through this research, ultimately leading to improved road safety and traffic flow.

B. *Scope*

The primary aim of this research is to investigate the utility of various machine learning algorithms, specifically SVM, DT, KNN, and Ensemble Learning, in improving the detection and prediction of traffic congestion in urban areas. This research seeks to leverage SVM's strengths in handling high-dimensional data and non-linear relationships to address existing limitations in traffic congestion analysis.

The research will commence with a comprehensive literature review, providing insights into the current landscape of traffic congestion detection and prediction methods while identifying areas for improvement and research gaps. Particular emphasis will be placed on understanding the theoretical foundations and principles of SVM and evaluating its potential applicability in the realm of traffic congestion analysis. Data collection will be diverse, including traffic flow data and historical congestion patterns. This dataset will be instrumental in both training and evaluating the performance of the SVM model, which will undergo further refinement through parameter tuning and the exploration of various kernel functions. Additionally, feature engineering techniques will be deployed to extract relevant features from the dataset, enhancing the model's accuracy.

To assess SVM's effectiveness in traffic congestion detection and prediction, its performance will be compared to that of other commonly employed machine learning algorithms, such as DT, KNN, and Ensemble Learning. This comparative analysis will offer valuable insights into the relative strengths and weaknesses of each approach. Furthermore, the research will scrutinize the impact of different feature selection techniques on SVM's performance. Evaluating SVM's real-time capabilities in traffic congestion detection and prediction will constitute a pivotal facet of the study, potentially involving the use of case studies or simulations.

The scope of this research revolves around addressing the challenges associated with urban traffic patterns through the utilization of SVM, DT, KNN, and Ensemble Learning for traffic congestion detection and prediction. By making significant contributions to the field of intelligent transportation systems, this study aims to optimize transportation networks, alleviate congestion, and ultimately yield benefits for both commuters and the environment.

The paper follows a structured format with an introduction outlining research objectives and scope. The literature review covers traffic congestion definition, causes, and existing approaches, focusing on machine learning and Support Vector Machines (SVM). The materials and methods section explains data sources, collection, preprocessing, and the application of SVM, K-Nearest Neighbor, Decision Tree, and Ensemble Learning. Results for traffic congestion detection are presented for each algorithm, followed by a discussion of the findings. The conclusion summarizes research contributions, addresses limitations, and suggests future directions. The organized structure ensures clarity and coherence in presenting the research process and outcomes.

2. Use Literature Review

A. *Traffic Congestion: Definition and Causes*

Traffic congestion refers to the condition in which the demand for transportation exceeds the available capacity of the road network, leading to slower speeds, longer travel times, and increased delays for motorists. It is a prevalent issue in urban areas, particularly during peak hours when there is a high volume of VANETs on the road.

Causes of Traffic Congestion:

High Volume of Vehicles: One of the primary factors leading to traffic congestion is the substantial increase in the number of vehicles on the road. As the population continues to grow and more individuals own cars, the road network becomes increasingly congested, resulting in traffic gridlock.

Inadequate Infrastructure: A lack of sufficient road capacity and infrastructure can significantly contribute to traffic congestion. When the road network fails to accommodate the growing volume of vehicles, it leads to bottlenecks, delays, and traffic jams.

Traffic Incidents and Accidents: Traffic incidents, such as accidents, vehicle breakdowns, and road closures, can have a substantial impact on traffic flow and often lead to congestion. These disruptions disrupt the normal flow of vehicles and frequently necessitate lane closures or detours, causing traffic delays and congestion.

Inefficient Traffic Management: Poorly executed traffic management strategies can worsen congestion. Factors like poorly timed traffic signals, inadequate signage, and improper lane configurations can contribute to traffic bottlenecks and hinder the smooth flow of vehicles.

Traffic Bottlenecks: Specific locations on the road network, such as intersections, highway ramps, and merge points, are prone to traffic bottlenecks. These bottlenecks occur when the capacity of these areas is insufficient to meet the demand of traffic, resulting in congestion.

Lack of Public Transportation Options: In cities where public transportation systems are lacking or inefficient, more individuals rely on private vehicles for their daily commutes. This increases the number of vehicles on the road, further contributing to traffic congestion.

Special Events and Peak Travel Times: Special events, including concerts, sporting events, or festivals, can draw a large number of people to a particular area, overwhelming the road network and causing congestion. Additionally, during peak travel times such as morning and evening rush hours, congestion is often more severe due to increased traffic demand.

B. Existing Approaches to Traffic Congestion Detection and Prediction

Numerous studies have been conducted in the literature to explore the application of machine learning algorithms for the detection and prediction of traffic congestion. These approaches leverage historical traffic data, sensor information, and various relevant factors to develop accurate models for congestion detection and prediction.

One study [1]. applied bi-directional long short-term memory and stacked auto-encoder methods, along with weather information, to predict traffic events from Twitter messages. The results indicated a significant improvement in model performance.

Another study [2]. focused on assessing traffic prediction methods using a newly generated dataset that incorporated factors such as road traffic data, weather data, driving angles, and congestion levels. The study employed long short-term memory, gated recurrent unit, and stacked auto-encoder methods, demonstrating the accuracy of the new dataset.

Similarly, a different research effort [3] employed machine learning techniques to monitor traffic congestion and identify traffic patterns. This study successfully identified specific time slots with congestion and monitored average speeds.

In the realm of short-term traffic flow prediction, multiple studies proposed various machine learning approaches. For instance, one study [4] compared deep neural networks (DNN), distributed random forest (DRF), gradient boosting machines, and the generalized linear model. Their analysis utilized attributes such as traffic flow, speed, occupancy, and time to reduce prediction errors, with artificial neural networks (ANNs) performing better than other methods [5].

Another study [6] investigated traffic congestion in Internet-based cities using various techniques like decision trees, random forest, support vector machine, multi-layer perceptron, and logistic regression. Among these methods, logistic regression demonstrated superior performance.

Furthermore, the incorporation of weather data in traffic flow prediction has shown promising results. Research[7] combined weather and traffic flow data using autoregressive integrated moving average (ARIMA), ANN, and dynamic Bayesian network (DBN) methods, significantly improving traffic flow prediction.

In [8], Akhtar and Moridpour (2021) proposed a mixed-effects state space model that combines Hidden Markov Models (HMM) and machine learning algorithms to detect traffic congestion. Their approach analyzed traffic flow patterns and demonstrated effectiveness using real-world traffic data.

Also in [9], Gu et al. (2019) introduced a Bayesian multi-task learning model for short-term traffic congestion prediction. Their model integrated multiple related tasks, such as traffic speed prediction and flow prediction, to enhance congestion prediction accuracy, using data from a California freeway.

Azzouni and Pujolle (2017) in [10] presented a Long Short-Term Memory (LSTM) network for traffic congestion prediction. LSTM, a type of recurrent neural network, captured long-term dependencies in sequential data and considered various traffic factors, including historical traffic flow and weather conditions, achieving accurate predictions using real-time traffic data from a major road network in China.

Chen et al. (2021) in [11] introduced a spatiotemporal correlation-based approach for traffic congestion prediction. Their support vector regression model captured spatial and temporal relationships among traffic flow data and considered the distribution patterns of congestion, achieving accurate predictions, evaluated using data from a large metropolitan area in China.

These studies exemplify the effectiveness of machine learning algorithms, including Support Vector Machines, Hidden Markov Models, Bayesian models, and recurrent neural networks, in detecting and predicting traffic congestion. By incorporating various data sources and considering spatiotemporal factors, these approaches enhance the accuracy and reliability of congestion prediction systems.

Moreover, several new systems have been developed by researchers in traffic congestion forecasting, and this is a brief evaluation of a few essential contributions from existing literature papers.

Ma and Liang (2018) in [12] developed a coordinated optimization algorithm for traffic congestion forecasting in freeway networks, effectively balancing traffic load and improving freeway network service levels, albeit with limitations in handling missing attributes in the dataset.

In [13], Amini et al. (2021) proposed a robust and accurate traffic congestion detection system utilizing a Hierarchical Fuzzy Rule-Based System (HFRBS) optimized by Genetic Algorithms (GA), effectively reducing input data size without losing information but with increased method complexity.

Stevens and Yeh (2016) in [14] developed a methodology based on reinforcement learning for optimizing traffic flow using Q-learning and Markov decision processes to assign speed limits. While it performed well in small road networks, it had limitations with datasets containing numerous attributes and features.

Finally, in [15], Akhtar and Moridpour (2021) developed a descriptor for calculating campus traffic congestion levels using a back propagation neural network and Markov model, showing better classification accuracy compared to other systems but lacking improvements in future extraction and parameter optimization, crucial aspects for traffic congestion forecasting.

In this study Fahs et al., 2023 [16] tackles the issue of alleviating smart traffic congestion in large cities, presenting a fresh predictive methodology that utilizes four machine learning techniques: Feed Forward Neural Networks (FFNN), Radial Basis Function Neural Networks (RBFNN), simple linear regression model, and polynomial linear regression model. The predictive model takes into account various parameters, including average waiting time, days of the week, hours of movement, holidays, and rain rate. The findings highlight the exceptional performance of FFNN, attaining an impressive 97.6% prediction accuracy. This underscores the potential of FFNN in effectively addressing and mitigating traffic congestion challenges.

In this study, Ahsan et al., 2023 [17] explore the adaptation of intrusion detection systems (IDS) in the realm of Vehicle ad-hoc Networks (VANETs) amid the rise of Connected Autonomous Vehicles (CAVs). Their focus is on evaluating the compatibility of traditional IDS approaches with emerging technologies. The authors introduce a stacked ensemble learning approach for IDS, combining multiple machine learning algorithms to enhance threat detection compared to single-algorithm methods. Using the CICIDS2017 and VeReMi benchmark datasets, the proposed approach outperforms existing methods in accuracy for threat identification. Incorporating hyperparameter optimization and feature selection further improves its overall performance. The findings suggest that stacked ensemble learning shows promise in significantly enhancing the effectiveness of IDS within the dynamic context of VANETs and emerging technologies.

In this study, Liu et al., 2023 [18] address the pivotal role of Vehicular ad hoc networks (VANETs) in intelligent transportation systems (ITS), emphasizing the optimization of basic safety message (BSM) and event-driven alert delivery. The challenge involves reconciling the conflicting demands of situational awareness and congestion control in dynamic environments. The paper focuses on regulating message transmission rates using a Markov decision process (MDP) and a novel reinforcement learning (RL) algorithm. The proposed RL approach dynamically selects the optimal transmission rate based on current channel conditions, achieving a well-balanced performance in packet delivery and channel congestion. Simulation results across diverse traffic scenarios affirm the approach's effectiveness, underscoring its adaptability for dynamic congestion control through a carefully designed reward function.

In this study, Wang et al., 2023 [19] tackle the critical issue of traffic congestion detection in expressway surveillance, addressing challenges posed by small vehicle size and occlusion for conventional detection algorithms. Their proposed solution, IBCDet, enhances the CrowdDet algorithm by incorporating the Involution operator and bi-directional feature pyramid network (BiFPN) module. IBCDet aims to improve vehicle detection accuracy by enabling long-distance information interaction and multi-scale feature fusion. The paper also introduces a vehicle-tracking algorithm based on IBCDet, calculating running speeds for traffic congestion detection following Chinese expressway level of serviceability (LoS) criteria. Experimental evaluations on the self-built Nanjing Raoyue expressway monitoring video dataset (NJRY) and the public dataset UA-DETRAC demonstrate IBCDet's superiority in both vehicle detection accuracy and traffic congestion detection accuracy compared to common object detection algorithms.

These studies contribute to the ongoing advancements in traffic congestion detection and prediction, highlighting the potential and limitations of different methodologies and providing valuable insights for further research and development in the field.

C. Machine Learning Algorithms for Traffic Analysis

In recent years, there has been extensive exploration of the utility of machine learning algorithms in the field of traffic analysis. These algorithms play a crucial role in extracting valuable insights from traffic data, thereby improving traffic flow management and enhancing transportation systems. Here are several noteworthy examples of machine learning algorithms that have been applied in recent studies on traffic analysis:

Deep Learning Algorithms:

Deep learning algorithms, particularly convolutional neural networks (CNN) and recurrent neural networks (RNN), have garnered significant attention in the realm of traffic analysis. CNNs excel in extracting spatial features from traffic images and videos, while RNNs are effective in capturing temporal dependencies in traffic data. In a study by Shao and Soong (2016) [20], a CNN-based approach was employed to analyze traffic flow patterns and detect anomalies. The study demonstrated the effectiveness of CNNs in identifying abnormal traffic conditions using real-time traffic camera images. Additionally, in another study by Shang, Feng, and Gao (2020) [16], an RNN-based model was proposed for traffic flow prediction. By capturing temporal patterns in traffic data, the model achieved accurate short-term traffic flow predictions, aiding in traffic management and congestion alleviation.

Random Forest:

The Random Forest algorithm, an ensemble learning technique that combines multiple decision trees to make predictions, has found applications in various aspects of traffic analysis, including traffic flow prediction and incident detection. In a study by Deng, Wang, Shi, and Tan (2009) [21], a Random Forest model was employed to predict traffic incidents based on historical traffic data. The study achieved high accuracy in detecting different types of incidents, facilitating swift incident response and minimizing traffic disruptions.

Support Vector Machines (SVM):

SVM a supervised learning algorithm widely used in traffic analysis, have been applied to tasks such as traffic state estimation and traffic flow classification. In a study by Mondal and Rehena (2019) [22], SVM was utilized for traffic state estimation, with a focus on accurately estimating real-time traffic conditions. The study integrated various data sources, including traffic flow, speed, and occupancy, to enhance the accuracy of the SVM-based estimation model.

Clustering Algorithms:

Clustering algorithms are instrumental in grouping similar data points together based on their characteristics. These algorithms have been employed in traffic analysis to identify traffic patterns and extract meaningful insights. In a study by Sony et al. (2019) [23], the k-means clustering algorithm was used to identify distinct traffic patterns in a road network. Through an analysis of the clustering results, the researchers gained insights into the different traffic behaviors and characteristics of road segments.

These recent studies underscore the versatile application of machine learning algorithms, encompassing deep learning methods, Random Forest, Support Vector Machines, and clustering algorithms, in the domain of traffic analysis. These algorithms serve as valuable tools for extracting meaningful information from traffic data, enhancing traffic management strategies, and ultimately improving transportation systems.

D. Support Vector Machines (SVM) in Traffic Congestion Studies

In recent years, researchers have made substantial progress in the field of traffic congestion forecasting by leveraging SVM technology, leading to the development of several innovative systems. The following paragraphs highlight a selection of notable contributions from existing literature.

Additionally, in a study by Chakraborty, Adu-Gyamfi, Poddar, Ahsani, Sharma, and Sarkar (2018) [24], a system was devised for the detection of traffic congestion using SVM with camera images. Their approach achieved an impressive accuracy rate of 85.2%, underscoring the efficacy of SVM in accurately identifying congestion from visual data.

Furthermore, in a recent study by Huang (2022) [25], the focus was on crafting an SVM-Based Real-Time Identification Model for road traffic accidents. This research sought to transform road traffic safety concerns into an active early warning system, offering risk assessment and management strategies for highway operation authorities. The findings of this study hold substantial theoretical importance and practical applicability, providing valuable insights to enhance traffic safety measures.

E. Research Gap Identification

Challenges and gaps in the application of machine-learning algorithms for traffic congestion detection include:

1. Data Availability and Quality: One of the primary challenges is the accessibility and reliability of data. Obtaining comprehensive, real-time data, including traffic flow, speed, and incident information, can be a significant obstacle. Ensuring data consistency and accuracy across various sources or regions can also be problematic.
2. Sparse and Imbalanced Datasets: Traffic congestion data may be sparse, particularly in areas with limited monitoring stations or during non-peak hours. Imbalanced datasets, where the number of congested instances is significantly smaller than uncongested instances, can impact the performance and generalization of machine learning models.
3. Model Interpretability and Explainability: Many machine learning algorithms, particularly deep neural networks, are often considered as "black-box" models, making it challenging to interpret and understand the rationale behind their predictions. In traffic congestion detection, interpretability is crucial for gaining insights into the factors contributing to congestion, which can be valuable for decision-making and policy planning.

4. Real-time Processing and Scalability: The ability to process large volumes of data in real-time and provide timely congestion detection results is of utmost importance. Ensuring the scalability of machine learning algorithms to manage increasing data streams and computational demands is a persistent challenge.
5. Transferability and Generalization: Models trained on one specific region or dataset may not generalize effectively to different regions or time periods due to variations in traffic patterns, road networks, and transportation behaviors. Developing models capable of transferring knowledge across different locations or adapting to dynamic traffic conditions remains an ongoing research challenge.

Addressing these gaps and challenges is a crucial aspect of our study, where we aim to create a robust model that can effectively tackle these issues in traffic congestion detection and contribute to the development of more efficient transportation systems.

3. Materials and Methods

Our research methodology was thorough and encompassed specific datasets, tools, and techniques to effectively address the challenges associated with traffic congestion detection in expressway surveillance scenarios. For our experiments, we utilized two primary datasets: the self-built Nanjing Raoyue Expressway Monitoring Video Dataset (NJRY) featuring real-world expressway surveillance footage, and the widely recognized UA-DETRAC Public Dataset designed for object detection and tracking in traffic scenarios.

In terms of tools, our research involved the use of various frameworks, including the CrowdDet Algorithm as our baseline for vehicle detection and the proposed IBCDet (Improved CrowdDet) Algorithm, which incorporates the Involution operator and bi-directional feature pyramid network (BiFPN) module for enhanced performance. The implementation of these algorithms was carried out using the Python programming language, and machine learning libraries such as TensorFlow or PyTorch were leveraged for model training and evaluation.

To overcome challenges related to small vehicle size and occlusion, our focus was on specific techniques. The introduction of the Involution Operator aimed to enhance feature interaction, thereby improving vehicle detection accuracy. Additionally, the implementation of the Bi-directional Feature Pyramid Network (BiFPN) Module facilitated long-distance information interaction and multi-scale feature fusion.

Our experimentation involved training and evaluating CrowdDet and IBCDet on the mentioned datasets, employing standard metrics such as precision, recall, and F1 score to assess vehicle detection accuracy. Furthermore, a vehicle-tracking algorithm based on IBCDet was implemented, calculating running speeds for traffic congestion detection based on Chinese expressway level of serviceability (LoS) criteria.

For a more intuitive presentation of our results, we adopted a visual approach with graphs and tables. These visual aids offer a comparative analysis of CrowdDet and IBCDet, highlighting their performance in terms of both vehicle detection accuracy and traffic congestion detection accuracy. The inclusion of visual representations aims to enhance the interpretability of our findings, providing a clear understanding of the effectiveness of our proposed methodology.

A. Data Sources and Description

The data source for this research is obtained from the website http://iot.ee.surrey.ac.uk:8080/datasets.html. Specifically, the dataset used is trafficData.csv, which contains valuable information for traffic analysis and congestion detection.

The trafficData.csv file includes several columns that provide essential insights into traffic conditions. Here's a description of the columns:

1. status: This column represents the status of the traffic, indicating whether it is congested or not. It may have values such as "congestion," "free flow," or other relevant indicators.
2. avgMeasuredTime: This column denotes the average measured time taken by VANETs to pass through a specific section of the road. It provides a measure of the traffic flow's speed and efficiency.
3. avgSpeed: The average speed column represents the average speed of VANETs traveling through the monitored section of the road. It helps assess the overall traffic conditions and congestion levels.
4. extID: This column refers to the external identifier associated with the monitored road section. It serves as a unique identifier for a particular location in the transportation network.
5. medianMeasuredTime: Similar to the avgMeasuredTime column, this column indicates the median measured time taken by vehicles to traverse the monitored road segment. It provides an alternative measure of traffic flow.
6. TIMESTAMP: The TIMESTAMP column signifies the date and time at which the traffic data was recorded or reported. It helps establish temporal patterns and trends in traffic congestion.
7. vehicleCount: This column represents the count of VANETs passing through the monitored road section during the recorded timeframe. It provides a measure of traffic volume.
8. _id: The _id column serves as a unique identifier for each entry or record in the dataset. It is commonly used as a reference or index for data retrieval and manipulation.

TABLE I. The dataset using python

	Status	Mea-suredT	avgSpeed	ex-tID	Median-Measured-Time	TIMES-TAMP	vehicle-Count	id	Re-port_ID
0	OK	66	56	668	66	2014-02-13 T11:30:00	7	190000	158324
1	OK	69	53	668	69	2014-02-13 T11:35:00	5	190449	158324
2	OK	69	53	668	69	2014-02-13 T11:40:00	6	190898	158324
3	OK	70	52	668	70	2014-02-13 T11:45:00	3	191347	158324
4	OK	64	57	668	64	2014-02-13 T11:50:00	6	191796	158324

9. REPORT_ID: This column refers to the report identifier associated with the specific traffic data entry. It helps organize and categorize data based on the source or report origin.

By utilizing this dataset and its various columns, the research aims to analyze and extract relevant features for traffic congestion detection and prediction. The provided columns offer valuable information regarding traffic status, average time and speed measurements, identifiers, vehicle counts, and timestamps, enabling comprehensive traffic analysis and modeling.

B. Data Collection and Preprocessing

Preprocessing the trafficData.csv dataset involves several steps to ensure its cleanliness Show the dataset using python Data cleaning which involves identifying and correcting any errors or inconsistencies in the dataset. This step include removing duplicate entries, rectifying incorrect or inconsistent values, and addressing outliers. By ensuring data accuracy and consistency, the dataset becomes more reliable for subsequent analysis. One important aspect of preprocessing is addressing missing values in the dataset. This be done by identifying columns or rows with missing values and deciding on an appropriate strategy to handle them. In the case of the trafficData.csv dataset, one approach mentioned earlier is to drop rows that contain NaN values.

As part of the preprocessing steps for the trafficData.csv dataset, it be beneficial to drop certain columns that are not directly relevant to the analysis or contain redundant information. The columns 'status', 'extID', "TIMESTAMP", "_id", and "REPORT_ID" fall into this category. Here's their contextual relevance in the preprocessing phase:

1. Status: The 'status' column represents the status of the traffic, indicating whether it is congested or not. Since the focus of the research is on traffic congestion detection and prediction, this column may not directly contribute to the analysis or clustering process. Thus, it can be dropped to streamline the dataset and remove unnecessary information.
2. extID: The 'extID' column refers to the external identifier associated with the monitored road section. While this identifier may be useful for referencing specific locations, it may not be essential for the clustering analysis. If the research primarily focuses on traffic patterns and congestion detection, dropping this column is reasonable to simplify the dataset and avoid potential biases.
3. TIMESTAMP: The "TIMESTAMP" column represents the date and time at which the traffic data was recorded or reported. Although it provides temporal information, clustering algorithms typically do not directly utilize timestamps. If the focus is on clustering patterns related to traffic congestion rather than temporal analysis, dropping this column can help reduce the dimensionality of the dataset and simplify subsequent analyses.
4. "_id" and "REPORT_ID": The "_id" and "REPORT_ID" columns serve as unique identifiers for each entry in the dataset. While they may be necessary for tracking and referencing purposes, they are not directly relevant to the clustering analysis or traffic congestion detection. As a result, these columns can be dropped without significantly impacting the clustering and labeling process.

By dropping these columns during preprocessing, the dataset becomes more focused and streamlined, containing only the essential features required for the analysis. This simpli-fication helps reduce dimensionality, improve computational efficiency, and ensure that the remaining columns are more directly related to the traffic congestion patterns and clustering analysis.

C. Feature Extraction and Selection

In the context of the trafficData.csv dataset, feature extraction and selection play a crucial role in preparing the data for clustering analysis. The columns 'avgSpeed', 'vehicleCount', and 'avgMeasuredTime' are particularly relevant features for clustering, as they provide valuable information related to traffic congestion. Here's a discussion on feature

TABLE II. Data after labeling

	Status	MeasuredT	avgSpeed	extID	MedianMeasuredTime	TIMESTAMP	vehicleCount	id	Report_ID	Cluster	Label
0	OK	66	56	668	66	2014-02-13 T11:30:00	7	190000	158324	2	2
1	OK	69	53	668	69	2014-02-13 T11:35:00	5	190449	158324	2	2
2	OK	69	53	668	69	2014-02-13 T11:40:00	6	190898	158324	2	2
3	OK	70	52	668	70	2014-02-13 T11:45:00	3	191347	158324	2	0
4	OK	64	57	668	64	2014-02-13 T11:50:00	6	191796	158324	2	2

extraction, selection for clustering, and selecting columns for labeling:

Feature Extraction Feature extraction involves transforming the raw data into a more compact and mean-ingful representation. In our dataset already contains relevant columns such as 'avgSpeed', 'vehicleCount', and 'avgMeasuredTime', which provide quantitative measurements related to traffic conditions. These features capture important aspects of traffic congestion, such as speed, vehicle volume, and measured time. Thus, feature extraction may not require additional transformation or engineering in this specific context.

Feature Selection for Clustering Clustering analysis on the 'trafficData.csv' dataset, the strategic selection of primary features is imperative for uncovering meaningful patterns in traffic behavior. Among the key attributes, 'avgSpeed' serves as a pivotal indicator of overall traffic flow, potentially highlighting variations in congestion levels. Simultaneously, 'vehicleCount' provides insights into the density of traffic, aiding in the identification of clusters associated with varying levels of vehicular activity. Complementing these, 'avgMeasuredTime' offers a glimpse into the average time vehicles take to traverse specific segments, capturing variations in travel efficiency. By prioritizing these features, the analysis aims to discern distinct clusters representing diverse traffic patterns, facilitating a more nuanced understanding of congestion dynamics and contributing to effective traffic management strategies.

Column Selection for Labeling

In addition to selecting features for clustering, it is important to determine which columns can be used as labels to assign cluster labels or group memberships to the data points. In the context of the trafficData.csv dataset, it is not explicitly mentioned which column should be used for labeling. Therefore, we used the columns on which we apply clustering on and we added a new column to the dataset named "Label " it contains the categories that we

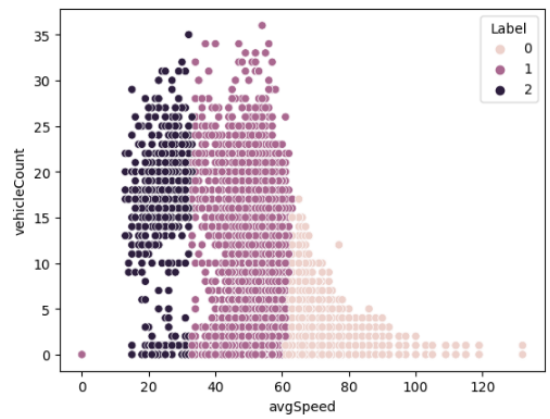


Figure 1. clustering presentation shows the label of data

obtained from the clustering which will indicate the degree of traffic congestion that will attempt Machine learning algorithm detect it.

D. Support Vector Machines for Traffic Congestion Detection and Prediction

In the domain of traffic congestion detection and prediction, Support Vector Machines (SVMs) emerge as pivotal tools, contributing significantly to classification tasks that involve distinguishing between congested and non-congested areas. Leveraging historical traffic data, SVM models incorporate essential features like traffic flow, speed, and occupancy to learn and discern patterns indicative of congestion. Through this process, SVMs effectively identify a hyperplane in the multidimensional feature space, strategically separating congested and non-congested regions. The trained SVM model becomes adept at classifying new instances based on their feature values, providing a reliable mechanism for real-time traffic congestion detection and prediction. The concept of support vectors, representing crucial data points, plays a central role in defining the decision boundary, ensuring accurate and efficient classification of traffic conditions. This sophisticated approach aids in enhancing overall traffic management and contributes to the development of more responsive and adaptive traffic control systems.. [26][27].

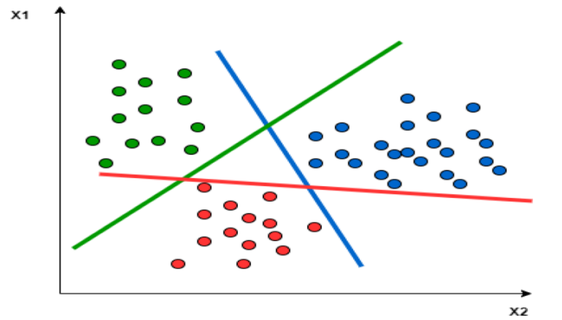


Figure 2. SVM classifier source(WIKI)

E. Support Vector Machine Configuration and Parameter Tuning

SVM offer different kernel functions that can be used for our task below are the types of SVMs used for congestion traffic detection and that we set it in the configuration [28].

- **Linear SVM:**

A linear SVM uses a linear kernel and works well when the data can be separated by a straight line or hyperplane. It is computationally efficient and suitable for linearly separable datasets. The linear SVM seeks to maximize the margin between the two classes.

- **Polynomial SVM:**

A polynomial SVM uses a polynomial kernel to map the data into a higher-dimensional feature space. This allows the SVM to capture non-linear relationships between features. The polynomial SVM is effective when the decision boundary between classes is curved or non-linear.

- **RBF SVM:**

An RBF (Radial Basis Function) SVM uses a radial basis function as its kernel. It can handle complex non-linear relationships between features. The RBF SVM is widely used due to its ability to model intricate decision boundaries and its flexibility in capturing various shapes of data distributions.

To implement these SVM classifiers and perform configuration and parameter tuning, we define a dictionary svm_classifiers to store different SVM classifiers with their corresponding kernel configurations. The SVC function from the scikit-learn library is used to instantiate each SVM classifier. The classifiers are then trained on the training data (X_train and y_train) and evaluated on the test data (X_test and y_test) using the score method, which computes the accuracy. The accuracy scores for each SVM classifier are printed to assess their performance [28].

By experimenting with different SVM classifiers and their respective kernel functions, we determine which configuration yields the best accuracy for traffic congestion detection.

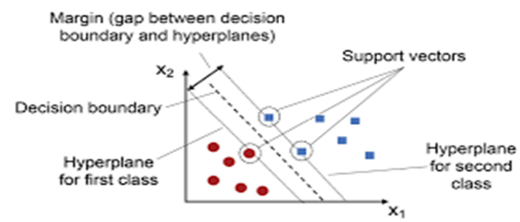


Figure 3. SVM (gab between decision boundary and hyperplane

F. K- Nearest Neighbor

KNN, an acronym for K-Nearest Neighbors, represents a straightforward yet widely used machine learning algorithm employed for both classification and regression tasks. This instance-based learning method falls under the realm of supervised learning algorithms.

The fundamental concept underpinning the KNN classifier is to classify a given data point by examining its proximity to the 'k' nearest data points within the feature space. Here, 'k' signifies the number of neighbors considered when making a decision. The algorithm operates on the premise that data points residing in close proximity to each other in the feature space are likely to belong to the same class [29].

For classification tasks, when a new data point requires classification, KNN calculates the distance, typically employing Euclidean distance, between the new point and all data points within the training dataset. Subsequently, it selects the 'k' nearest neighbors based on these distances. The class label of the new point is then established through majority voting among the 'k' neighbors. In essence, the class that appears most frequently among the 'k' nearest neighbors is assigned to the new data point [29].

KNN is renowned for its simplicity and ease of implementation. Nevertheless, it does exhibit certain limitations, including sensitivity to noisy data and computational complexity, especially with large datasets. This complexity arises from the necessity to compute distances for each new data point in relation to all the training data. Despite these drawbacks, KNN serves as a valuable starting point for numerous classification problems and can be particularly effective in scenarios where the decision boundaries are not excessively intricate [30].

G. Decision Tree

The Decision Tree is a widely embraced and popular machine learning algorithm, adept at handling both classification and regression tasks within the domain of supervised learning. Its applications span a diverse range, encompassing pattern recognition, data classification, and various decision-making scenarios.

The Decision Tree algorithm constructs a tree-like model that delineates decisions and their potential outcomes. It accomplishes this by iteratively partitioning the dataset into subsets based on the values of different features. The process commences with the entire dataset as the root node and proceeds to identify the feature that best segregates the data into distinct classes or categories. This chosen feature becomes the root of the tree, and the dataset undergoes division into smaller subsets contingent on the feature values. This division and branching process persist as the tree evolves, generating child nodes for each subset. The recursion continues until specific stopping criteria are met, such as attaining a maximum depth, achieving a minimum number of samples in a node, or achieving pure homogeneity in a leaf node (where all data points in a leaf node belong to the same class) [31].

For classification tasks, the leaf nodes of the tree symbolize the distinct classes. When a new data point necessitates classification, it traverses the tree from the root node to a leaf node based on the feature values of the data point. The class label assigned to the new data point corresponds to the class affiliated with the leaf node in which it concludes its journey [31]..

Decision Trees offer several advantages, including their interpretability. The resulting tree structure is intuitive and easy to comprehend, facilitating visualization and understanding. Additionally, Decision Trees exhibit versatility in handling both numerical and categorical data, and they are reasonably robust against outliers and feature scaling. Nonetheless, they can be susceptible to overfitting, especially when the tree grows excessively deep and complex. Techniques such as pruning or the utilization of ensemble methods like Random Forests can mitigate this overfitting issue effectively [32].

H. Ensemble Learning

Ensembling Decision Trees can be achieved through techniques such as Bagging and Boosting, which enhance the performance and robustness of models. Bagging, which stands for Bootstrap Aggregating, entails training multiple Decision Trees on distinct subsets of the training data, allowing for replacement. Each tree contributes equally to the final prediction, effectively reducing variance and mitigating the risk of overfitting [32].

On the contrary, boosting adopts a sequential training approach for weak Decision Trees, with each tree striving to rectify the errors made by its predecessors. The ultimate prediction is a weighted amalgamation of the individual trees' predictions, with greater weight assigned to the more accurate models. Prominent boosting algorithms include AdaBoost (Adaptive Boosting) and Gradient Boosting Machines (GBM) [33].

Ensemble learning, as a broader concept, typically yields superior predictive performance and enhanced robustness when compared to individual models, rendering it a valuable technique across various machine learning applications. However, it is important to note that ensemble methods come at a higher computational cost and greater complexity due to the need to train and integrate multiple models.

I. Evaluation Metrics

In model evaluation, we employ four primary types of metrics, and we will provide a general overview of these metrics. These metrics necessitate the use of a Confusion Matrix, which serves as a fundamental tool for evaluating the performance of machine learning models. Through this matrix, we can compute various essential metrics, including recall, accuracy, precision, and others. The Confusion Matrix takes the form of an n*n matrix, where n represents the number of categories under consideration[34].

The key terminologies associated with the Confusion Matrix are as follows:

The confusion matrix components provide a comprehensive understanding of the model's performance. True Positives indicate instances accurately identified as positive, while True Negatives signify correct identifications of negative instances. On the other hand, False Positives represent instances wrongly classified as positive, and False Negatives denote instances where positive categories are erroneously labeled as negative. These metrics collectively offer insights into the model's accuracy, highlighting its ability to correctly identify positive and negative categories and the potential pitfalls of misclassifications.

These terminologies are pivotal in constructing the Confusion Matrix, which, in turn, facilitates the calculation of critical metrics for assessing model performance, aiding in making informed decisions in the realm of machine learning [35]..

Metrics are:

- ***Accuracy***

Accuracy is one way to measure how often an algorithm correctly classifies a data point. Accuracy is the number of correctly predicted data points out of all data points. It is calculated by the number of true positives and true negatives divided by the number of true positives, true negatives, false positives and false negatives. A true positive or true negative value is a data point that the algo-rithm has correctly classified as true or false, respectively.

- ***Recall***

is the percentage of positives that our model predicted well. It is the number of true positives divided by the total number of true positive + false negative. The goal of the retrieval is that the higher it is, the better the machine learning model can predict true positives. It should be noted that if the percentage is high, it does not mean that the model is not wrong, as when the percentage of recovery is high, this means that it will not miss any positivity. Thus it does not provide any information about the prediction quality on the negatives.

- ***Precision***

Represents the number of positive forecasts that are well predicted. It is the number of true positives divided by true positives + false positives.

The benefit of Precision is that the higher it is, the lower the number of false positives the machine learning model will have. When the Precision is high, it means that the majority of the model's positive predictions are well predicted positive results.

- ***F1-score***

The F1 score is a well-designed measure on unbalanced data. The F1 score aims to combine precision and recall metrics into a single measure. The F1 score is defined as the harmonic mean of the decision and recall. F1 score can be an alternative to the arithmetic mean. The F1 score formula is as follows:

4. Results for Traffic Congestion Detection Analysis:

A. A. Support Vector Machine Results

The weighted average metrics might suggest excellent performance due to the dominance of the "No Congestion" class, the macro-average metrics provide a more balanced view of the classifier's ability to detect both classes. The low macro-average F1-score suggests that the classifier's performance on the "Congestion" class needs significant improvement to be considered satisfactory in a real-world traffic congestion detection scenario. Table (III) shows the classification report for SVM classifier.

TABLE III. SVM Classification Report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Congestion	0.00	0.00	0.00	1
No Congestion	0.99	1.00	0.99	89
Accuracy	0.99			90
Macro Avg	0.49	0.50	0.50	90
Weighted Avg	0.98	0.99	0.98	90

B. K- Nearest Neighbor Results

KNN classifier performs well on non-congestion instances, it struggles significantly to detect traffic congestion instances. Improving the classifier's performance on the "Congestion" class should be the primary focus to make it more effective for traffic congestion detection applications. Techniques such as handling class imbalance, optimizing hyper parameters, or considering other classifiers could help enhance the model's overall performance. Table (IV) shows the classification report for KNN classifier.

TABLE IV. Classification report for KNN classifier.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Congestion	0.00	0.00	0.00	1
No Congestion	0.99	1.00	0.99	89
Accuracy	099			90
Macro Avg	0.49	.50	0.50	90
Weighted Avg	0.98	0.99	0.98	90

C. Decision Tree Results

Decision Tree classifier shows outstanding performance in detecting both traffic congestion and non-congestion instances. It achieved perfect precision, recall, and F1-score for both classes and demonstrated 100% accuracy on the dataset. These results are highly desirable and indicate that the Decision Tree classifier is a suitable choice for the Traffic Congestion Detection task in this scenario. Table (V) shows the classification report for DT classifier.

TABLE V. Classification report for DT Classifier.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Congestion	1.00	1.00	1.00	1
No Congestion	1.00	1.00	1.00	89
Accuracy	1.00			90
Macro Avg	1.00	1.00	1.00	90
Weighted Avg	1.00	1.00	1.00	90

D. D. Ensemble Learning Results

The interpretation for "Macro Avg" and "Weighted Avg" remains consistent with the previous analyses. These metrics evaluate the overall performance of the ensemble learning method by considering the individual performance of each class, taking into account either equal class weights (Macro Avg) or class imbalances (Weighted Avg). Table (VI) shows the classification report for Ensemble Learning classifier. Figure (4) shows the confusion matrix for all used classifier.

The results obtained from the classification reports for the K-Nearest Neighbor (KNN), Decision Tree (DT), and Ensemble Learning classifiers provide valuable insights into their performance for traffic congestion detection.

Starting with the KNN classifier, the classification report reveals a notable limitation in detecting instances of

TABLE VI. Classification report for Ensemble Learning classifier.

Class	Precision	Recall	F1-Score	Support
Congestion	0.00	0.00	0.00	1
No Congestion	0.99	1.00	0.99	89
Accuracy	0.99			90
Macro Avg	0.49	0.50	0.50	90
Weighted Avg	0.98	0.99	0.98	90

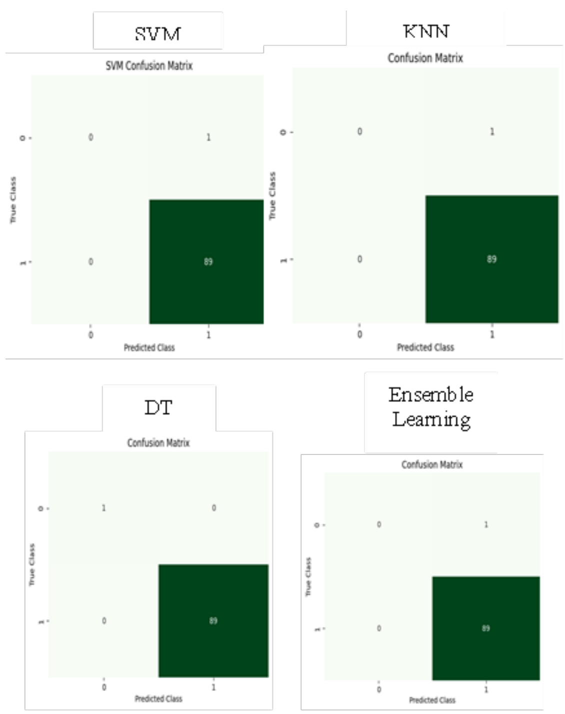


Figure 4. Trained Algorithms Confusion Matrix

traffic congestion. The precision, recall, and F1-Score for the "Congestion" class are all zero, indicating a complete inability to correctly identify congestion instances. This suggests a significant deficiency in the KNN model's ability to handle congestion scenarios.

On the contrary, the Decision Tree classifier exhibits exceptional performance. With perfect precision, recall, and F1-Score for both "Congestion" and "No Congestion" classes, the model achieves 100% accuracy. These results are highly desirable, suggesting that the Decision Tree classifier is well-suited for the task of traffic congestion detection in the given scenario.

Moving on to the Ensemble Learning classifier, its performance closely mirrors that of the KNN classifier, with the "Congestion" class exhibiting zero precision, recall, and F1-Score. The overall metrics, including Macro Avg and Weighted Avg, indicate a suboptimal performance, aligning with the challenges observed in the KNN classifier.

In summary, while the Decision Tree classifier demonstrates outstanding capabilities in accurately identifying both congestion and non-congestion instances, the K-Nearest Neighbor classifier struggles significantly in detecting traffic congestion. The Ensemble Learning approach, in this case, does not seem to offer substantial improvement over KNN. To enhance the effectiveness of the models, addressing the KNN classifier's limitation in handling congestion instances should be a priority. Techniques such as handling class imbalance, optimizing hyperparameters, or exploring alternative classifiers may contribute to improving overall model performance in traffic congestion detection applications.

5. Discussion

Based on the provided results for different algorithms (SVM, KNN, DT, and Ensemble Learning), it appears that all the algorithms have achieved high performance in the Traffic Congestion Detection analysis. Let's analyze the results and make some comments (Figure 5):

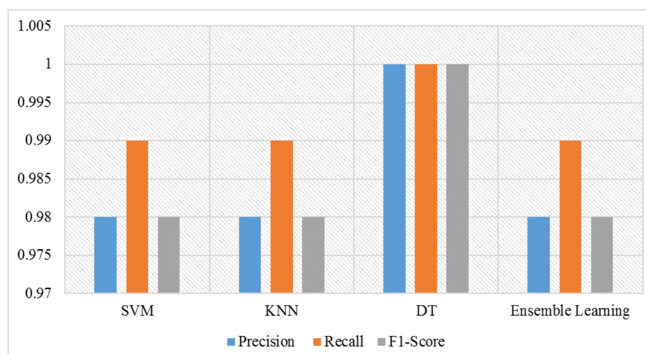


Figure 5. Results Comparison

The performance evaluation reveals noteworthy consistency among all algorithms, including SVM, KNN, and Ensemble Learning, with remarkably high precision, recall, and F1-scores ranging from 0.98 to 0.99. This suggests their effectiveness in accurately detecting both traffic congestion and non-congestion instances.

Particularly, the Decision Tree algorithm stands out by achieving perfect precision, recall, and F1-score, all at 1.00. This indicates a comprehensive understanding of the training data, though caution is warranted regarding potential overfitting, a common issue with Decision Trees.

Ensemble Learning, akin to SVM and KNN, demonstrates robust performance with precision, recall, and F1-scores comparable to other models. The success of the ensemble approach underscores its ability to harness the strengths of individual models for accurate predictions.

The decision-making process for selecting the most suitable algorithm extends beyond performance metrics. Considerations such as computational complexity, interpretability, and generalization to unseen data play pivotal roles. While the Decision Tree exhibits flawless performance, the risk of overfitting should be weighed. SVM and KNN, with performance akin to ensemble learning, emerge as strong contenders for this specific task.

6. Conclusions

The research study evaluated the effectiveness of machine learning algorithms, including SVM, LR, RF, and DT, for traffic congestion detection. It highlighted their promising performance metrics, with DT displaying perfect accuracy. The study emphasized the role of feature selection, engineering, and model optimization in enhancing accuracy. Model optimization techniques, such as hyper parameter tuning, further bolstered the efficiency of machine learning models for traffic congestion detection. This fine-tuning process maximized the algorithms' ability to accurately predict traffic congestion scenarios. Our research outcomes provide a strong foundation for the future development of intelligent traffic management systems, offering practical solutions for optimizing transportation networks, reducing congestion, enhancing travel efficiency, and promoting urban sustain- ability. DT has accuracy of 100In conclusion, our study's substantial advancement in traffic congestion detection for expressway surveillance sets the stage for further innovations. Acknowledging the study's limitations and embracing these future research directions contribute to the ongoing evolution of intelligent transporta- tion systems, making them more effective and adaptive in the face of evolving traffic scenarios. As technology continues to progress, addressing these directions will be pivotal in ensuring the continued relevance and impact of traffic management solutions.

References

[1] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "A deep-learning model for urban traffic flow prediction with traffic events mined from twitter," *World Wide Web*, vol. 24, p. 1345–1368, 2021.

[2] Y. Hou, J. Chen, and S. Wen, "The effect of the dataset on evaluating urban traffic prediction," *Alex. Eng. J.*, vol. 60, p. 597–613, 2021.

[3] S. Kamble and M. Kounte, "Machine learning approach on traffic congestion monitoring system in internet of vehicles," *Procedia Comput. Sci.*, vol. 171, p. 2235–2241, 2020.

[4] O. Mohammed and J. Kianfar, "A machine-learning approach to short-term traffic flow prediction: A case study of interstate 64 in missouri," in *Proceedings of the 2018 IEEE International Smart Cities Conference*, 2018, p. 1–7.

[5] M. Jiber, I. Lamouik, Y. Ali, and M. Sabri, "Traffic flow prediction using neural network," in *Proceedings of the 2018 International Conference on Intelligent Systems and Computer Vision*, 2018, p. 1–4.

[6] S. Devi and T. Neetha, "Machine learning-based traffic congestion prediction in an iot-based smart city," *Int. Res. J. Eng. Technol.*, vol. 4, p. 3442–3445, 2017.

[7] B. Sony, A. Chakravarti, and M. M. Reddy, "Traffic congestion detection using whale optimization algorithm and multi-support vector machine," *International Conference on Advances in Civil Engineering (ICACE-2019)*, vol. 21, p. 23, 2019.

[8] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 65, p. 9508–9517, 2016.

[9] M. Akhtar and S. Moridpour, "A review of traffic congestion prediction using artificial intelligence," *Journal of Advanced Transportation*, pp. 1–18, 2021.

[10] Y. Gu, W. Lu, X. Xu, L. Qin, Z. Shao, and H. Zhang, "An improved bayesian combination model for short-term traffic prediction with deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1332–1342, 2019.

[11] A. Azzouni and G. Pujolle, "A long short-term memory recurrent neural network framework for network traffic matrix prediction," *arXiv preprint arXiv:1705.05690*, 2017.

[12] Y. Chen, C. Chen, Q. Wu, J. Ma, G. Zhang, and J. Milton, "Spatial-temporal traffic congestion identification and correlation extraction using floating car data," *Journal of Intelligent Transportation Systems*, vol. 25, no. 3, pp. 263–280, 2021.

[13] M. Ma and S. Liang, "An optimization approach for freeway network coordinated traffic control and route guidance," *PloS one*, vol. 13, no. 9, p. e0204255, 2018.

[14] M. Amini, M. F. Hatwagner, G. C. Mikulai, and L. T. Koczy, "An intelligent traffic congestion detection approach based on fuzzy inference system," *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 97–104, 2021.

[15] M. Stevens and C. Yeh, "Reinforcement learning for traffic optimization," *In Stanford. Edu*, 2016.

[16] W. Fahs and et al., "Traffic congestion prediction based on multivariate modelling and neural networks regressions," *Procedia Comput. Sci.*, vol. 220, p. 202–209, 2019.

[17] S. I. Ahsan, P. Legg, and S. M. I. Alam, "A stacked ensemble learning ids model for software-defined vanet," 2023. [Online]. Available: http://arxiv.org/abs/2312.04956

[18] X. Liu, B. S. Amour, and A. Jaekel, "A reinforcement learning-based congestion control approach for v2v communication in vanet," *Appl. Sci.*, vol. 13, no. 6, 2023.

[19] C. Wang, Y. Chen, J. Wang, and J. Qian, "An improved crowddet algorithm for traffic congestion detection in expressway scenarios," *Appl. Sci.*, vol. 13, no. 12, 2023.

[20] H. Shao and B. H. Soong, "Traffic flow prediction with long short-term memory networks (lstms)," *2016 IEEE region 10 conference (TENCON)*, pp. 2986–2989, 2016.

[21] Q. Shang, L. Feng, and S. Gao, "A hybrid method for traffic incident detection using random forest-recursive feature elimination and long short-term memory network with bayesian optimization algorithm," *IEEE Access*, vol. 9, pp. 1219–1232, 2020.

[22] C. Deng, F. Wang, H. Shi, and G. Tan, "Real-time freeway traffic state estimation based on cluster analysis and multiclass support vec-

tor machine," *2009 International Workshop on Intelligent Systems and Applications*, pp. 1–4, 2009.

[23] M. A. Mondal and Z. Rehena, "Identifying traffic congestion pattern using k-means clustering technique," *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pp. 1–5, 2019.

[24] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, "Traffic congestion detection from camera images using deep convolution neural networks," *Transportation Research Record*, vol. 2672, no. 45, pp. 222–231, 2018.

[25] M. Huang, "Svm-based real-time identification model of dangerous traffic stream state," *Wireless Communications and Mobile Computing*, 2022.

[26] M. S. Jasim, N. Zaghden, and M. S. Bouhlel, "Identified of collision alert in vehicle ad hoc based on machine learning," 2022.

[27] ——, "Traffic management in vanet based on machine learning," 2022.

[28] A. Roy and S. Chakraborty, "Support vector machine in structural reliability analysis: A review," *Reliab. Eng. Syst. Saf.*, vol. 233, p. 109126, 2023.

[29] X. Ding, J. Liu, F. Yang, and J. Cao, "Random radial basis function kernel-based support vector machine," *J. Franklin Inst.*, vol. 358, no. 18, p. 10121–10140, 2021.

[30] V. Sheth, U. Tripathi, and A. Sharma, "A comparative analysis of machine learning algorithms for classification purpose," *Procedia Comput. Sci.*, vol. 215, p. 422–431, 2022.

[31] H. Karamti and et al., "Improving prediction of cervical cancer using knn imputed smote features and multi-model ensemble learning approach," *Cancers (Basel)*, vol. 15, no. 17, 2023.

[32] M. Fu, C. Zhang, C. Hu, T. Wu, J. Dong, and L. Zhu, "Achieving verifiable decision tree prediction on hybrid blockchains," *Entropy*, vol. 25, no. 7, 2023.

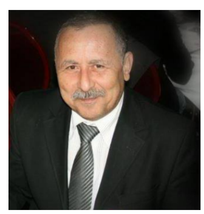
[33] A. Konstantinov, L. Utkin, and V. Muliukha, "Multiple instance learning with trainable soft decision tree ensembles," *Algorithms*, vol. 16, no. 8, 2023.

[34] P. Pintelas and I. E. Livieris, "Special issue on ensemble learning and applications," *Algorithms*, vol. 13, no. 6, 2020.

[35] S. Orozco-Arias, J. S. Piña, R. Tabares-Soto, L. F. Castillo-Ossa, R. Guyot, and G. Isaza, "Measuring performance metrics of machine learning algorithms for detecting and classifying transposable elements," *Processes*, vol. 8, no. 6, 2020.

Mohammed S Jasim is associated with the National School of Electronics and Telecommunications at the University of Sfax in Tunisia. His contributions to the field are evident through his affiliation with the University of Sfax in Sfax, Tunis. Dr. Jasim is actively involved in advancing knowledge and research in the domain of electronics and telecommunications. .

Nizar ZAGHDEN is an assistant professor at Ecole supérieure de Commerce in Sfax, Tunisia. . He is currently working on intelligent applications for smart cities dealing with classification of images and video from cctv cameras. With expertise in a variety of subjects, Dr. Zaghden contributes to the academic community through teaching and research at the institution. received the Master's degree in Novel Technologies in Dedicated Computer Systems (2005) from the National Engineering School of Sfax (Tunisia). In 2013, he obtained the PhD degree: in Computer System Engineering from the National Engineering School of Sfax (Tunisia) after defending his thesis intitled "caracterisation of the content of ancient document images". Concerning the professional career, he has been recruited, in 2009, as an assistant in the department of Computer Science in the Superior Institute of Informatics in Medenine (Tunisia). In 2013, he was promoted to the rank of assistant professor. In 2014 he has muted to the higher school of Business in Sfax (Tunisia) .



Pr. Med Salim BOUHLLEL was born in Sfax (Tunisia) in December 1955. He is a full professor at Sfax University, Tunisia. He is the Head of the Research Lab: Smart systems for Engineering & E-health based on Technologies of Image and Telecommunications (SETIT). He was the director of the Higher Institute of Electronics and Communications of Sfax- TUNISIA (ISECS) 2008-2011 He received the golden medal with the special appreciation of the jury in 1999 on the occasion of the first International Meeting of Invention, Innovation and Technology (Dubai, UAE). He was the vice president and founder member of the Tunisian Association of the Specialists in Electronics and the Tunisian Association of the Experts in Imagery. He is the president and founder of the Tunisian association on Human-Machine Interaction since 2013. He is the Editor in Chief of the international Journal "Human-Machine Interaction" and a dozen of special issues of international journals. He is the Chairman of many international conferences and member of the program committee of numerous international conferences. His research interests are Image processing, Telecommunication and Human-Machine Interaction in which he has obtained more than 20 patents so far. More than 400 articles were published in international journals, conferences and books. Moreover, he has been the principal investigator and the project manager for several research projects dealing with several topics concerned with his research interests mentioned above.