Machine Learning-Driven Strategies for Efficient Traffic Congestion Management

Mohammed Khasawneh

A Thesis

in

The Department

of

Information and Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Information and Systems Engineering) at

Concordia University

Montréal, Québec, Canada

September 2024

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis

prepared By: **Mohammed Khasawneh**

Entitled: **Machine Learning-Driven Strategies for Efficient Traffic Congestion Management**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________ Chair
Dr. Andrea Schiffauerova

______________________ External Examiner
Dr. Golam Kabir

______________________ Arm's Length Examiner
Dr. Fuzhan Nasiri

______________________ Examiner
Dr. Manar Amayri

______________________ Examiner
Dr. Farnoosh Naderkhani

______________________ Thesis Supervisor
Dr. Anjali Awasthi

Approved by ______________________
Dr. Farnoosh Naderkhani, Graduate Program Director

11/29/2024 ______________________
Dr. Mourad Debbabi, Dean, Gina Cody School of Engineering and Computer Science

Abstract

Machine Learning-Driven Strategies for Efficient Traffic Congestion Management

Mohammed Khasawneh, Ph.D.

Concordia University, 2025

Urban regions have a notable obstacle in the form of traffic congestion, which results in longer trip durations, higher fuel usage, and increased pollution levels. This study aims to tackle this issue by presenting a three step approach. The first approach uses Machine learning for Proactive Traffic Congestion Prediction. We explore multiple machine learning algorithms, such as Long Short-Term Memory (LSTM), Decision Tree (DT), Recurrent Neural Network (RNN), AutoRegressive Integrated Moving Average (ARIMA), and Seasonal ARIMA (SARIMA), to predict traffic congestion levels in different zones of the Montreal area. The results indicate that the Decision Tree approach surpasses other algorithms, attaining faster convergence, lower loss values, and a considerably higher R^2 score. After predicting the congestion using one of the prediction algorithms mentioned above, metaheuristic optimization algorithms are used to find near optimal cycle time for each traffic light. In step 2 Enhanced Bat Algorithm (EBAT) is proposed to adaptively modify traffic signal timings based on expected congestion levels. The EBAT algorithm utilizes adaptive parameter adjustment and guided exploration techniques that are dependent on the expected congestion. This results in enhanced performance when compared to the conventional Bat Algorithm. We conduct a comparative analysis of EBAT with various meta-heuristics, namely Particle Swarm Optimization (PSO), Cuckoo Search (CS), JAYA, Sine Cosine Optimization (SCO), and Harris Haws Optimization (HHO). The evaluation considers three scenarios: fixed-time traffic lights (baseline), dynamic traffic lights without prediction, and dynamic traffic lights with predicted congestion. The results demonstrate that EBAT yields substantial enhancements in both the rate at which convergence is achieved and the quality of the solutions, as compared to fixed and non-predictive scenarios. The second approach is using Multilevel Learning for Enhanced Prediction Accuracy. The precision of predicting traffic congestion depends on the ability to recognize and manage abnormal traffic patterns, especially in highly populated regions. Traditional prediction methods are vulnerable to these anomalies, as they frequently do not handle or clean the data. This can result in inaccurate forecasts, as the data may encompass anomalous occurrences such as accidents or unforeseen road closures, which can greatly distort the underlying trends. The study presents a novel and creative strategy to learning at several levels, which combines anomaly detection and ensemble learning approaches to tackle this problem. Anomaly detection techniques are used to find abnormal patterns within the data, which is then followed by the process of data cleansing. First, a set of initial learner models are trained. The top-performing models are then selected for an ensemble procedure, which involves combining their predictions through stacking and voting. Evaluated using a real-world Montreal traffic dataset, this multilevel methodology demonstrates higher prediction accuracy when compared to traditional approaches. The dataset is subjected to preprocessing techniques, such as windowing, to transform time-series data into frequency patterns in order to create a more generalized model. To leverage the detected anomalies, we utilized clustering algorithms, specifically K-Means and Hierarchical Clustering, to segment these anomalies. Each clustering algorithm was used to determine the optimal number of clusters. Subsequently, we characterized these clusters through detailed visualization and mapped them according to their unique characteristics. This approach not only identifies traffic anomalies effectively but also provides a comprehensive understanding of their spatial and temporal distributions, which is crucial for traffic management and urban planning. In summary, this study showcases the efficacy of a synergistic method that combines machine learning for proactive prediction of traffic congestion with metaheuristics for dynamic regulation of traffic lights. This method has the capacity to mitigate urban traffic congestion and enhance traffic flow efficiency. In addition, the use of a multilevel learning strategy to improve forecast accuracy is a noteworthy contribution to intelligent transportation systems. An application for city of Montreal is provided.

Acknowledgments

I would like to extend my sincere gratitude to all those who have provided support and encouragement throughout my PhD journey. I am particularly grateful to my thesis supervisor, Dr. Anjali Awasthi, whose exceptional mentorship and steadfast support have been central to my academic and personal development. Dr. Awasthi's guidance has been invaluable, pushing me to surpass perceived limits and fostering a spirit of scholarly independence. Her belief in my potential has significantly contributed to my achievements. I also wish to express my appreciation to my advisory committee for their insightful feedback and guidance. My heartfelt thanks go to my parents, Amneh and Ali, whose wisdom and lessons have continually guided and inspired me. This academic accomplishment is a testament to their unwavering support and my deep affection for them. The dedication, patience, and encouragement from my immediate family—my brothers and sisters—have been a constant source of motivation. I am deeply appreciative of their selfless sacrifices and steadfast support. Lastly, I offer heartfelt thanks to my wife, Walaa, and my daughter, Shams. Your love, resilience, and joy have been my refuge. I deeply recognize and appreciate the sacrifices you have made throughout this challenging and rewarding journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

There are many factors that have compelled us to investigate the issue of traffic congestion, as it has the potential to impact various aspects of our daily routines. The primary motivation is to alleviate traffic congestion in the road network, thereby reducing the waiting time for both drivers and pedestrians. This will have a direct impact on the productivity of each individual in their daily lives. While the effect may seem insignificant for each person individually, when viewed holistically, it will have a substantial impact on society as a whole.

Another incentive is to reduce the emissions of CO and CO2 gases from automobiles, which significantly contributes to one of the prevailing global issues, namely the depletion of the ozone layer. This depletion has direct implications for human health and must be addressed. Furthermore, the use of this approach leads to a decrease in fuel consumption, allowing scientists an extended duration to explore alternative choices, such as renewable energy, to replace the existing energy sources that are projected to deplete in the foreseeable future. The implementation of congestion mitigation measures will significantly enhance user satisfaction with the roads and improve overall quality of life. Additionally, it will lead to a reduction in the number of car accidents, thereby saving lives globally[1].

Traffic congestion is a complex issue, and traditional prediction models frequently face difficulties in including unexpected events. Anomaly detection techniques provide a solution by accurately identifying these disruptions within the traffic data. This enables the prediction algorithm to adapt its predictions for enhanced precision and potentially identify the source of congestion (such as accidents or road closures).

Moreover, anomaly detection functions as a mechanism that filters out abnormal data points, which have the potential to distort the outcomes of the model. The process of ”cleaning” the data enhances the accuracy of predictions by specifically targeting common traffic flow patterns. Ultimately, the utilization of anomaly detection aids in the development of a more resilient and flexible traffic prediction model, hence facilitating smoother traffic flow and enhancing traffic management tactics.

However, despite the data being cleaned by anomaly detection, the nature of traffic flow can still be complex. traffic flow during rush hour are distinct from weekend traffic, and seasonal fluctuations might additionally influence these patterns. Here’s where windowing techniques come in. The data is divided into smaller time intervals, enabling the model to accurately capture the subtle variations in traffic patterns at different levels of detail.

Ultimately, even possessing a thorough and subtle comprehension of traffic patterns, relying solely on one prediction model may prove insufficient. This is when the combination of multilevel learning and ensemble learning becomes dominant. Consider the scenario of training numerous models using data from various time intervals, such as hourly, daily, and weekly. By combining forecasts from different models, the system exploits the advantages of each and enhances the overall resilience and precision of the traffic congestion prediction. Essentially, this sequential methodology facilitates the development of a more refined comprehension of the movement of vehicles, resulting in improved traffic flow and enhanced tactics for traffic management.

Our study is motivated by the necessity to enhance current anomaly detection methods through the utilization of clustering techniques that can segment and characterize identified anomalies. Visualizing and mapping these clusters allows transportation planners and managers to get significant insights into the spatial and temporal distributions of traffic anomalies. These insights are essential for making well-informed decisions in urban planning and traffic management, as they enable the identification of precise regions or time periods that require targeted interventions.

1.2 Existing Problems in Congestion Modeling

Metropolitan areas across the globe are confronted with an enduring and escalating challenge of traffic congestion. The continuous increase in the number of vehicles on the road exceeds the capacity of the current infrastructure. This disparity leads to a series of adverse consequences, such as extended travel times for motorists, elevated rates of traffic collisions, and intensified levels of air pollution. These problems highlight the urgent requirement for more efficient traffic management methods. Nevertheless, existing methods are impeded by constraints that restrict their capacity to successfully tackle these problems.

Problem 1: Inefficiencies in Traffic Management Systems

Current algorithms employed for optimizing traffic signal management frequently encounter difficulties with slow execution times. They take a significant amount of time to identify the best solutions for congestion control, limiting their effectiveness in real-world applications. Moreover, machine learning algorithms assigned with the responsibility of forecasting traffic congestion can exhibit inaccuracies. This poses a challenge for authorities to implement preemptive steps and mitigate the occurrence of congestion. The existing constraints in present traffic management systems greatly exacerbate the escalating issue of traffic congestion.

Problem 2:Challenge of Data Anomalies and Single Model Limitations in Traffic Forecasting.

Precisely predicting traffic congestion, especially in heavily populated locations, is crucial for efficient traffic light management. Nevertheless, traffic data obtained from real-world sources may be subject to noise and anomalies. These anomalies, such as unforeseen mishaps or meteorological occurrences, can greatly distort forecasts and make them unreliable. The presence of data anomalies poses a significant obstacle to accurately predicting traffic congestion, hence reducing the efficiency of existing traffic management systems.

Even with clean data, depending exclusively on a single machine learning model for predicting traffic encounters additional difficulties. The movement of traffic is an intricate occurrence that is affected by various elements, including weather conditions, special occasions, and the accidents. A single model may not be suitable for accurately capturing all of these complex relationships. Furthermore, certain machine learning models are susceptible to significant variability. Consequently, even minor alterations in the training data might result in substantial variations in forecasts, thereby diminishing the overall dependability of the forecast.

Problem 3:Identification and Characterization of Traffic Flow Anomalies.

The study of identifying and characterizing traffic flow anomalies is important because they have a major influence on transportation efficiency and urban mobility. Traffic anomalies, such as sudden increases or decreases in traffic volume, have the potential to cause congestion, delays, or accidents, hence interrupting the efficient functioning of transportation networks. These disruptions affect daily commuting, economic productivity, and public safety. Through the identification and characterization of these anomalies, authorities responsible for traffic management can promptly adopt interventions to mitigate congestion and solve safety issues. Analyzing the spatial and temporal patterns of traffic anomalies is crucial for informing long-term urban planning and infrastructure development. This analysis enables more efficient traffic signal regulation, optimization of road networks, and allocation of resources. Addressing traffic flow irregularities fundamentally enhances the efficiency, safety, and sustainability of transportation networks, a crucial aspect in rapidly expanding urban regions.

1.3 Thesis Objectives

In this section, we provide our main research objectives:

- **Intelligent Meta-Heuristic Based Optimization of Traffic Light Timing Using Artificial Intelligence Techniques:** We propose in this study the following solutions:**(i)** Design a machine learning algorithm to predict traffic congestion levels proactively. **(ii)** Design a meta-heuristic algorithm for dynamic traffic light control based on these predictions.

- **Multilevel Learning for Enhanced Traffic Congestion Prediction using Anomaly Detection and Ensemble Learning:** We manipulate in this study the challenge of data anomalies and single model limitations in traffic forecasting so we propose the following solutions: **(i)** implementing of diverse anomaly detection approaches to remove outliers from the traffic data. **(ii)** employing a multilevel learning technique that incorporates anomaly detection to improve data quality for prediction.

- **Clustering Based Approach for Enhanced Characterization of Anomalies in Traffic Flows:** In this study we enhance our analysis by employing clustering techniques, namely K-Means and Hierarchical Clustering, to further investigate the attributes of the found anomalies. By utilizing sophisticated windowing approaches, we achieved reliable anomaly identification, and by subsequently applying clustering, we obtain comprehensive understanding of their geographical and temporal patterns.

  These objectives are described in detail as follow:

1.3.1 Traffic Congestion Prediction Using Machine Learning Approaches

1.3.1.1 Design a machine learning algorithm

This section outlines the initial phase of our machine learning approach for proactive traffic congestion prediction. Here is a detailed analysis of the process:

- **Data Collection:** We collect comprehensive traffic data, including information on traffic volumes, vehicle speeds, and road occupancy. The purpose of collecting this data is to capture historical trends and seasonal variations.

- **Model Training and Evaluation:** We employ different machine learning algorithms like Recurrent Neural Networks, LSTM, DT, ARIMA, and SARIMA are used to forecast traffic congestion.These models are trained using historical data to learn knowledge about the correlations between traffic features and levels of congestion. The accuracy of these prediction algorithms is then assessed by comparing the projected values to real-world congestion data and use metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE). After the models have been verified, they are used to make predictions about traffic congestion, which can provide insight on how the situation will unfold in the future.

The model's possesses an ability to adjust to dynamic traffic situations. As further data becomes accessible, the model may be utilized to continuously provide predictions that accurately represent the changing traffic patterns.

1.3.2 Intelligent Meta-Heuristic Based Optimization of Traffic Light Timing

1.3.2.1 Design a meta-heuristic algorithm

The results of these forecasts are saved in a special Congestion Database, which is subsequently used for analysis and decision-making in the future. In the second solution, meta-heuristic algorithms are used to improve traffic flow by optimizing the green time for each traffic light. The green time for traffic lights at various junctions can be strategically adjusted with the help of algorithms like the BAT Algorithms, Particle Swarm Optimization, Cuckoo Search algorithm, and the Jaya algorithm. The ultimate goal is to reduce the amount of time cars spend waiting at these junctions. Congestion will be reduced, traffic flow will be improved, and a more effective transportation system will be created thanks to this optimization process.

1.3.3 Multilevel Learning for Enhanced Traffic Congestion Prediction using Anomaly Detection and Ensemble Learning

1.3.3.1 Implementing of diverse Windowing and anomaly detection approaches

Traffic data is partitioned into smaller time windows using windowing techniques.The windowing technique is a data preprocessing stage that involves transforming time series data of vehicle frequencies into sequence patterns using fixed intervals. This technique facilitates the analysis of data collected from city intersections, creating more structured and systematic sequence patterns. These patterns can be used to incorporate trends and frequency routines in the data based on given window settings in the learning process, gaining valuable insights into the behavior of vehicles at city intersections, which can lead to more accurate predictions. This enables the examination of short-term traffic patterns and the detection of anomalous events within certain time periods. Afterwards,we evaluate different anomaly detection techniques(Isolation Forest, Elliptic Envelope, and Local Outlier Factor) to identify unusual traffic patterns in different locations over time. After predicting the anomaly, the data pattern is cleaned.

1.3.3.2 Employing a multilevel learning technique

Our study involved using an ensemble learning strategy to improve the accuracy of our predictions. In this framework, we examine the application of both the voting model with equal weights and stacking models. Ensemble learning techniques strive to improve predictive accuracy by combining the outputs of many base models. This approach allows for the utilization of the strengths of individual models while minimizing their limitations. In order to guarantee the strength and durability of our ensemble, we carefully chose the top six performing models from the baseline linear regression models to be included in the ensemble architecture.

1.3.4 Clustering Based Approach for Enhanced Characterization of Anomalies in Traffic Flows

We investigated anomalies in traffic patterns using the anomalies determined in our previous study[1], where three anomaly detection techniques:Elliptic Envelope, Isolation Forest, and Local Outlier Factor—were employed. These methods were applied to a dataset that had been pre-processed using windowing techniques with different configuration settings to enhance the detection process. In this research, to leverage the detected anomalies, we utilized clustering algorithms, specifically K-Means and Hierarchical Clustering, to segment these anomalies. Each clustering algorithm was used to determine the optimal number of clusters. Subsequently, we characterized these clusters through detailed visualization and mapped them according to their unique characteristics. This approach not only identifies traffic anomalies effectively but also provides a comprehensive understanding of their spatial and temporal distributions, which is crucial for traffic management and urban planning.

1.4 Interrelations Among Research Contributions

The three research contributions are closely integrated, each addressing critical aspects of traffic congestion management, as shown in Figure 1.1. Together, they create a comprehensive framework for predicting traffic congestion, optimizing signal timing, and generating actionable insights for decision-makers.

The first contribution, depicted on the left side of the figure, focuses on traffic congestion prediction using machine learning models such as RNN, LSTM, Decision Tree, ARIMA, and SARIMA, combined with meta-heuristic algorithms for optimizing traffic flow. However, this phase has notable limitations. The data is used directly for prediction without segmenting it into time windows, which limits the model's ability to capture short-term or periodic fluctuations (e.g., rush hours) in traffic patterns. Additionally, there is no mechanism for detecting or handling anomalies, such as accidents or traffic spikes, which reduces the system's ability to adapt to irregular traffic patterns. This could lead to inaccurate predictions, as outliers are not filtered out.

The second contribution, illustrated on the right side of the figure, addresses these limitations by introducing essential data preprocessing steps. The data is segmented into time windows to capture temporal variations in traffic patterns, such as rush hour trends, which improves the accuracy and granularity of predictions. This phase also incorporates techniques to identify and remove anomalies from the dataset. By filtering out outliers like accidents and unusual traffic events, the system ensures that only reliable data is used for prediction, making the models more robust and accurate in real-world scenarios. Additionally, ensemble learning methods (e.g., stacking and voting) are employed to reduce bias and further improve prediction accuracy. This contribution enhances the foundation established by the first phase, resulting in a more reliable traffic congestion forecasting system.

As part of the second contribution, after the final prediction phase, meta-heuristic optimization is applied to find a near-optimal solution for signal timing (e.g., adjusting green light durations). This future work, as indicated in the figure, aims to optimize traffic light timings using the predicted congestion data, further improving traffic flow efficiency.

The third contribution, shown on the right side of the figure, involves clustering detected anomalies using techniques such as K-Means and Hierarchical Clustering. This phase provides valuable insights into the spatial and temporal patterns of traffic anomalies, enabling decision-makers to make informed adjustments to traffic signals, optimize resource allocation, and plan infrastructure improvements. By transforming the raw anomaly data into actionable insights, this contribution enhances long-term traffic management and urban planning strategies.

In conclusion, these three contributions work in harmony: the first establishes the predictive framework, the second improves prediction accuracy and robustness by addressing the limitations of the first, and the third provides decision-makers with practical insights to optimize traffic flow and improve urban mobility.

1.5 Thesis Organization

This thesis is organized into six chapters.Chapter 1 introduces this thesis work. Chapter 2 reviews the related literature. Chapter 3 focuses on the application of intelligent meta-heuristic optimization techniques for traffic light timing, utilizing Machine learning methods. Chapter 4 presents our research work on a multilevel learning approach for traffic congestion prediction, highlighting the use of anomaly detection and ensemble learning techniques. Chapter 5 presents a clustering-based approach for the characterization of anomalies in traffic flows. Finally, we conclude our thesis in Chapter 6.

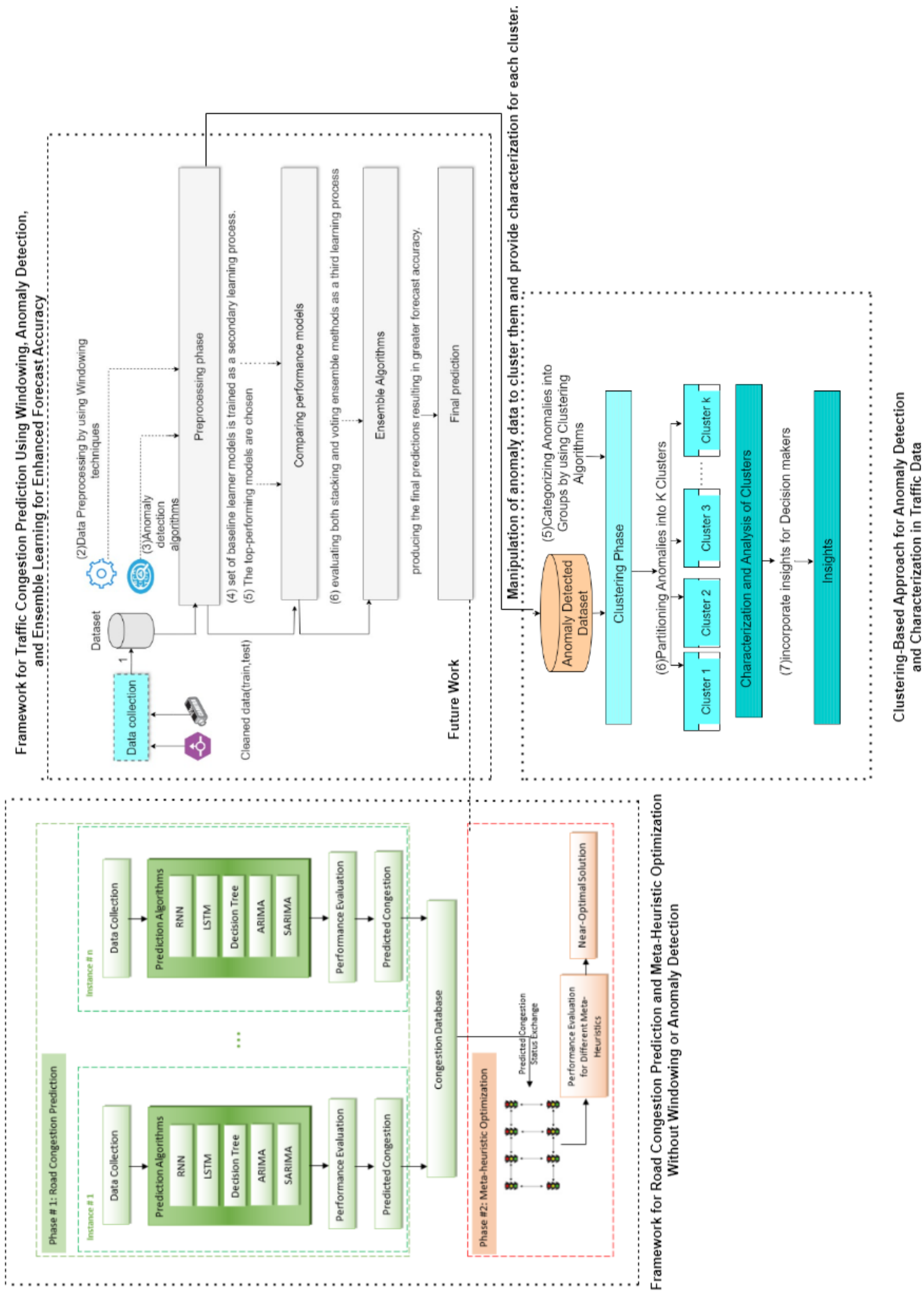


Figure 1.1: Overall Architecture of The Thesis

Chapter 2

Background and Literature Review

Traffic congestion presents a significant challenge in urban environments, where the demand for roadway space frequently surpasses the available capacity. This disparity leads to reduced speeds, longer travel times, and increased vehicle congestion. Congestion impairs the efficiency of the transportation system, resulting in delays that impact both individual commuters and the economy, as well as the environment. Congestion typically results from a confluence of elevated vehicle density and inadequate road infrastructure, frequently exacerbated by unforeseen incidents such as accidents or adverse weather conditions.

Many different kinds of traffic congestion exist, each characterized by unique causes and effects. Recurring congestion is the most prevalent, occurring during peak periods when road utilization is at its highest, namely during morning and evening rush hours. It is predominantly foreseeable and arises from everyday traffic patterns surpassing road capacity. Conversely, non-recurring congestion arises from transient disturbances, such accidents, road construction, or weather conditions. This form of congestion is unpredictable and can result in significant delays, even in regions where traffic typically flows smoothly.

Another type of congestion is bottleneck congestion, which occurs at specific points on the road network where the capacity is reduced, like toll plazas, constricted crossroads, or lane cutbacks. Localized difficulties generate traffic congestion when vehicles navigate through constrained segments of the roadway. Moreover, congestion associated to incidents arises from particular occurrences such as accidents or vehicle malfunctions that abruptly interrupt the standard traffic flow, frequently leading to significant delays that spread well beyond the site of the incident.

To manage and mitigate congestion, various approaches to modeling traffic have been developed. Deterministic models rely on established correlations among traffic flow, speed, and density. A crucial tool in this methodology is the Fundamental Diagram of Traffic Flow, which forecasts congestion by examining the impact of variations in speed and vehicle density on traffic flow. Deterministic models are beneficial for understanding fundamental congestion dynamics; yet, they may inadequately represent the complexity of real-world traffic situations.

Stochastic models, conversely, integrate randomization to address uncertainty in traffic patterns. These models employ probabilistic techniques to forecast traffic patterns, especially during unforeseen occurrences.Queueing theory can be utilized to evaluate the effects of stochastic vehicle arrivals at intersections or toll booths, offering insights into wait times and congestion development under unknown circumstances. Stochastic models provide a more adaptable method for predicting congestion, particularly in contexts where unpredictability is a crucial factor.

As traffic data becomes increasingly accessible, simulation models and machine learning algorithms have emerged as vital tools for traffic management. Simulation models, including microscopic and mesoscopic simulations, replicate the movement of individual cars to offer intricate depictions of traffic flow. These models are proficient in capturing the dynamic characteristics of traffic and identifying possible congestion hotspots. Machine learning algorithms, such as Decision Trees (DT), Long Short-Term Memory (LSTM), and AutoRegressive Integrated Moving Average (ARIMA), utilize historical and real-time data to forecast future congestion patterns, facilitating proactive traffic management strategies. Together, these approaches offer an extensive framework for understanding, forecasting, and alleviating traffic congestion in urban environments.

This chapter provides a comprehensive analysis of previous research conducted on the specific problem areas that we have identified.The review of literature is organized in the following manner. In Section 2.1, we conduct a literature study in the field of proactive traffic management. In Section 2.2, we conduct a literature analysis in the field of addressing data anomalies and single model limitations.In Section 2.3, we conduct a literature study in the field of traffic anomaly detection, clustering, and prediction techniques.

2.1 Proactive Traffic Management

Urban areas have significant challenges due to traffic congestion. Traditional traffic signal control systems, which rely on fixed timings, prove ineffective when confronted with fluctuating traffic situations. This section explores different methodologies aiming at enhancing traffic flow through proactive management.

Below is an analysis that is structured according to the methods and problems being discussed:

2.1.1 Machine Learning for Traffic Flow Prediction

Urban areas continue to face significant challenges in the way of traffic congestion, resulting in longer travel times, annoyance, and pollution of the environment. Traditional traffic signal control systems depend on predetermined timings, which face difficulties in adjusting to current traffic conditions. This section examines recent advances in machine learning, particularly Deep Reinforcement Learning (DRL), for the purpose of forecasting traffic flow and enhancing traffic light control.

2.1.1.1 Deep Reinforcement Learning for Traffic Light Control

DRL presents a highly favorable method for intelligent traffic light control. The system employs an agent that observes traffic patterns, such as the number of vehicles and the length of queues, and adapts traffic light cycles (red, green, yellow) in order to accomplish certain objectives. The objectives can vary from reducing the average waiting times for all vehicles to prioritizing emergency vehicles [2].

The DRL agent operates by employing a method of trial and error. The system is rewarded for successful actions, such as reducing congestion, and learns from its mistakes, such as causing long waits. The DRL model utilizes deep neural networks to identify complex relationships between traffic patterns and optimal light timings.This allows the agent to predict future traffic patterns and modify traffic light timings accordingly.

In [3] authors present a DRL method for real-time traffic light control.The system takes into account various factors such as queue lengths, delays, travel time, and throughput. This system exhibits significant enhancements in decreasing wait times and congestion as compared to traditional methods.

Reinforcement learning [4]–[6] has been used to dynamically adapt traffic lights to current traffic conditions. Both representing the environment and modeling the relationship between the environment and the decision provide significant obstacles to the practical use of traditional reinforcement learning. In order to overcome these obstacles, researchers [5] have recently applied deep reinforcement learning methods, such as Deep Q-Learning (DQN), to the traffic light management problem.

The deep reinforcement learning strategy [7], one of the machine learning techniques widely recognized to handle these kinds of issues, is utilized to model brain behavior for the processing phase. To optimize rewards, such as reducing average waiting times in traffic control scenarios, reinforcement learning encourages agents to learn the best course of action by observing and interacting with their surroundings [8]. For dynamic traffic signal control systems, research on the reinforcement learning technique, which defines the states by the waiting line length [9], [10], is sparse. However, the queue length does not always reflect actual traffic conditions. More accurate traffic light management, taking into account variations in vehicle types including ambulances, fire trucks, school buses, and police cars, is possible due to the proliferation of high-definition cameras and sensors made possible by technological progress [11]. In [12], the authors propose an Intelligent Vehicle Pedestrian Light (IVPL) system using deep reinforcement learning to optimize traffic signals for both vehicles and pedestrians, minimizing total user delays. Tested in Melbourne, the model outperforms traditional systems by adapting to real-world scenarios, including jaywalking, and balancing green time allocation for mixed traffic flows. In [13], a distributed deep reinforcement learning system for traffic light control that employs CNN-based deep Q-networks as well as a global update consensus mechanism is presented. In SUMO simulations, the model performs similarly to centralized learning while outperforming fixed-time and local learning strategies in controlling both homogeneous and heterogeneous traffic flows.

2.1.1.2 Dynamic and Intelligent Traffic Light Control System (DITLCS)

DITLCS utilizes DRL to make accurate predictions about the flow of traffic in real-time. The DRL model utilizes real-time data on vehicle count and types to make predictions about future traffic conditions[14]. DITLCS has the ability to adaptively adjust traffic signal timings in order to maximize the efficiency of traffic movement, as indicated by these forecasts. The system has the capability to function in several modes, giving priority to particular types of vehicles or emergency vehicles depending on the real-time situation .

In [14] DITLCS was presented as a solution, which uses inputs such as real-time traffic data to address these problems and improve the system's overall efficiency. In addition, the proposed DITLCS can operate in three distinct modes: Fair Mode (FM), Priority Mode (PM), and Emergency Mode (EM), where the first prioritizes all vehicles equally, the second prioritizes vehicles based on their category, and the third prioritizes emergency vehicles above all others. Additionally, a deep reinforcement learning model was presented to change the traffic lights between phases (red, green, and yellow), and a fuzzy inference system chooses one of three modes (FM,PM, and EM) based on the data. The researchers used an open-source simulator (the Simulation of Urban Mobility (SUMO) systems) to test DITLCS by simulating it on a map of the city of Gwalior, India.

2.1.2 Optimization Algorithms and Scheduling

Urban Traffic Light Scheduling Problems (UTSLPs) are studied by some authors, and they aim to minimize the sum of all vehicle and pedestrian delays within a specified time window. The UTLSP is first described using a centralized model that presents the cost functions and constraints of the two objectives [15]. To evaluate and rank approaches in terms of the two objectives, we use a metric that does not rely on a dominance strategy. Second, meta-heuristics like Harmony Search (HS) and Artificial Bee Colony (ABC) are used to find a solution to the UTLSP. There are four distinct approaches to managing traffic lights: fixed-time versus adaptive, and standalone versus collaborative.Some of the more prominent plans put forth in recent decades are TRANSYT, SCOOT [16], OPAC [17], PRODYN [18], CRONOS [19], and RHODES [20].

The enormous computational cost of optimization becomes the key challenge for realtime scheduling because of the massive size of a typical traffic network, which can have thousands of road links and hundreds of intersections. Many researchers have presented different optimization strategies for the traffic light control problem. Case studies have been conducted to evaluate the proposed architecture in a real-world traffic network , and an architecture for optimizing traffic light cycles using a Genetic Algorithm (GA) and Cellular Automata Simulators (CAS) has been developed [21]. Case studies in real-world traffic networks have confirmed the substantial benefits of the suggested approach [22], and a PSO-based approach was proposed and deployed, coupled with a microscopic traffic simulator [23]. A model of a traffic network with continuous flow is discussed in [24], with the decision to change traffic lights being depicted as a discrete event. Meta-heuristic optimization techniques, such as the Genetic GA, PSO, Differential Evolution DE, HS, and the ABC algorithm, are among the most promising new methods for addressing traffic light management issues [25]. In [26] create advanced techniques using meta-heuristics to enhance the management of traffic at a single traffic light in Dhahran, Saudi Arabia. The work attempts to optimize signal timing plans in order to enhance the level of service (LOS) at intersections, using the GA and DE. The findings indicate that both the GA and DE generate a methodical schedule for signal timings, resulting in a notable decrease in travel time delay ranging from 15 to 35% as compared to the current conditions.

While DE exhibits faster convergence to the target function, GA surpasses DE in terms of solution quality, specifically in minimizing vehicle delay. The validation results showcase the sufficiency and resilience of the proposed approaches, highlighting the significance of traffic signal control in intelligent transportation systems.

The use of an Adaptive Neuro-Fuzzy Inference System (ANFIS) is a viable approach to tackle the problem of traffic signal optimization. In [27] the ANFIS traffic signal controller employs meta-heuristic algorithms to determine the optimal duration for green lights at traffic signals, thereby minimizing both the queue length and the delay. The controller was simulated and implemented on intersections, showcasing exceptional efficacy in traffic prediction and control.

Another study in [28] presents the Meta-Heuristic Robust plan Approach (MHRA), which is a framework for fixed-time traffic lights that operates in an offline scenario-based manner. The

MHRA evaluates the most effective signal schemes for different demand scenarios. The effectiveness of the framework was confirmed through numerical experiments. These experiments demonstrated that the framework surpasses nominal plans and consistently performs well even when faced with changing demand. Comparing MHRA with other scenario-based methodologies through benchmarking demonstrates its superior efficiency.

Another research paper in [29] discussed how to create an advanced traffic control protocol utilizing the Non-dominated Sorting Genetic Algorithm II (NSGA-II) at isolated signalized intersections in Dhahran, Saudi Arabia. The Measures of Effectiveness (MOEs) taken into account encompassed mean vehicle latency, overall vehicle stops, mean fuel consumption, and vehicular emissions. The simulations demonstrated that the proposed strategy effectively optimized performance metrics, resulting in a 16% to 23% improvement in MOEs compared to the current settings. A Synchro traffic light simulation and optimization program was used to conduct an optimization analysis. The results indicated that the proposed approach performed better than the Synchro optimization results in terms of the percentage reduction in MOE values [29].

In [30], the paper showcases a case study of a significant intersection in Timisoara, where the synchronization of traffic lights is accomplished by the utilization of firefly algorithms. Although these algorithms are effective, the study demonstrated that more enhancements to traffic flow are still possible. In [31], the authors provided a thorough analysis of the most recent developments in swarm intelligence and evolutionary approaches as they are applied to traffic control and optimization in metropolitan networks.

In [32], the authors propose a Discrete Sine-Cosine Algorithm (DSCA) and its derivatives to optimize urban traffic light scheduling for mixed pedestrian-vehicle networks. The technique improves traffic flow efficiency and decreases delays by balancing vehicle and pedestrian circulation. This metaheuristic-based approach enhances signal timing methods in complex metropolitan settings.

Table 2.1.1 provides a comprehensive summary of previous research efforts in the field of traffic prediction techniques. It details various methods that have been utilized in past studies, highlighting their strengths and drawbacks.

Table 2.1.1: Comprehensive Analysis of Traffic Prediction Techniques

References	Methods	Strength	Drawback
Kumar et al. (2021) [14]	Deep Reinforcement Learning (DRL)	Effective in learning complex traffic patterns; Real-time adaptability; Combines fuzzy logic for enhanced decision making	High computational cost; Requires large datasets for training
Gupta et al. (2022) [12]	Deep Reinforcement Learning (IVPL)	Optimizes signals for both vehicles and pedestrians; adapts to jaywalking	Requires real-world calibration and complex data collection
Chengula et al. (2024) [13]	Distributed Deep Q-Networks (DQN)	Effective for heterogeneous traffic flows; decentralized control	High computational demand; requires consensus mechanism
Kuyer et al. (2008) [4]	Multi-Agent Reinforcement Learning	Handles multi-agent interactions; Effective for complex traffic scenarios; Scalability	Coordination among agents can be challenging; High computational complexity
Van der Pol & Oliehoek (2016) [5], Wiering (2000) [6]	Multi-Agent Reinforcement Learning, Coordination Graphs	Improved coordination among traffic signals; Higher efficiency in real-time traffic control; Early demonstration of RL feasibility	Requires large training data; High computational requirements; Less efficient compared to modern techniques
Shen et al. (2014) [7], Wang et al. (2010) [10]	Particle Swarm Optimization (PSO)	Effective for multicast routing; Balances exploration and exploitation; Hierarchical approach	Specific to communication networks; Not directly applied to traffic light control
Taherkhani & Pierre (2016) [8], Kumar & Kumar (2016) [9]	Machine Learning Clustering Algorithm, Position-Based Routing	Efficient data congestion control; Maximizes network lifetime	Primarily focused on vehicular ad hoc networks; Limited direct application to traffic lights
Hunt et al. (1982) [16], Gartner (1983) [17]	SCOOT On-Line Optimization, OPAC (Demand-Responsive Strategy)	Real-time adaptability; Continuous optimization; Responsive to real-time demand	High initial setup cost; Requires continuous monitoring and adjustment
Farges et al. (1983) [18], Boillot et al. (1992) [19]	PRODYN Real-Time Algorithm, Optimal Signal Control	Real-time adaptability; Effective in complex urban networks	High computational requirements; Implementation complexity
Sen & Head (1997) [20]	Phase Optimization, Evolutionary Optimization	Systematic approach to phase optimization; Applicable to real-world scenarios	Limited adaptability; Fixed timing plans; Computationally intensive
Sanchez et al. (2004) [21], García-Nieto et al. (2012) [22], Garcia-Nieto et al. (2013) [23]	Genetic Algorithms, Cellular Automata, Swarm Intelligence (PSO)	Combines GA with cellular automata; Effective for large-scale optimization; Continuous adaptation	High computational requirements; Complexity in implementation
Göttlich et al. (2015) [24], Cheng et al. (2017) [25]	Modeling and Optimizing Traffic Light Settings, Fuzzy Group-Based Intersection Control	Continuous flow modeling; Effective optimization strategies; Effective for smart transportations	High computational cost; Requires detailed and continuous data
Jamal et al. (2020) [26], Shirke et al. (2022) [28]	Meta-Heuristic Search Algorithms, MHRA	Delay optimization; Significant reduction in travel time delay; Effective for varying traffic demand	High computational requirements; Complex implementation
Shahkar et al. (2023) [27], Al-Turki et al. (2020) [29]	ANFIS, Meta-Heuristic Algorithms, NSGA-II	Effective traffic prediction and control; Minimizes queue length and delay; Significant improvement in performance metrics	Requires continuous data; Computationally intensive
Tang et al. (2023)	Discrete Sine-Cosine Algorithm (DSCA)	Balances pedestrian and vehicle flow; improves traffic efficiency	Optimization results depend on parameter tuning
Szatmari et al. (2022) [30], Jamal et al. (2023) [31]	Firefly Algorithms, Swarm Intelligence, Evolutionary Approaches	Effective synchronization of traffic lights; Continuous optimization; Thorough analysis; Effective for traffic control and optimization	Computationally intensive; Complexity in implementation; High computational requirements

2.2 The Challenge of Data Anomalies and Single Model Limitations

This section delves into the most recent studies that focused on predicting traffic patterns and detecting anomalies. We highlight the importance of windowing techniques and ensemble learning algorithms in achieving accurate forecasts. Additionally, we discuss the potential of these techniques in improving traffic management within Intelligent Transportation System(ITS). In order to provide a comprehensive understanding, we have classified previous research based on the techniques employed. Further information is presented in the following sections.

2.2.1 Traffic Flow Prediction using Windowing Techniques with Deep Learning Models

Traffic congestion at intersections continues to be a significant bottleneck in urban transportation networks. Several studies have investigated methods for forecasting and controlling traffic congestion by combining windowing techniques with Deep Learning models. These techniques involve dividing historical traffic data, such as speed and volume, into smaller intervals, known as windows, to capture the temporal patterns of traffic flow and enhance the precision of predictions.

2.2.1.1 Deep Learning Models for Windowing-based Prediction

Deep learning demonstrates remarkable proficiency in analyzing sequential data through the utilization of windowing-based prediction. This technique involves segmenting the data into fixed-size windows, allowing the model to focus on relevant patterns within each timeframe [33]. This approach is prevalent in time series forecasting, signal processing, and other tasks that involve sequential data analysis, enabling the model to capture complex relationships and make accurate predictions.

In [34], the authors introduced LSTM recurrent neural networks as a method for predicting short-term traffic flow. A windowing-based prediction approach was used, where historical traffic flow data was divided into fixed-size windows. Each window consisted of a series of past traffic flow measurements and different data, such as traffic flow values, time of day, day of the week, and maybe weather conditions, that have been gathered. Subsequently, the LSTM network underwent training utilizing these windows as input, allowing it to scrutinize patterns in the training data and generate forecasts for forthcoming traffic flow values.

In a similar manner, [35] presented a technique for forecasting short-term traffic patterns utilizing networks based on LSTM. They implemented a prediction approach based on windowing, where they divided the past traffic flow data into chunks of a fixed size. It is that important attributes, such as the amounts of traffic flow and temporal characteristics, were taken out from each window. The LSTM network was then trained using these windows as input, enabling it to acquire temporal correlations in the data and produce accurate predictions for future time points.

A specific design of recurrent neural network (RNN) was suggested in [36] to accurately anticipate traffic flow over long durations. They have employed windowing-based prediction by dividing previous traffic flow data into overlapping or non-overlapping windows. the features that capture long-term interdependence in traffic flow data were obtained from each window. The RNN model was trained using these windows as input, allowing it to comprehend temporal patterns in traffic flow and make predictions for extended time periods.

In a similar manner, spatiotemporal recurrent convolutional networks(SRCNs) were created by [37] for the purpose of predicting traffic patterns. The utilization of a windowing-based prediction methodology have been implemented by partitioning spatiotemporal traffic flow data into windows of a predetermined size.the features that encompass both the spatial and temporal aspects of traffic flow were retrieved from each window. The SRCN model was subsequently trained using these windows as input, allowing it to understand the geographical and temporal linkages in the data and produce accurate forecasts for future time periods.

Furthermore, the study conducted in [38] delved into the application of deep learning techniques for the purpose of predicting short-term traffic flow. The primary emphasis of this research was on the analysis of time series data. Windowing-based prediction involved dividing past traffic flow data into fixed-size windows or time frames. In [39] the researchers examined the prediction of immediate demand for online car-hailing services using sequential data. They suggested the use of windowing-based prediction, which involves breaking past demand data into predetermined-sized windows.

In a similar manner, in [40], the authors employ a windowing strategy to improve the precision of their models. This method entails dividing a continuous stream of data into smaller, fixed-size parts or ”windows.” The model can examine and find patterns inside each segment individually as each window has a specific period of data. This approach facilitates the handling of extensive datasets and enhances the identification of temporal trends that are vital for forecasting traffic accidents. By successively analyzing these frames, the model can accurately and promptly forecast traffic events by capturing the dynamic changes in traffic circumstances over time.

In [41], the authors present a robust multi-modal pedestrian detection method using a deep convolutional neural network and ensemble learning.The model achieves 99.30% accuracy, exceeding typical CNNs, making it ideal for video monitoring in smart transportation systems.

In [42], a Dynamic Factor Model (DFM) for multi-step traffic performance prediction in metropolitan road networks. The model accurately forecasts traffic states by incorporating spatial-temporal correlations between road segments, as measured by the Traffic Performance Index. This strategy promotes proactive traffic management and congestion reduction, hence improving decision-making in urban transportation networks is presented.

2.2.2 Anomaly Detection in Traffic Data with Advanced Techniques

Traffic anomalies are variations from normal traffic patterns, usually induced by accidents, weather conditions, or other interruptions. As previously mentioned, windowing approaches can serve as a basis for detecting anomalies. However, more sophisticated methods harness the capabilities of specialized models.

2.2.2.1 Causal Discovery for Anomaly Detection

Causal discovery for anomaly detection refers to the process of identifying and understanding the causal relationships among variables in a dataset to detect anomalies or unusual patterns [43]. This approach leverages causal inference techniques to uncover the underlying structure and dynamics of the data, which can improve the accuracy and interpretability of anomaly detection.

The authors in [44] utilized causal discovery techniques to ascertain causal links among variables in the data. They used techniques like Bayesian networks or causal inference algorithms to deduce causal structures from the available data. Through comprehending these causal connections, the authors were able to detect anomalous patterns or anomalies that diverge from the anticipated causal relationships.

In the same way, the authors in [45] employed causal discovery techniques to reveal the fundamental reasons behind traffic abnormalities. They have utilized techniques such as structural equation modeling or Granger causality analysis to ascertain causal connections between traffic variables and probable contributing factors. By doing this study, the authors successfully identified the fundamental factors responsible for traffic anomalies, which in turn facilitated the process of root cause analysis and improved the effectiveness of anomaly detection.

In [46], the authors propose a novel method for detecting traffic flow outliers based on Stochastic Differential Equations (SDEs) and Gaussian Process Regression (GPR). The strategy increases real-time detection by collecting dynamic changes in traffic data, resulting in greater robustness and adaptability than existing methods.

2.2.2.2 Spatial-Temporal Graph Neural Networks (GNNs)

GNNs are a type of neural network specifically designed to handle data that has both spatial and temporal components. The authors in [48] concentrated on detecting aberrant patterns or abnormalities in remote sensing data. Although GNNs were not explicitly utilized, their approach could potentially gain advantages by integrating these models in future studies. GNNs are particularly suitable for processing spatial-temporal data that is represented as graphs. This capability makes GNNs useful for discovering intricate patterns and abnormalities in Earth observation data.

a Temporal Graph Convolutional Network (T-GCN) was introduced as a method for predicting traffic patterns T-GCN utilizes graph convolutional layers to exploit the spatial and temporal relationships in traffic data, which is represented as spatio-temporal graphs. This allows T-GCN to capture the spatial correlations between distinct traffic regions and the temporal dependencies across time. Utilizing GNNs in T-GCN allows the model to efficiently acquire knowledge about intricate spatial and temporal patterns in traffic data, resulting in enhanced predictive capabilities.

In a similar manner, In [50] the researchers presented a Bi-directional Graph Recurrent Convolutional Network(Bi-GRCN), as a method for predicting spatio-temporal traffic flow. Bi-GRCN employs a graph neural network structure to capture the geographical relationships between traffic regions and the temporal changes in traffic flow over time. Bi-GRCN successfully captures the intricate spatio-temporal correlations in traffic data by combining graph convolutional and recurrent layers. This allows for precise prediction of traffic flow patterns.

2.2.2.3 Informer for Anomaly Score Generation

The Informer model for anomaly score generation refers to the use of the Informer architecture, a specific type of transformer-based neural network, to identify and score anomalies in time-series data[51], where the Informer architecture is a scalable and efficient transformer model designed for long-sequence time-series forecasting. Traditional transformers struggle with long sequences due to their quadratic complexity with respect to sequence length. In [51], the Informer tool was employed for the purpose of anomaly identification, where it had a vital function in producing scores for anomalies. The authors utilized Informer to calculate anomaly scores by measuring prediction errors or differences between real traffic data and model predictions. By utilizing the features of Informer, the authors sought to detect irregular patterns or abnormalities in traffic data. This method facilitated prompt identification and reaction to probable traffic problems or irregularities.

Similarly, in [52], the Informer model was shown as a highly effective transformer architecture specifically designed for accurately predicting long sequence time-series activities. By utilizing the predictive capabilities of Informer, anomaly scores can be computed by comparing anticipated and actual time series values for differences. The anomaly scores functioned as markers of atypical patterns or deviations from typical behavior in the time series data, thereby aiding in the detection of anomalies.

In a similar manner,In [49] the researchers utilized Graph Convolutional Networks (GCNs) to use the natural connections within traffic networks. Nevertheless, these techniques usually depend on fixed associations among network nodes, disregarding the fluid nature of traffic patterns. This research introduces a novel approach called Coupled Generative Graph Convolutional Network (CG-GCN) to overcome this drawback by effectively capturing the changing connections across various regions in the traffic network. The utilization of CGGCN has the potential to enhance the precision of traffic flow estimates in comparison to conventional GCN-based techniques.

2.2.2.4 Spatio-Temporal K-Nearest Neighbors (KNN)

KNN is an extension of the traditional KNN algorithm designed to handle data that varies across both space and time. Traditional KNN is a simple and widely used machine learning algorithm for classification and regression tasks, where the classification or regression is based on the ”k” nearest data points in the feature space. In [57] the authors presents a new method for detecting anomalies, which utilizes a modified version of the KNN algorithm in a spatio-temporal framework.

In [53] the authors highlights the constraints identified in techniques that only consider either the spatial or temporal aspects of traffic data separately.By utilizing the adapted KNN method, it is possible to successfully detect aberrant traffic patterns that differ substantially from the anticipated distribution of traffic movement at particular sites. this integrated strategy, which combines both spatial and temporal elements, improves the system's ability to identify unusual patterns in traffic data. This, in turn, leads to better traffic management and faster reaction to incidents.

The ”Multi-level Spatial-Temporal Fusion Neural Network” developed by [47] aims to enhance forecasting abilities by combining spatial and temporal information. Both methodologies in [47], [53] emphasize the significance of integrating spatial and temporal aspects in traffic flow prediction in order to attain more precise outcomes.

2.2.3 Multilevel Ensemble Learning for Enhanced Traffic Flow Prediction

Multilayer ensemble learning techniques provide an extra level of enhancement. Ensemble learning is a method that combines predictions from numerous models to get more reliable and precise results, in contrast to utilizing just one model [54], [58].

Multiple research has examined the efficacy of multilevel ensemble algorithms in predicting traffic flow. These studies highlight the potential of ensemble algorithms to improve prediction accuracy by utilizing a variety of models and efficiently merging their predictions.

- **Consensus Ensemble System**: A Consensus Ensemble System refers to a machine learning approach that combines multiple models to improve overall performance. The idea is that by aggregating the predictions of several models [59], the ensemble can produce more accurate and robust results than any single model. This technique leverages the diversity among the individual models to reduce the likelihood of errors and overfitting.

  In [55] the authors presents a specialized Consensus Ensemble System designed specifically for predicting traffic flow. The main objective of the system is to predict the flow of traffic on major roads many steps in advance. The work seeks to improve the practical usability of traffic flow predictions by employing ensemble learning, which combines the strengths of many prediction models.

- **Seamless Multilevel Ensemble Transform Particle Filter (SMETPF)**: The SMETPF is a sophisticated method used in data assimilation, combining aspects of ensemble-based filtering and particle filtering to improve the accuracy and efficiency of state estimation in complex systems, particularly those with high-dimensional state spaces. In [56] investigates the application of a SMETPF for the purpose of predicting traffic flow.

Table 2.2.1 provides a concise summary of various techniques used in traffic flow prediction and anomaly detection studies, along with key details and research gaps identified in each paper.

2.3 Clustering Based Approach for Enhanced Characterization of Anomalies in Traffic Flows

In the field of urban traffic management, it is crucial to detect anomalies in the frequency patterns of congestion in order to enhance traffic flow and minimize bottlenecks. These anomalies may suggest unusual occurrences such as accidents, road closures, or unforeseen fluctuations in traffic volume. By employing sophisticated data analysis methods, these outliers can be identified by continuously monitoring congestion data in real-time. After identification, clustering techniques are utilized to categorize related abnormalities, enabling a more methodical and effective approach to traffic disruption management. This approach not only improves the capacity to promptly address current traffic problems but also facilitates long-term strategic planning and improvement of urban transportation networks. The article examines recent studies on the prediction of traffic patterns and the identification of anomalies. It utilizes clustering techniques to segment these detected anomalies. Furthermore, it examines the ability of these techniques to improve understanding of their spatial and temporal patterns, which is essential for efficient traffic control and urban development. Current research is focused around incorporating advanced methods, such as improved Gated Recurrent Unit(GRU) models, spatiotemporal clustering, and joint clustering and prediction approaches, to forecast traffic patterns, identify anomalies, and divide these anomalies using clustering algorithms in intelligent transportation systems. To ensure a thorough comprehension, we have categorized previous studies according to the approaches utilized and the issues tackled. Additional information is provided in the subsequent sections.

2.3.1 Traffic Anomaly Detection

Traffic Anomaly Detection using clustering-based methodologies use algorithms like K-means, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and EFMS-Kmeans to detect and analyze abnormal traffic patterns, hence improving the accuracy and efficacy of anomaly detection in traffic systems.

[60] propose a traffic anomaly detection method based on an improved GRU for traffic prediction and EFMS-Kmeans clustering. This approach combines the techniques of traffic prediction and clustering to improve the accuracy and efficiency of anomaly detection.

[61] proposes a marine traffic anomaly detection approach by combining the DBSCAN clustering algorithm with k-nearest neighbors analysis. This method efficiently detects anomalies in vessel traffic using historical marine data by utilizing spatial clustering algorithms.Furthermore, in the domain of condition monitoring in a marine engine system, researchers have investigated the use of cluster-based anomaly detection techniques. These techniques involve the application of algorithms such as K-means clustering, Mixture of Gaussian models, density-based clustering, self-organizing maps, and spectral clustering.

[62] compares two clustering methods (density-based and representative-based), for the purpose of detecting anomalies in traffic data. The study seeks to enhance the detection of congestion, accidents, or other traffic difficulties by categorizing data points based on density and selecting representative points within clusters to identify normal traffic patterns and identify substantial departures as anomalies.

Traffic Anomaly Detection using Graph Neural Networks utilizes the interconnected nature of traffic data to recognize unusual patterns, hence enhancing the accuracy and promptness of traffic detection.

[63] presents a new method for detecting traffic anomalies by utilizing a graph autoencoder with mirror temporal convolutional networks. This technique utilizes graph structures to capture the relationships between traffic data points, while the mirror temporal convolutions address the time-dependent character of network flow. The model's objective is to find unusual patterns in encoded traffic data by comparing it to the original data. This can potentially enhance the identification of

traffic anomalies such as congestion or accidents.

[64] proposes an approach that employs spatial-temporal graph neural networks. These networks have the ability to acquire knowledge from the spatial relationships between road segments (spatial) and the fluctuations in traffic patterns over time (temporal). The method seeks to automatically find unusual traffic patterns by examining the acquired representations of the traffic data. This can potentially enhance the identification of congestion, accidents, or other anomalies on the road network.

In [65], a cluster-guided denoising graph auto-encoder (CG-DGAE) for traffic data imputation and fault detection. The method leverages spatial-temporal context to achieve 99.09% accuracy in fault detection, enhancing data integrity in sensor networks is presented.

2.3.2 Traffic Prediction and Forecasting

For traffic prediction and forecasting employ spatiotemporal clustering algorithms to analyze patterns of spatial and temporal data. The goal is to develop accurate predictions about future traffic conditions and enhance traffic management.Clustering-based methods in traffic prediction and forecasting utilize algorithms to analyze traffic patterns and improve the accuracy of traffic flow and travel time forecasts. [66] propose a high-performance traffic speed forecasting approach by spatiotemporal clustering of road segments. This method utilizes non-parametric clustering to accurately predict traffic patterns based on traffic data dynamics.[67] presents a prediction model for short-term traffic flow. The approach combines K-means clustering with a GRU. This model seeks to enhance the accuracy of short-term traffic flow forecast by analyzing different traffic flow patterns.[68] propose a method that combines clustering and prediction to accurately estimate travel times. This methodology effectively models real-world traffic scenarios by forming clusters based on their travel times.

2.3.3 Anomaly Detection in Various Domains

Anomaly Detection in Various Domains employs sophisticated techniques to detect abnormal pattern in several sectors, including industrial systems, network traffic, and maritime operations, thereby enhancing safety and operational effectiveness.

[69] investigates the utilization of clustering algorithms to detect anomalous data from sensors installed on a ship's engine, which monitor aspects such as temperature and pressure. The approach involves the clustering of similar sensor readings into clusters, which are then used to compare new readings. If a new reading measurement considerably diverges from the existing clusters, it is identified as an anomaly, which could suggest a problem in the engine. This technique allows for ongoing surveillance and prompt identification of possible problems, so preventing failures and enhancing the overall well-being of the engine.

[70] investigates the application of Graph Neural Networks (GNNs) in identifying abnormal patterns in industrial environments. GNNs utilize the interconnected structure of Industrial Internet of Things (IIoT) systems, in contrast to traditional approaches that evaluate sensor data from individual units.GNNs can analyze data from individual devices and the relationships between them by modeling equipment and sensors as nodes in a graph. This methodology captures intricate interconnections and detects anomalies that could be overlooked by analyzing individual sensor measurements. As a result, GNNs can increase the accuracy of anomaly detection and allow for earlier identification of problems, hence enhancing the reliability and efficiency of the system.

[71] introduces a novel method for detecting anomalies in network traffic. It combines two methodologies. The X-means clustering algorithm has been enhanced to automatically estimate the best number of traffic patterns. Additionally, the Isolation forest algorithm has been developed to rapidly identify anomalies by isolating them from the data.

[72] proposes a technique for consistently detecting abnormal network behavior. This approach is highly likely to adjust to evolving network activity over time, making it well-suited for real-world scenarios. Traditional anomaly detection techniques may have difficulties in detecting changing network patterns. To fully understand the model, one would need to refer to the complete study. However, it is probable that the model employs strategies that can acquire knowledge and adapt to new network data, enabling it to accurately identify anomalies in dynamic network settings.

[73] investigates a unique technique for identifying anomalies in network traffic. Instead of depending on traditional statistical approaches, it utilizes principles from catastrophe theory. This theory examines how systems with multiple variables can undergo sudden, drastic shifts in behavior. The study proposes that this theory can be utilized for analyzing network traffic data. The method

seeks to discover locations of considerable deviation from usual patterns in network traffic by examining specific properties of the data that are related to catastrophe theory. This has the potential to enable the detection of abnormalities that could be missed by traditional methods.

[74] present a methodology for Anomaly Detection in IIoT environments using Graph Neural Networks. The objective of this framework is to enhance the identification of abnormal behavior in IoT contexts by offering capabilities for explainable artificial intelligence.

In [75] an Advanced Driver Assistance Systems (ADAS) by utilizing Explainable AI (XGBoost) to detect driver abnormalities and evaluate driver behavior. This method promotes road safety by recognizing unsafe activities and provides interpretable insights for real-time anomaly detection in transportation systems is presented.

[76] improve the dissemination of information in large-scale vehicular networks by forecasting traffic patterns in specific geographical areas, such as traffic hotspots, in order to boost the efficiency of the network.

A cluster of anomalies in intelligent transportation systems is a crucial factor that greatly affects the efficiency and safety of the transportation system. Anomaly detection is crucial for mitigating congestion, improving safety, and offering useful insights for traffic prediction and road infrastructure planning. [77].

The studies together enhance anomaly detection approaches in intelligent transportation systems by stressing the use of sophisticated techniques such as graph neural networks and explainable AI frameworks for efficient anomaly detection and analysis.

Table 2.3.1 provides a summary of the related work, highlighting the strengths and drawbacks of various traffic anomaly detection, clustering, and prediction techniques.

Table 2.2.1: Comprehensive Analysis of Traffic Prediction Techniques and Anomaly Detection Studies

Methods	Strength	Drawback
LSTM [34], [35], [38], [47]	Effective Handling of Time Series Data, Potential for High Accuracy, Applicability in Real-World Scenarios	Data Dependency and Computational Complexity, Overfitting, Limited Generalization
RNNs for long-term traffic flow [36], [39]	Potential for Capturing Complex Patterns, Applicability to Urban Environments	Computational Complexity, Generalization to Diverse Conditions
SRCNs [37]	Ability to Handle Multidimensional Data, Robustness to Noise and Variability	Interpretability, Generalization Across Different Networks
Causal sliding windows [44]	Applicability to Various Domains	Computational Overhead
Uneven diffusion model [45]	Practical Implications	Model Complexity, Generalization
T-GCN [48], [49]	Robustness to Dynamic Traffic Patterns	Computational Complexity
Deep Convolutional Neural Network (CNN) integrated with an Ensemble Learning Model[41]	Effectively detects pedestrians under varying conditions	Requires significant processing power due to the complexity of multi-modal data and ensemble models
Dynamic Factor Model (DFM) [42]	Effectively forecasts traffic performance over multiple future time steps	The DFM requires significant computational resources for processing large-scale urban traffic data
Bi-GRCN [50]	Integration of Temporal Components	Model Complexity, Interpretability
Informer model [51], [52]	Applicability in Intelligent Transport Applications	Dependency on Training Data, Interpretability
KNN algorithm [53]	Simple Implementation	Scalability, Imbalanced Data
EEMD-ANN method [54]	Multiscale Prediction, Adaptability to Varied Time Scales	Complexity and Computational Cost, Generalization Across Traffic Conditions
Ensemble learning approach [55]	Robustness, Adaptability to Diverse Traffic Conditions	Complexity, Training and Maintenance Costs
ETPF [56]	Adaptability to Nonlinear and Non-Gaussian Systems, Efficient Sampling and Resampling	Computational Complexity, Sampling and Propagation Errors
Gaussian Process Regression (GPR),Stochastic Differential Equations (SDE)[46]	Effectively identifies anomalies in traffic flow data, improving smart mobility systems	GPR and SDE are computationally intensive, especially for large datasets

Table 2.3.1: Comprehensive Analysis of Traffic Anomaly Detection, Clustering, and Traffic Prediction Techniques

Methodology	**Strength**	**Drawback**
Improved GRU for traffic prediction and EFMS-Kmeans clustering [60]	Combines traffic prediction with clustering for enhanced accuracy and efficiency.	May require significant computational resources.
DBSCAN with k-nearest neighbors analysis [61]	Efficiently detects anomalies in vessel traffic using spatial clustering algorithms.	Limited by quality of historical data.
Density-based clustering [62]	Enhances detection of congestion and other traffic difficulties.	May struggle with high-dimensional or complex traffic patterns.
Data imputation and fault detection using a Cluster-guided Denoising Graph Auto-Encoder (CDGAE)[65]	The CDGAE demonstrates superior performance in traffic data imputation compared to existing methods, especially in scenarios with missing data and noise	The CDGAE, like other deep learning models, is susceptible to overfitting, especially when trained on limited data
Graph autoencoder with mirror temporal convolutional networks [63]	Captures relationships between traffic data points and addresses time-dependent network flow.	Complex to implement and interpret.
Spatial-temporal graph neural networks [64]	Gains knowledge from spatial and temporal fluctuations, improving anomaly detection.	High computational complexity and data requirements.
Spatiotemporal clustering for traffic speed forecasting [66]	Accurately predicts traffic patterns based on data dynamics.	Non-parametric methods may not generalize well.
K-means with GRU for short-term traffic prediction [67]	Enhances short-term traffic flow accuracy.	K-means may not handle non-linear patterns well.
Clustering and travel time estimation [68]	Effectively models real-world traffic scenarios by clustering travel times.	Requires extensive historical data.
Clustering for marine anomaly detection [69]	Allows surveillance and prompt identification of problems.	Sensor data quality affects anomaly detection accuracy.
Proposes an Explainable Artificial Intelligence (XAI) approach to enhance Advanced Driver Assistance Systems (ADAS) by detecting driver anomalies [75]	The XAI-based approach can improve the safety of ADAS by proactively detecting and responding to driver anomalies	raining and deploying deep learning models can be computationally expensive, especially for real-time applications
Graph Neural Networks in IIoT [70], [74]	Analyzes data from individual devices and relationships.	High complexity and scalability issues.
Enhanced X-means and Isolation Forest for network anomalies [71]	Estimates best traffic patterns and rapidly detects anomalies.	Sensitive to parameter tuning.
Adaptive model for network behavior [72]	Adapts to evolving network activity, suitable for real-world scenarios.	Challenges in model training and adjustment periods.
Catastrophe theory for anomaly detection [73]	Analyzes locations of major deviation from usual patterns.	Complex theory, hard to implement practically.
Forecasting traffic patterns in vehicular networks [76]	Improves efficiency by forecasting traffic patterns.	Requires extensive data for accuracy.

Chapter 3

Intelligent Meta-Heuristic Based Optimization of Traffic Light Timing Using Artificial Intelligence Techniques

3.1 Introduction

The rapid increase in the number of automobiles has contributed to congestion, pollution, and logistical delays. The financial toll of gridlock is rising. Specifically, Forbes reported in [78] that traffic congestion costs the United States economy $124 billion annually. Around one percent of the Gross Domestic Product (GDP) [79] of the European Union is lost due to traffic congestion costs. A major obstacle to creating smart cities is the exponential rise in vehicle ownership despite inadequate public transportation facilities. Increased traffic congestion and fuel costs are just two of the numerous negative consequences of dense vehicle numbers, which also include air and noise pollution, stress and disease, accidents, and high fuel costs. An increase in the number of individual vehicles is a result of the development of a nation on the other side of the globe. This has led to heavier traffic in major urban centers. yearly, traffic jams get worse in Indonesia. Indonesia, which in 2015 ranked No. 11 on the list of the world's most jammed countries behind only Brazil and Argentina, had an estimated jamming time of 40.58 minutes on average. With a time index of 49.44 minutes in 2017, the position moves up to second in 2017 [80]. We need a better traffic management system as a result. Time and money could be saved with efficient and trustworthy traffic management and control. With the advent of the IoT, we are one step closer to completely automated, highly controlled processes and systems. In order to detect, gather, and transmit data, IoT-based ITM sensors are put in autonomous vehicles and smart devices. Machine Learning (ML) has the potential to improve transportation in other ways as well. Traffic congestion, delays, and fatalities are all too common due to the various flaws in the current transportation management systems. Long waiting times, unnecessary fuel consumption, and rising carbon emissions are only some of the problems with the current vehicle traffic light signal control system. Drivers experience a great deal of stress as a result, and emergency and other high priority vehicles, are delayed in their arrivals. Congestion can be alleviated through the smart management of traffic lights. Controlling traffic lights intelligently is essential for a smooth running transportation network [81]. An intelligent traffic signal control system would automatically adapt to the current flow of traffic, as opposed to the static regulations used by the current system. Most traffic signals today are still programmed to operate on a predetermined schedule [82], [83] rather than being based on observations of actual traffic patterns. Some recent research has proposed custom-made criteria based on actual traffic data [84], [85] . Unfortunately, these rules remain statically specified, making them inflexible in the face of fluctuating traffic conditions. Existing research, however, has not yet validated the methodology using real-world traffic data and has instead focused solely on analyzing incentives without interpreting policy. An improved deep reinforcement learning model for managing traffic lights is proposed in this paper. To ensure the efficacy of our method, we use a massive collection of actual traffic data captured. Intriguing case studies of policies gleaned from actual data are also presented. Reinforcement learning [4]–[6] has been used to dynamically adapt traffic lights to current traffic conditions. Both representing the environment and modeling the relationship between the environment and the decision provide significant obstacles to the practical use of traditional reinforcement learning. In order to overcome these obstacles, researchers [5], [86] have recently applied deep reinforcement learning methods, such as Deep Q-Learning (DQN), to the traffic light management problem. Many reasons have urged us to study the traffic congestion problem, since it might affect various sectors in our daily life. The first incentive is to reduce the congestion in the road network and hence the waiting time for roads users (drivers and pedestrians), which will directly affect each user productivity in their daily life, it might be a small effect for each individual alone but if we examine the change in a holistic manner, it will be a huge effect on a society level. Another motivation is to decrease the CO gas emissions from the vehicles, which is also have a massive impact on one of the main problems in the world at the current time, which is the destruction of the ozone layer that directly affects people's health if not solved. In addition to that, a reduced fuel consumption is achieved which gives scientists a longer time to look for other options such as renewable energy to replace the current energy sources that will run out in the near future. The congestion mitigation will also drastically increase the roads user satisfaction and the quality of life, decrease the number of car accidents which saves lives around the world [14]. Many research papers are using the meta-heuristics techniques to solve different problems in different fields, the researchers in [87] introduces a Two-level Particle Swarm Optimization (TLPSO) for managing credit portfolios, with the goal of reducing losses while staying below financial limits. Comparative investigations show that TLPSO, with its novel dual searching mechanism, outperforms conventional methods like the Genetic Algorithm and the Particle Swarm Optimization. Another paper [88] presents a strategy for improving disaster response logistics by balancing facility placement and transportation routes. Discrete Particle Swarm Optimization and Harris Hawks Optimization are combined into a new hybrid algorithm that is introduced to solve this difficult issue. A COVID-19 case study conducted in Wuhan validates the method's efficacy, showing that it is both more accurate and more efficient than alternative approaches. Other problems have used meta-heuristic algorithms to find near optimal solution [89], [90]. According to No Free Lunch (NFL) theory, there is no single meta-heuristic that is suitable for any problem, that is why in this paper, we investigated many meta-heuristics and compared them together to determine which one is better as shown in the results section below.

After studying the literature of intelligent traffic lights, many research gaps have been discovered as explained in more details in the related work section below, the problems that we are going to address are briefly summarized as follows:

- Increased road traffic congestion due to the increasing numbers of road users and the limited capacity for the current road infrastructure that could not be extended easily, where this main problem could cause many subproblems such as increased waiting time, increased number of accidents and increased levels of CO and CO2 gas emissions[91].

- Current problem exists in the proposed meta-heuristics algorithms in the literature concerning both the execution time and convergence speed, which are two important factors that should be minimized/improved when designing a meta-heuristic algorithm. This will help us in making faster decisions and better solutions that are more near to the optimal solution.

- Current problem exists in the proposed machine learning models in the literature when it comes to the prediction accuracy of the solution. Better predictions will enable us to make more wise future decision and help solve the problem before occurring. This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

The selected algorithms in our proposed solution are renowned for their ability to achieve a balance between exploration (seeking out new regions) and exploitation (improving known good areas), which is essential in traffic light optimization where both new solutions and improvement of old solutions are required. Additionally, they provide a favorable combination of randomization and deterministic principles, which aids in adjusting to the unpredictable characteristics of traffic patterns. The improved BAT algorithm, specifically, may provide a distinct benefit in terms of speed and convergence, rendering it more appropriate for real-time applications such as traffic signal optimization.

3.2 Motivation and Preliminaries

Many reasons have urged us to study the traffic congestion problem, since it might affect various sectors in our daily life. The first incentive is to reduce the congestion in the road network and hence the waiting time for roads users (drivers and pedestrians), which will directly affect each user productivity in their daily life, it might be a small effect for each individual alone but if we examine the change in a holistic manner, it will be a huge effect on a society level. Another motivation is to decrease the CO and CO2 gas emissions from the vehicles, which is also have a massive impact on one of the main problems in the world at the current time, which is the destruction of the ozone layer that directly affects people's health if not solved. In addition to that, a reduced fuel consumption is achieved which gives scientists a longer time to look for other options such as renewable energy to replace the current energy sources that will run out in the near future. The congestion mitigation will also drastically increase the roads user satisfaction and the quality of life, decrease the number of car accidents which saves lives around the world.

3.3 Background

3.3.1 Prediction Algorithms

3.3.1.1 Recurrent Neural Network (RNN)

Traditional neural network is class of artificial neural networks (ANN) and are a series of algorithms which is very much closely and derived from human brain working.where they interpret sensory data whether historical or live data (offline or online data) by a type of machine perception, labeling or clustering raw input to recognize the patterns. An ANN usually contains many processors which is working in parallel and arranged in layers. Where the first layer receives the input information. And each layer receives the output from the layer that preceding it [92]. Recurrent Neural Network (RNN) is type of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input.

As shown in 3.1.it takes the X(0) from the sequence of input and then it outputs h(0) which together with X(1) is the input for the next step. So, the h(0) and X(1) is the input for the next step. Similarly, h(1) from the next is the input with X(2) for the next step and so on. Due to can be difficult to train RNNs to solve problems which need learning long-term temporal dependencies and this their reason that the gradient of the loss function vanish exponentially with time (called the vanishing gradient problem) where RNN has short term memory so the LSTM is used to solve this

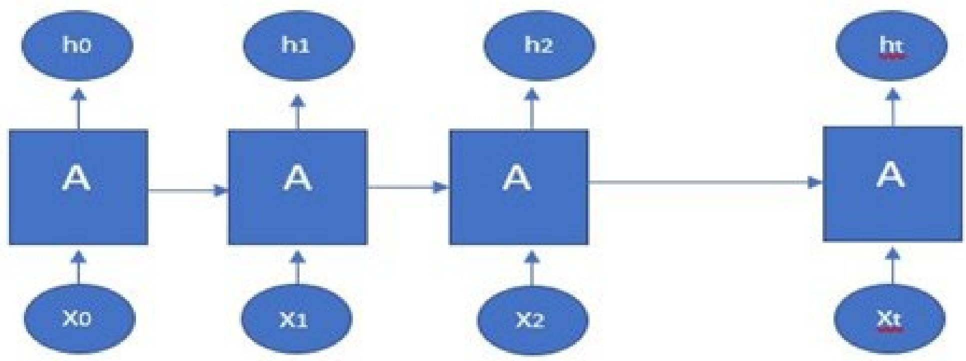


Figure 3.1: The Recurrent Neural Network.

problem.

3.3.1.2 Long Short Term Memory (LSTM)

LSTM is similar to control flow in RNN and it has internal mechanisms called gates that can regulate the flow of information and these gates can learn which data in a sequence is important or not to keep it or throw it way. As shown in 3.2 the LSTM from three gates: forget gate, input gate, and output gate. Forget gate which used to decide which information keep or forget.information from a previous hidden state and information from the current input is passed through the sigmoid function that decide which values will be updated by transforming values to be between [0,1],values between [0,1] the closer to 0 will be removed and values closer to 1 will be kept [93].

Cell state acts as transport highway that transfer relevant information all the way down to the sequence chain and it can be considered as memory of the network.

3.3.1.3 Decision Tree

A decision tree algorithm is a popular machine learning technique used for both classification and regression tasks. It's a type of supervised learning algorithm that takes a set of input features and produces a decision tree as its output. This decision tree is essentially a flowchart-like structure where each internal node represents a decision based on a particular feature, each branch represents the outcome of that decision, and each leaf node represents a class label or a predicted value[94].

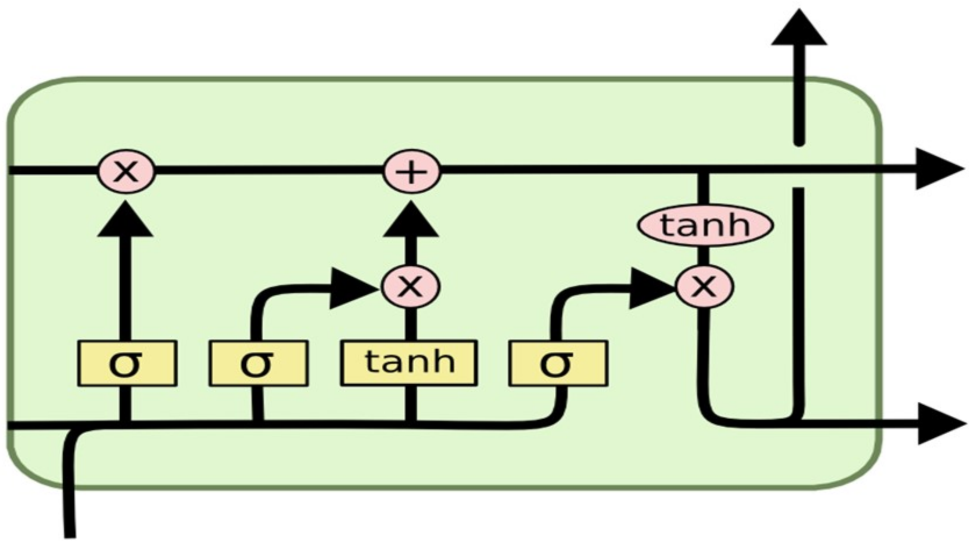


Figure 3.2: The Long Short Term Memory.

The primary goal of a decision tree algorithm is to create a model that can make accurate predictions or classifications by learning patterns and relationships within the input data. Here's a simplified overview of how a decision tree algorithm works:

1. **Data Splitting:**

   The algorithm starts with the entire dataset at the root node of the tree.

2. **Feature Selection:**

   It selects the best feature from the dataset to split the data into subsets. The "best" feature is typically chosen based on criteria that aim to maximize the separation between different classes or minimize the variance within each subset. Common criteria include Gini impurity, entropy, or mean squared error.

3. **Recursive Process:**

   The algorithm recursively applies the same process to each subset of data, creating child nodes for each split. This process continues until a stopping condition is met. This condition could be a certain depth of the tree, a minimum number of samples in a node, or other similar criteria.

4. **Leaf Node Labeling:**

   Once the recursive process is complete, the leaf nodes are assigned class labels (for classification tasks) or predicted values (for regression tasks), usually based on the majority class in a classification problem or the average value in a regression problem.

5. **Prediction:**

   To make predictions for new data, you start at the root node and traverse the decision tree by following the path that corresponds to the feature values of the new data. You end up at a leaf node, and the class label or predicted value associated with that leaf node becomes the model's prediction.

3.3.1.4 Auto-Regressive Integrated Moving Average(ARIMA)

The AutoRegressive Integrated Moving Average (ARIMA) is a popular time series forecasting approach. It is well suited to time series data such as stock prices, sales figures, and weather patterns since it predicts links between observations and lagged values.

3.3.1.5 Seasonal ARIMA(SARIMA)

SARIMA (Seasonal ARIMA), an extension of ARIMA, can be used to model seasonal patterns in time series data. It adds additional terms to model seasonal fluctuations, increasing prediction precision for repeating data.

3.3.2 Meta-heuristics Algorithms

3.3.2.1 Phases in Meta-heuristics Algorithms

The phrases ”exploration” and ”exploitation” refer to two basic features of the behavior of meta-heuristic algorithms like evolutionary algorithms, swarm intelligence, and simulated annealing as they search for optimal solutions in a complex search space.

- **Exploration:** First, we have exploration, which is the process of examining a large portion of the solution space for novel and varied solutions. The algorithm places a premium on trying out fresh solutions, even if they don't look promising at first. By searching for other regions of the solution space, exploration aims to prevent getting stuck in local optima (suboptimal solutions). In other words, the goal of exploration is to keep the population of solutions diverse and to stop it from settling too quickly into a suboptimal zone.

- **Exploitation:** The second strategy, "exploitation," is a thorough investigation of proven methods in order to hone and enhance them. The algorithm's focus throughout the exploitation phase is on using the insights acquired from previously found optimal solutions to further enhance and perfect them. This is usually done through local search techniques that center on making minor adjustments close to probable solutions. The goal of exploitation is to improve upon already promising solutions so that they converge on the optimal solution.

3.3.2.2 BAT Meta-heuristic algorithm

The Bat Algorithm (BA) is an optimization technique that mimics the way bats use echolocation. In 2010, Xin-She Yang introduced it as a metaheuristic optimization technique for dealing with difficult optimization issues. Applications of BA can be found in many different fields, including engineering, economics, medicine, and more due to BA's versatility and effectiveness with continuous and discrete optimization issues.

The Bat Algorithm, at its core, is a simulation of bat foraging behavior; each "bat" stands for a possible answer to an optimization issue. The program is modeled after the way bats fly by sending out ultrasonic pulses, listening for echoes, and altering their flight path accordingly. A series of mathematical equations and operators are developed to include this behavior into the optimization procedure.

Each bat in the Bat Algorithm's population represents a possible solution, and the algorithm works by moving these bats throughout the solution space. In order to identify the best possible or nearly best possible solution, the algorithm iteratively improves upon these candidates. The main parts and procedures of the Bat Algorithm are as follows:

- The algorithm begins by creating a population of bats with uniformly distributed initial positions in the solution space (referred to as ”initialization”). Each bat has a unique pulse emission rate and loudness value that influences how it flies.

- Emission and Movement: From their current locations, bats emit ultrasonic pulses, with the intensity of the pulse decreasing with each repetition. Bats make course corrections using information from the radiated pulses as well as their own past positions and velocities. By flying about, bats can test out many strategies and eventually settle on the most effective one.

- Pulse Frequency and Velocity: A bat’s search radius surrounding its current position is determined by the frequency of its pulses. More global exploration is encouraged by bats with higher pulse frequencies, whereas local exploitation is prioritized by those with lower frequencies. Each bat’s speed is revised every time its pulse frequency or volume is detected.

- Local Search and Global Search: When looking for interesting regions in the solution space, bats undertake local search around their current places. They also alter their speeds in the direction of the greatest solution identified so far in the population, which aids in worldwide research.

- The Loudness and Pulse Frequency of each bat are adjusted to reflect their current performance. Bats with better solutions have their volume and frequency settings preserved, while bats with worse solutions have their settings lowered to promote more research.

- At each iteration, the objective function of the optimization problem is used to assess the fitness of each bat’s solution. Bats who come up with superior solutions contribute to improving the overall best one.

- Termination: The algorithm repeats this process until a convergence condition is fulfilled or a predetermined number of iterations have passed. In the end, you’ll have the optimal solution discovered by any bat.

Because it strikes a good balance between exploration and exploitation, the Bat Algorithm can be applied to a wide variety of optimization problems. Its flexibility and durability come from

Algorithm 1 Bat Algorithm

Data: Objective function $f(x)$, Population size N, Number of generations G, Frequency scaling factor α, Pulse emission rate γ, Lower bound L, Upper bound U

Result: Optimal solution x^*

Initialize population X with random solutions Initialize pulse rates r_i and loudness A_i for each bat Initialize best solution x^*

```
for t ← 1 to G do
    for i ← 1 to N do
        Generate a new solution x_i by adding random step to X_i  if rand() < r_i then
            Generate a new solution x_i by adding a random step and a fraction of the current best solution
        end
        Evaluate the fitness of x_i  if rand() < γ and f(x_i) < f(x*) then
            Accept x_i as the new best solution x*
        end
        Update r_i and A_i using equations
    end
end
```

the fact that it can dynamically modify bats' search behavior via pulse frequency and loudness adjustments. The efficiency of the Bat Algorithm, like that of any optimization algorithm, might shift based on the details of the problem and the values chosen for its parameters. Researchers are still looking at new ways to tweak and refine the algorithm so that it can more quickly and accurately address difficult optimization challenges.

The Bat Algorithm (BA) is a nature-inspired optimization algorithm that mimics the echolocation behavior of bats. It is often used for solving optimization problems. The key equations of the Bat Algorithm are as follows:

1. **Position Update**

   The position of each bat is updated based on its current position, velocity, and the pulse rate of the bat. The updated position x_i of the i-th bat is given by:

$$x_i^{t+1} = x_i^t + \epsilon \cdot (x_{\text{best}} - x_i^t) + \alpha \cdot A \cdot \text{rand}() \tag{1}$$

   Where:

   - x_i^{t+1} is the updated position of the i-th bat at time $t+1$.

- x_i^t is the current position of the i-th bat at time t.
- x_{best} is the current best solution found by any bat.
- ϵ is a random scaling factor.
- α is the loudness of the bat.
- A is the pulse rate of the bat.
- rand() generates a random number between 0 and 1.

2. **Velocity Update**

   The velocity v_i of each bat is updated to move towards the updated position. The velocity update equation is given by:

$$v_i^{t+1} = v_i^t + (x_i^{t+1} - x_i^t) \tag{2}$$

   Where:

   - v_i^{t+1} is the updated velocity of the i-th bat at time $t+1$.
   - v_i^t is the current velocity of the i-th bat at time t.

3.4 Methodology

As shown in the Figure 3.3 The Road Congestion Prediction procedure begins with Phase 1. The initial step is data gathering, which includes recording information about things like traffic volumes, vehicle speeds, road occupancy, and other characteristics over time. Different prediction methods are implemented on top of this dataset. Data collection and processing algorithms like Recurrent Neural Networks, Long Short-Term Memory networks, Decision Trees, AutoRegressive Integrated Moving Average, and Seasonal ARIMA are used to forecast traffic congestion. The accuracy of these prediction algorithms is then assessed by comparing the projected values to real-world congestion data and use metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE). After the models have been verified, they are used to make predictions about traffic congestion, which can provide light on how the situation will unfold in the future. Adaptability to shifting traffic circumstances is ensured by repeating this forecasting method throughout several instances or time periods (N instances)[95]. The results of these forecasts are saved in a special Congestion Database, which is subsequently used for analysis and decision-making in the future. In the second phase, metaheuristic algorithms are used to improve traffic flow. The green time for traffic lights at various junctions can be strategically adjusted with the help of algorithms like the BAT Algorithms, Particle Swarm Optimization, Cuckoo Search algorithm, and the Jaya algorithm. The ultimate goal is to reduce the amount of time cars spend waiting at these junctions. Congestion will be reduced, traffic flow will be improved, and a more effective transportation system will be created thanks to this optimization process. When Phase 1 predictions are combined with Phase 2 optimization efforts, a unified system is produced that can not only anticipate congestion but also take proactive steps toward its reduction by optimizing traffic signal timings. Transportation systems, in general, and traffic management in particular, stand to benefit greatly from this all-encompassing strategy. The success of both stages is dependent on the accuracy of the data acquired, the efficacy of the prediction algorithms, and the efficacy of the optimization process.

3.5 Experimental Setup and Evaluation

The experimental setup evaluates the proposed methodology, which combines traffic congestion prediction using machine learning models and traffic signal optimization via metaheuristic algorithms. In the first phase, real-world traffic data is used to train models such as RNN, LSTM, Decision Trees, ARIMA, and SARIMA, with predictive accuracy assessed through metrics like MSE and MAE over multiple time periods. In the second phase, the predictions guide the optimization of traffic light timings using algorithms like BAT, PSO, Cuckoo Search, and Jaya, aiming to reduce vehicle waiting times and improve traffic flow. The system's effectiveness is analyzed by comparing prediction accuracy, congestion reduction, and computational efficiency, providing a comprehensive evaluation of the integrated approach.

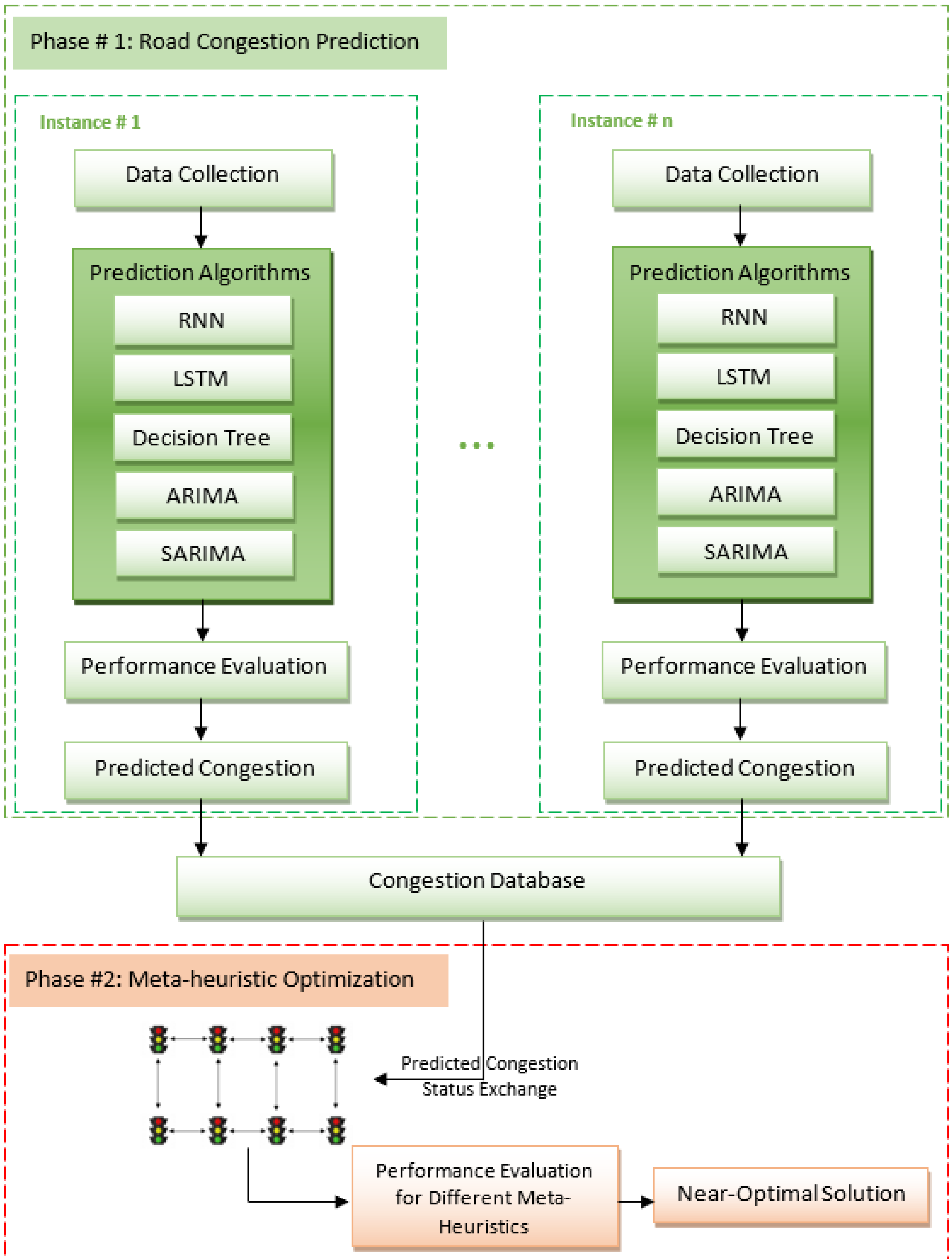


Figure 3.3: The Proposed Methodology

3.5.1 Performance Metrics

In this study, we evaluated the prediction precision using various commonly used metrics. As outlined in [96], each metric offers a unique perspective on the prediction's accuracy and the errors' dispersion. Below are more details about each metric, assuming that $\hat{y}_i$ represents the predicted value for the i-th pattern, y_i represents its actual value, and n represents the size of the data set.

3.5.1.1 Mean Absolute Error

Mean Absolute Error(MAE) metric takes the magnitude of the difference between the predicted and actual values, regardless of whether it is an overestimation (positive difference) or an underestimation (negative difference). In addition, the units of MAE are the same as the units of the examined data, which makes it easy to understand the average amount of error. This metric can be calculated using the following equation.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

3.5.1.2 Mean Squared Error

Mean Squared Error(MSE) is a metric that computes the average squared difference between the predicted and actual values. This error metric penalizes more significant errors, making it more sensitive to outliers than the MAE metric. It also provides information about the model's bias, as a consistently positive MSE can be a clue. However, it does not definitively tell the direction of the bias (overestimation or underestimation). This error metric is calculated as follows.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

3.5.1.3 Root Mean Squared Error

Root Mean Squared Error(RMSE) is derived from MSE by calculating the square root of the mean of the squared mistakes. RMSE, like MSE, is similarly susceptible to outliers because of the squaring operation. However, it presents the error in the same units as the original data, facilitating the assessment of the error's magnitude. Mathematically, it is represented as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

3.5.1.4 Coefficient of Determination

Coefficient of Determination (R^2) is metric reveals the model's goodness of fit to the data. It is a quantified score that represents the proportion of variance in the dependent variable explained by the independent variables. The value of *1* signifies a perfect prediction, which means that the model explains all the variance in the dependent variable. The value of *0* indicates that the model does not improve the prediction over the mean of the target value, which means that the model does not explain any of the variances. These characteristics make this metric a valuable score for evaluating the performance of regression models. However, using R2 alone is not enough to evaluate the quality of a regression model. It is essential to use other metrics, such as MAE and MSE, alongside R2 to get a complete picture of the performance of the regression models. This quantified metric is calculated using the following equation.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

3.5.2 Zone Based Analysis

As discussed earlier,there are many options for studying the road congestion problem. The first option is to study this problem in a local manner, which means that each traffic light only considers the congestion level at its intersection without any consideration of the congestion level at other traffic lights, and adjust its green timing for each traffic light accordingly. The other option, is to have a global view of all the congestion levels at each traffic light and then adjust the green timing for each traffic light according to the local congestion level at the specific traffic light and according to the congestion level at other traffic lights in the city of Montreal as shown in 3.4. In our study, we divided the simulation area for different zones and studied our problem for each zone separately, each traffic light should adjust their green timing dynamically according to the congestion level at this specific traffic light and the congestion levels at all the neighboring traffic lights in the same

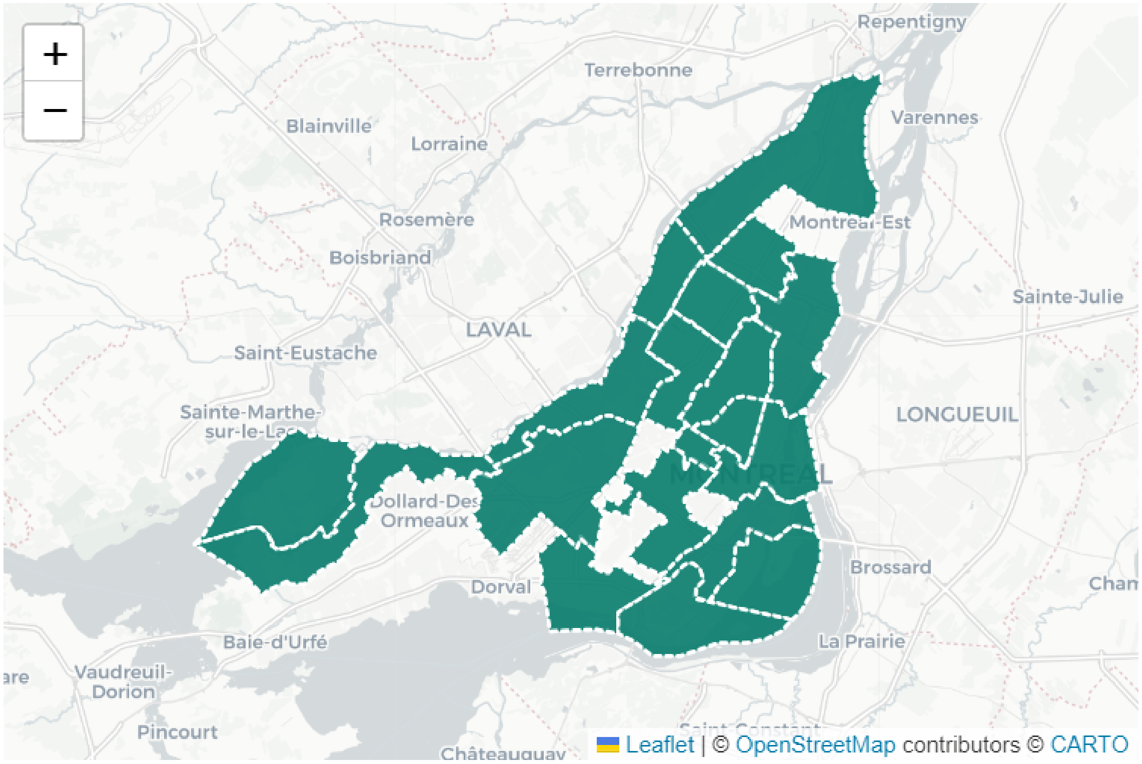


Figure 3.4: Simulation area in Montreal [97].

zone.

No matter how big the network is, even if it is a large-sized networks, it will be divided into zones and the prediction algorithm will be separately applied on each zone, in our simulation each zone will have 19 traffic lights in it, this makes our proposed algorithm scalable.

3.6 Results and Analysis

3.6.1 Result for the Road Congestion Prediction(Phase 1)

We divided our extensive case study into several zones, each of which contained 19 traffic signals. Our focus was on using advanced machine learning algorithms to predict and address congestion before it occurred. This proactive strategy enabled us to make educated judgments based on the algorithms' forecasts. As shown in the 3.5, we applied our methodology and demonstrated the results to a smaller zone containing 4 traffic lights. This allowed us to focus on a more detailed analysis of traffic behavior and validate the effectiveness of our predictions and optimizations in a

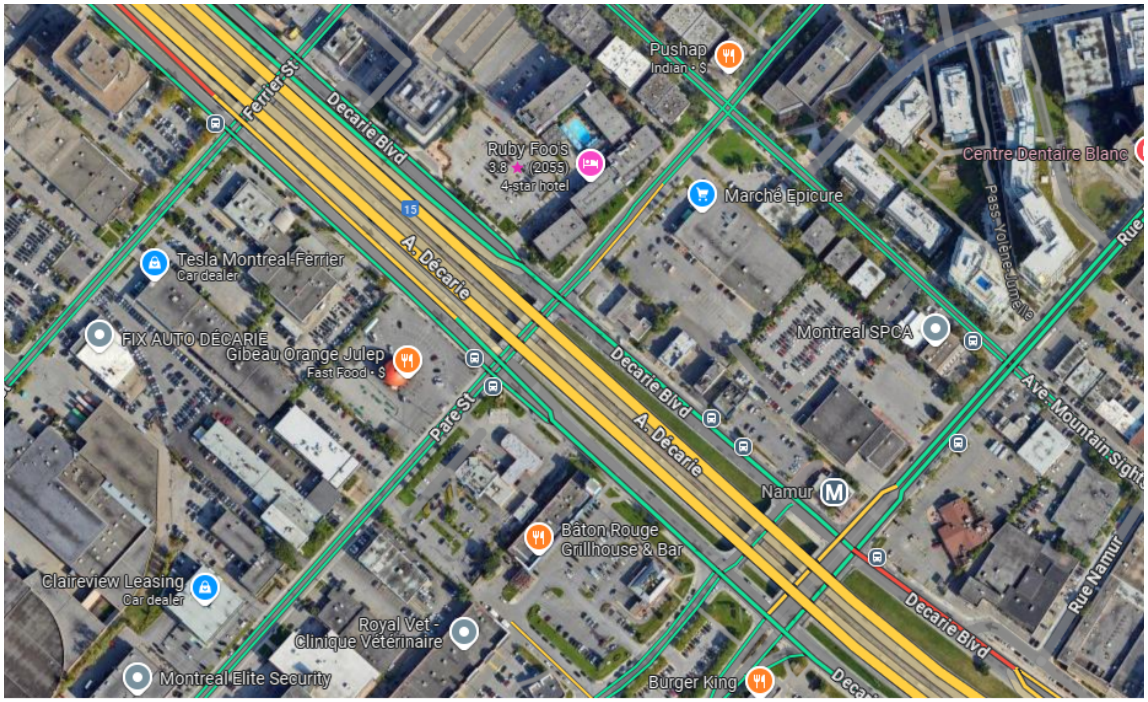


Figure 3.5: Simulation of a Sub-Area in Montreal

controlled environment before scaling it to larger zones.

We thoroughly analyzed the efficacy of our machine learning algorithms in the ensuing graphical representations. We used key measures including the Mean Square Error (MSE), Mean Absolute Error (MAE), and Coefficient of Determination (R2) to quantify their performance. We separated the dataset into two portions to thoroughly validate our models: an 80% training subset and a 20% testing subset. Using this method, we were able to effectively analyze the model's predictive skills throughout 300 iterations.

Figures 3.6 to 3.9 depict the congestion level forecasts provided by each separate machine learning method, along with the related MSE, MAE, and R2 values. The results clearly show that the Decision Tree algorithm performed better than the other machine learning approaches. This superiority was demonstrated by its faster convergence rate, lower loss values, and significantly greater coefficient of determination.

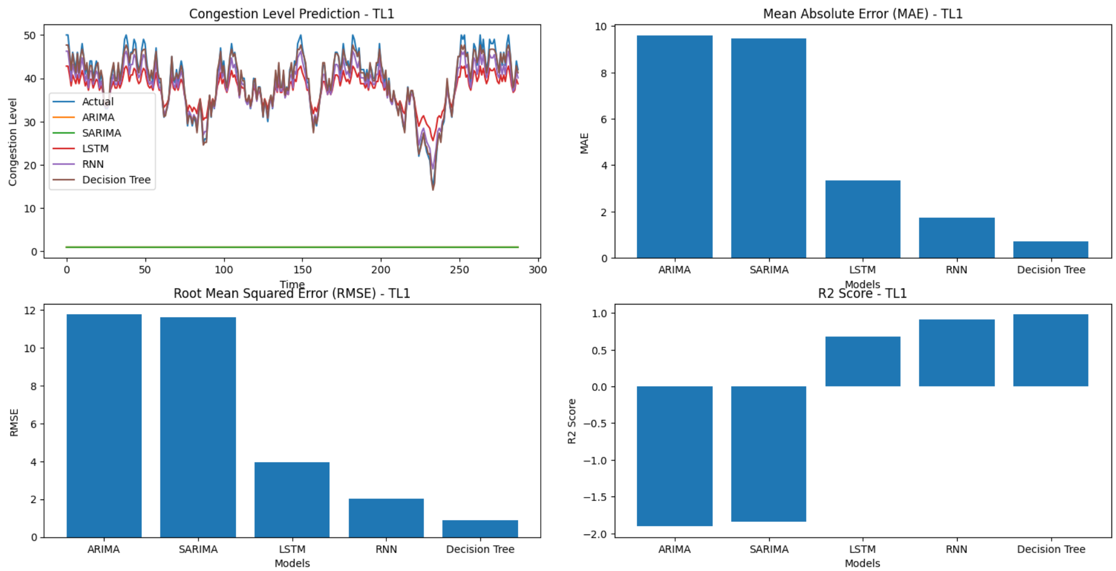


Figure 3.6: Traffic Light 1

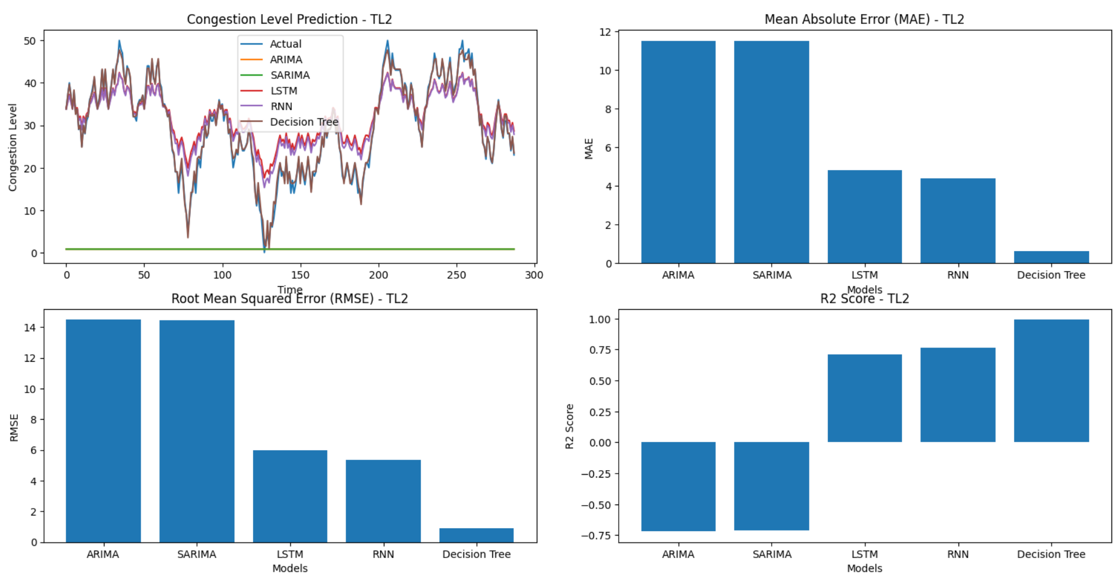


Figure 3.7: Traffic Light 2

3.6.2 Result for the optimization of traffic light timings using meta-heuristic algorithms(Phase 2)

One of our contributions is to develop an enhanced approach for BAT algorithm, in order to achieve this, we modified on the exploration phase of the algorithm. In BAT algorithm, there is an exploration phase, where the solution is generated randomly and then their fitness is calculated and

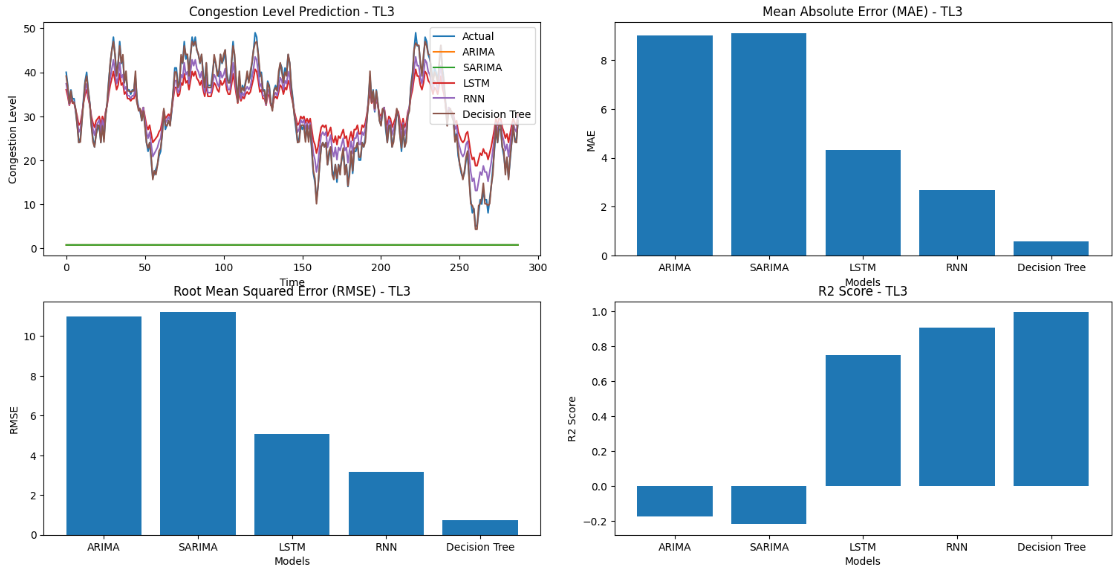


Figure 3.8: Traffic Light 3

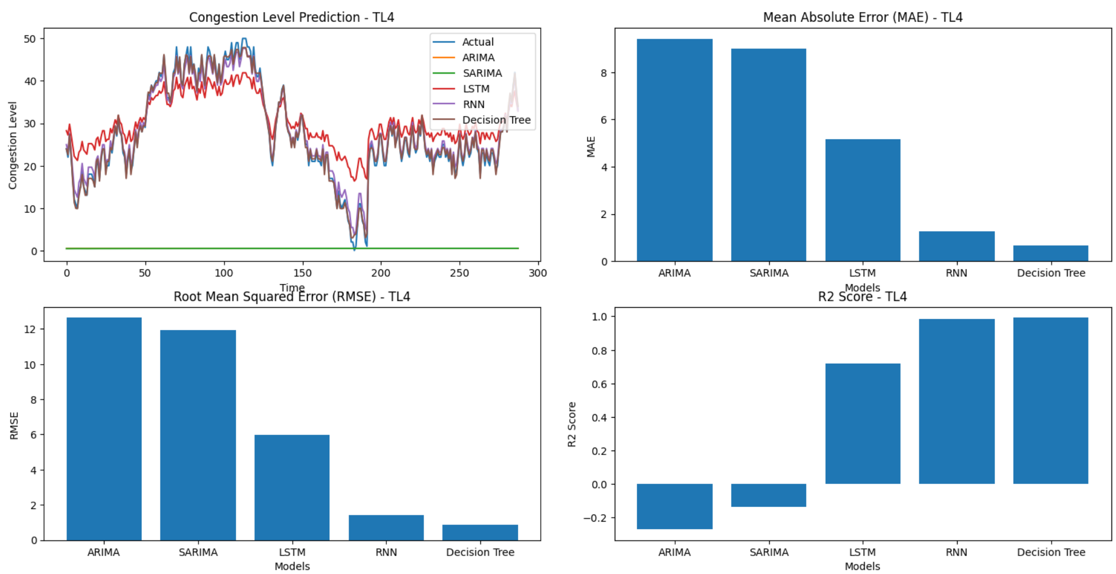


Figure 3.9: Traffic Light 4

compared with the best solution obtained so far and update it accordingly. In our modified version of the BAT algorithm, the exploration phase is not fully random but it is semi-random exploration phase, which means that the algorithm is directed to obtain better solution by taking into consideration the congestion status, as shown in the equation below:

$$x_i \leftarrow x_i + C \times (x^* - x_i)$$

Other enhancement is to tune the parameters in the BAT algorithm to have a better results in terms of convergence rate or the quality of the solution. Below is the algorithm of our proposed enhanced BAT algorithm.

Algorithm 2 Enhanced Bat Algorithm

Data: Objective function $f(x)$, Population size N, Number of generations G, Frequency scaling factor α, Pulse emission rate γ, Lower bound L, Upper bound U

Result: Optimal solution x^*

```
Initialize population X with random solutions  Initialize pulse rates r_i and loudness A_i for each bat
 Initialize best solution x*  for t ← 1 to G do
   for i ← 1 to N do
      Generate a new solution x_i by adding random step to X_i  if rand() < r_i then
         Generate a new solution x_i by adding a random step and a fraction of the current best
           solution
         Enhanced exploration term: x_i ← x_i + C × (x* − x_i)
      end
      Evaluate the fitness of x_i  if rand() < γ and f(x_i) < f(x*) then
         Accept x_i as the new best solution x*
      end
      Update r_i and A_i using equations
   end
end
```

Figures 3.10 and 3.11, we compare different meta-heuristic algorithms in terms of convergence rate. As shown in the results, Figure 3.10, categorized under high congestion show that Enhanced BAT algorithms outperforms other algorithms in terms of solution quality and minimize the objective function and converges to the near optimal solution after 25 iterations, which is defined as the average waiting time for each vehicle in the road network. The second best algorithm is the BAT algorithm which converges to its local minimum solution after almost 30 iterations. In this specific scenario, where the exploration phase is semi-random, which means that the exploration phase benefits from the predicted congestion status at each traffic light before adjusting the green timing for each traffic light in the zone, the initial solution is around 9 for both BAT and EBAT. Figure 3.11, categorized under low congestion show that Enhanced BAT algorithms outperforms other algorithms in terms of solution quality and minimize the objective function and converges to the near optimal solution after 25 iterations, which is defined as the average waiting time for each vehicle in the road network. The second best algorithm is the BAT algorithm which converges to its local minimum solution after almost 28 iterations. In this specific scenario, where the exploration

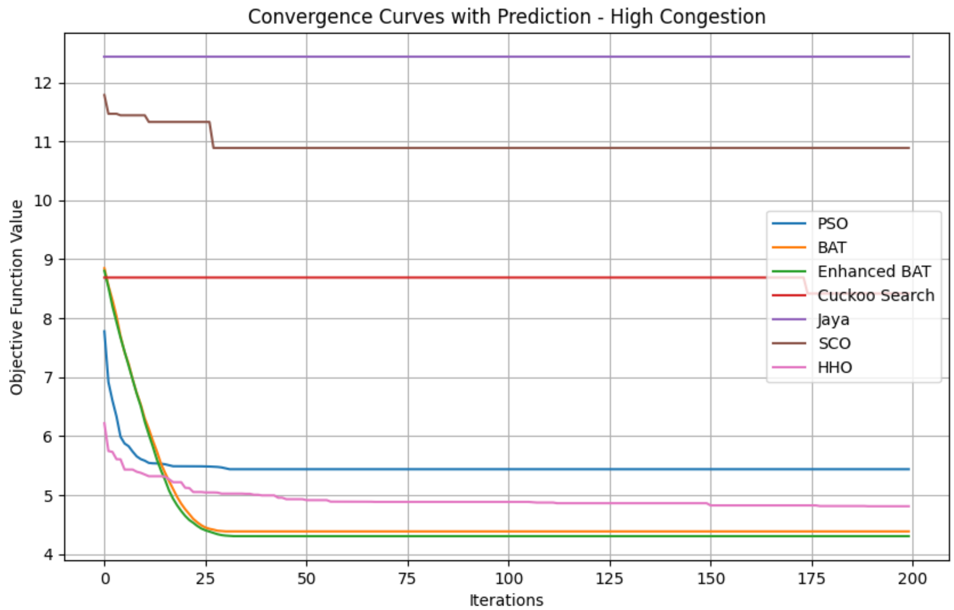


Figure 3.10: The Convergence Curves with Prediction - High Congestion

phase is semi-random, which means that the exploration phase benefits from the predicted congestion status at each traffic light before adjusting the green timing for each traffic light in the zone, the initial solution is around 6.5 for both BAT and EBAT.

.

Figures 3.12 and 3.13 show the results of the other scenario, which is also a dynamic scenario, where the timing for the traffic lights are changing dynamically based on the current congestion status and not the predicted congestion status as in the previous scenario. We compared different meta-heuristic algorithms in terms of convergence rate. As shown in the results, Figure 3.12, categorized under high congestion show that Enhanced BAT algorithm outperforms other algorithms in terms of solution quality and minimize the objective function and converges to the near optimal solution after 30 iterations. The second best algorithm is the BAT algorithm which converges to its local minimum solution after almost 37 iterations. In this specific scenario, nothing has changed on the exploration phase, the initial solution is around 15.5 for both BAT and EBAT. We noticed that the solution quality for the EBAT and BAT algorithms are almost the same, this is because the enhanced BAT algorithm in this case has only one improvement which is the parameter tuning but

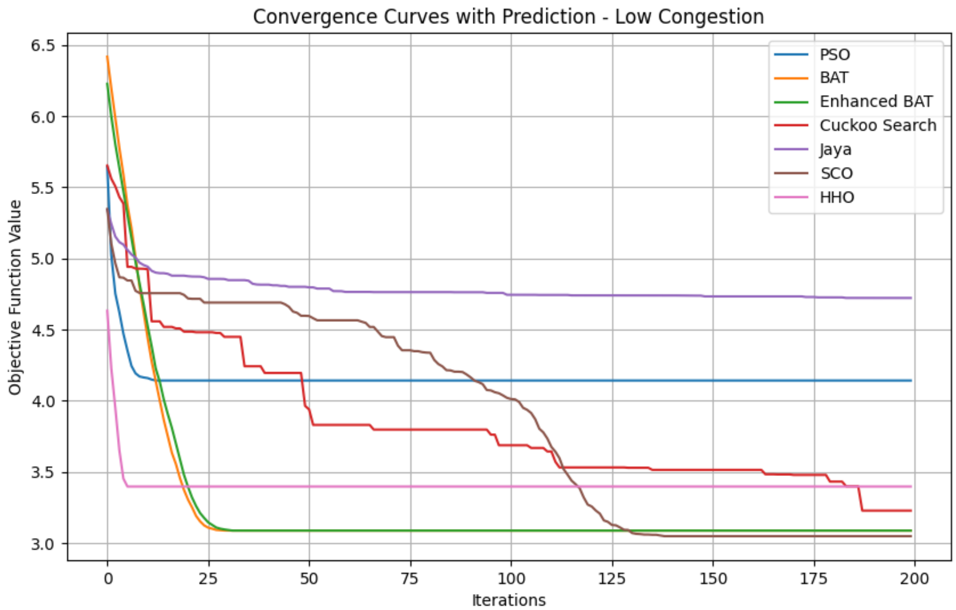


Figure 3.11: The Convergence Curves with Prediction - Low Congestion

there is no improvement on the exploration phase as in the previous scenario. Figure 3.13, categorized under low congestion show that Enhanced BAT algorithm outperforms other algorithms in terms of solution quality and minimize the objective function and converges to the near optimal solution after 30 iterations. The second best algorithm is the BAT algorithm which converges to its local minimum solution after almost 30 iterations. In this specific scenario, nothing has changed on the exploration phase, the initial solution is around 13.8 for both BAT and EBAT. We noticed that the solution quality for the EBAT and BAT algorithms are almost the same, this is because the enhanced BAT algorithm in this case has only one improvement which is the parameter tuning but there is no improvement on the exploration phase as in the previous scenario.

Figures 3.14 and 3.15 show the results of the third scenario, which is the fixed approach scenario, where the timing for the traffic lights are fixed at all times. We compared different meta-heuristic algorithms in terms of convergence rate. As shown in the results,Figure 3.14, categorized under high congestion shows that Enhanced BAT algorithms outperforms other algorithms in terms of solution quality and minimize the objective function and converges to the near optimal solution after 30 iterations. The second best algorithm is the BAT algorithm which converges to its local minimum

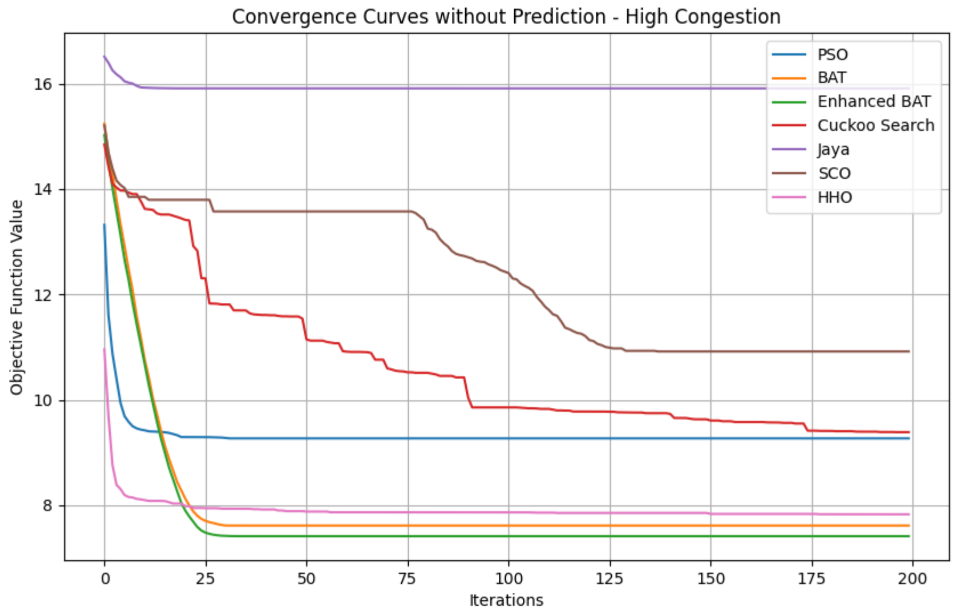


Figure 3.12: The Convergence Curves without Prediction-High Congestion

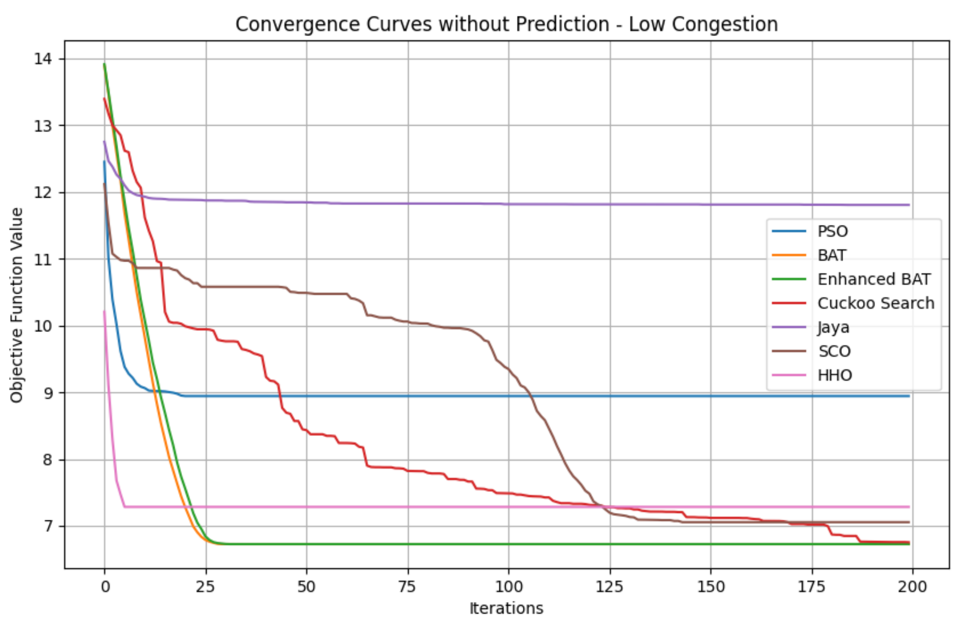


Figure 3.13: The Convergence Curves without Prediction-Low Congestion

solution after almost 37 iterations. In this specific scenario, nothing has changed on the exploration phase, the initial solution is around 19 to 19.5 for both BAT and EBAT. We noticed that the solution

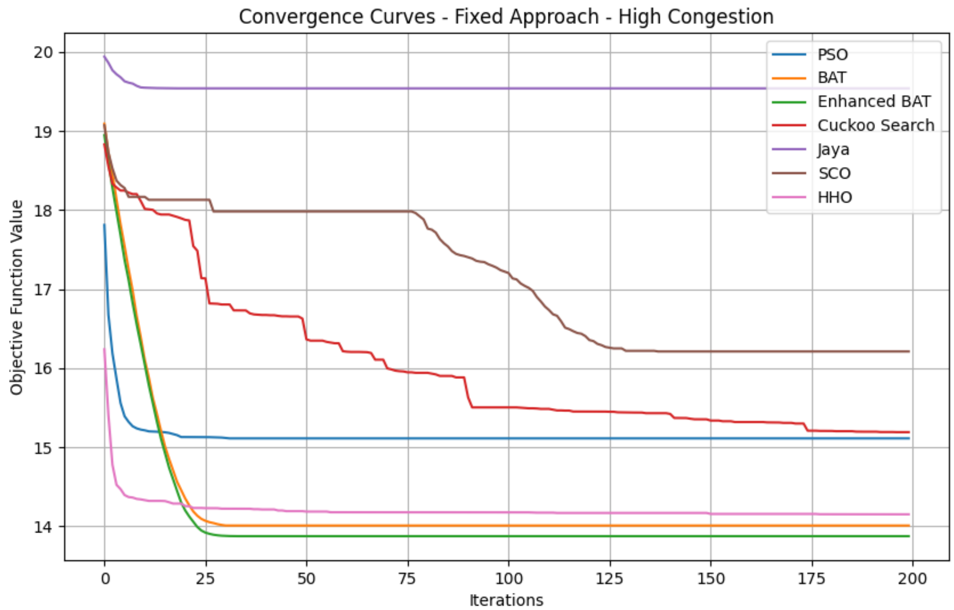


Figure 3.14: The Convergence Curves - Fixed Time Traffic Light - High Congestion

quality for the EBAT and BAT algorithms are almost the same, this is because the enhanced BAT algorithm in this case as well has only one improvement which is the parameter tuning but there is no improvement on the exploration phase as in the first scenario. Figure 3.15, categorized under low congestion shows that Enhanced BAT algorithms outperforms other algorithms in terms of solution quality and minimize the objective function and converges to the near optimal solution after 30 iterations. The second best algorithm is the BAT algorithm which converges to its local minimum solution after almost 35 iterations. In this specific scenario, nothing has changed on the exploration phase, the initial solution is around 6 to 6.5 for both BAT and EBAT. We noticed that the solution quality for the EBAT and BAT algorithms are almost the same, this is because the enhanced BAT algorithm in this case as well has only one improvement which is the parameter tuning but there is no improvement on the exploration phase as in the first scenario.

As you can see from the convergence rate in all the investigated scenarios, the initial solution is the key for a better solution and fast convergence as shown from Figures 3.16 to 3.21.

For example, in the fixed approach scenario because of the bad initial solution due to the fixed green timing regardless of the congestion at each traffic light, the EBAT algorithm converges after

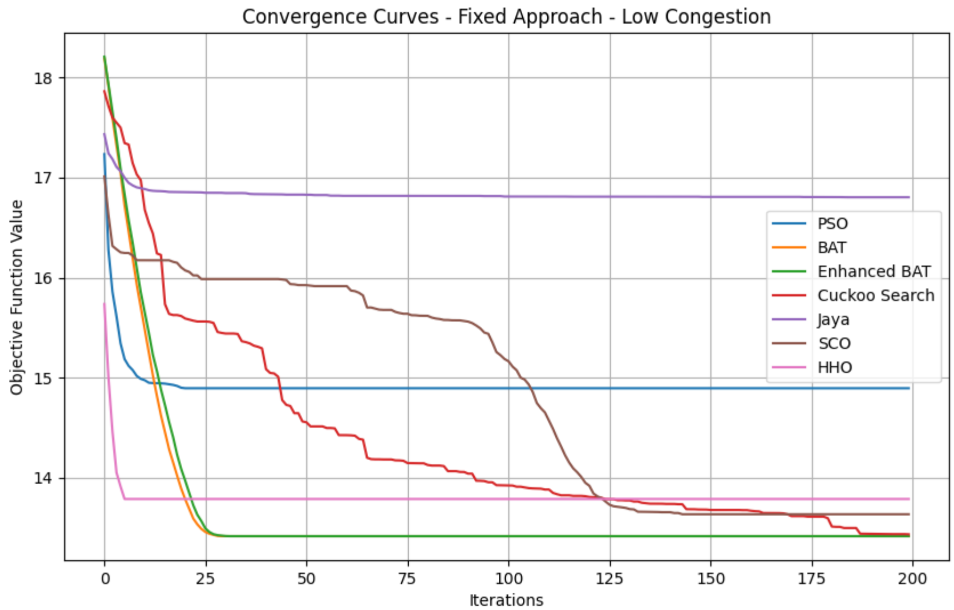


Figure 3.15: The Convergence Curves - Fixed Time Traffic Light - Low Congestion

44 iterations and 29 iterations for BAT algorithm and with the initial solutions to be 19 for both of them.

On the other side, the dynamic with no prediction approach converges after 42 iterations for the EBAT and 27 iterations for the BAT algorithm and with the initial solutions to be 15 for both EBAT and BAT algorithms.

In the third scenario, which is the dynamic with prediction approach, it converges after 37 iterations for the EBAT algorithm and 24 iterations for the BAT algorithm and with the initial solutions to be 9 for both EBAT and BAT algorithms.

In terms of execution time, there is slight difference between different algorithms with a reasonable execution time except for the JAYA, SCO and HHO algorithms as shown on the Figures above. Where the execution time for the third approach which is the dynamic with prediction approach is the highest and the execution time for the fixed approach is the lowest.

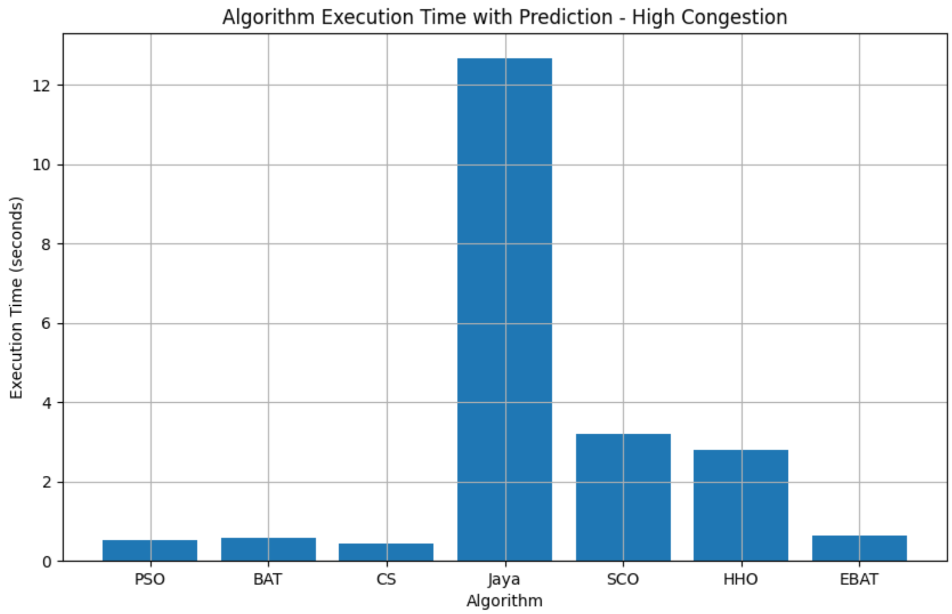


Figure 3.16: Execution Time with Prediction - High Congestion

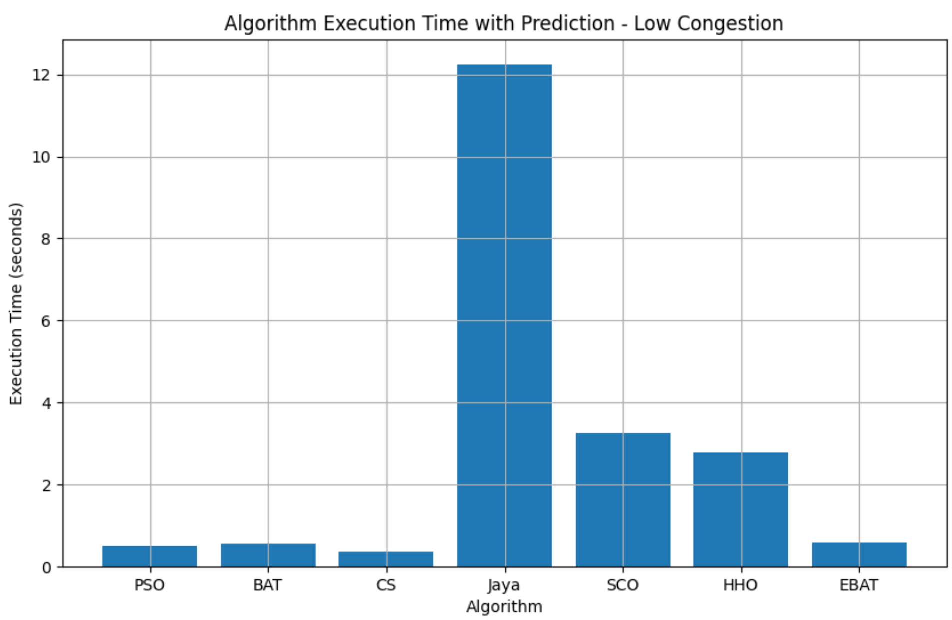


Figure 3.17: Execution Time with Prediction - Low Congestion

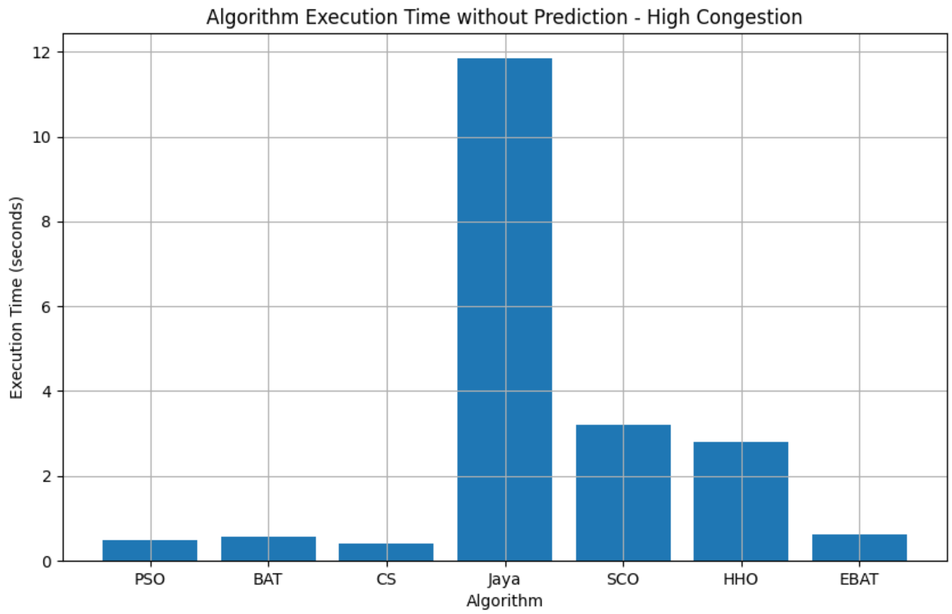


Figure 3.18: Execution Time without Prediction - High Congestion

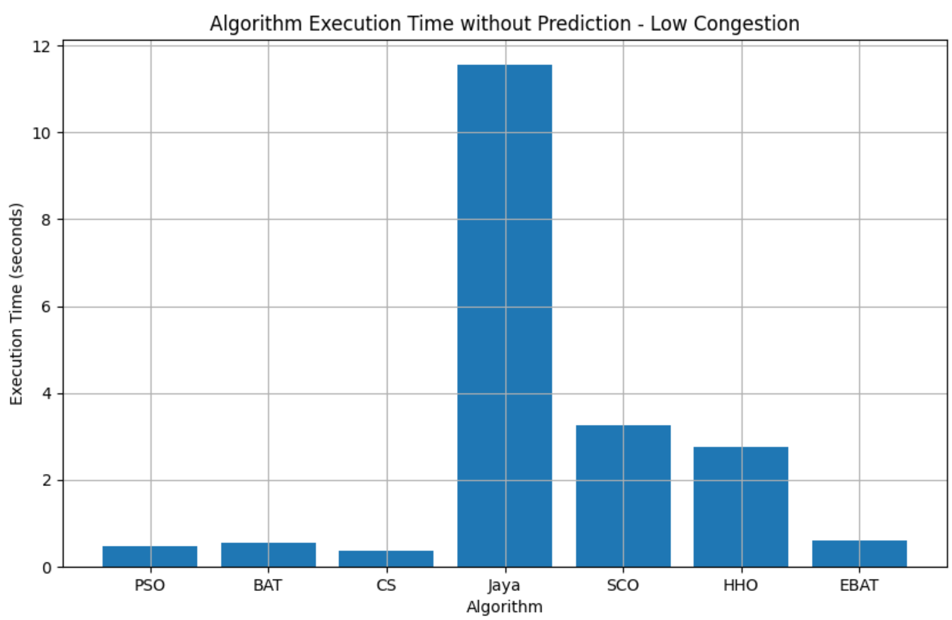


Figure 3.19: Execution Time without Prediction - Low Congestion

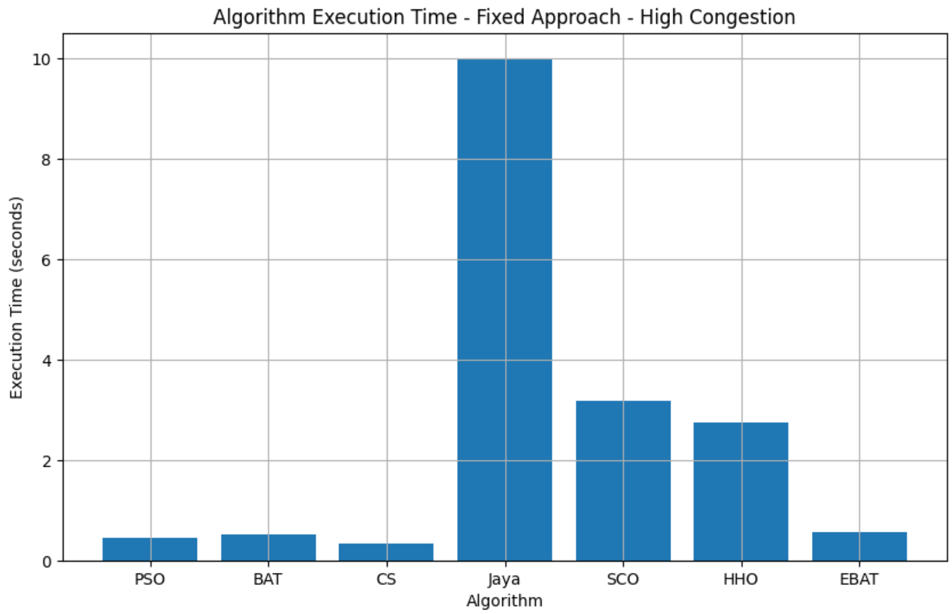


Figure 3.20: Execution Time - Fixed Time Traffic Light - High Congestion

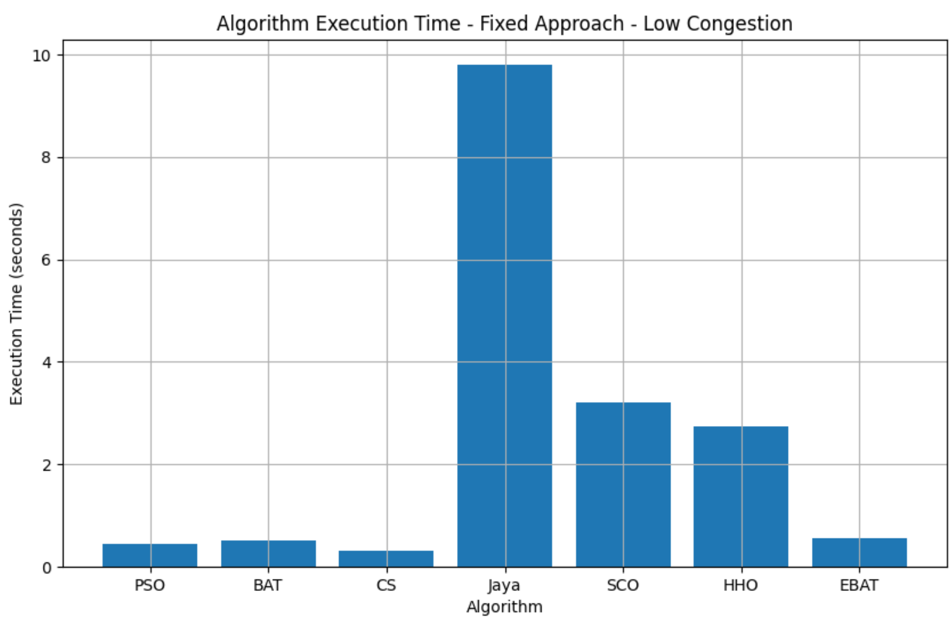


Figure 3.21: Execution Time - Fixed Time Traffic Light - Low Congestion

3.7 Summary

In our study the road Congestion Prediction is a process that involves two phases: Phase 1 and Phase 2. Phase 1 involves gathering comprehensive data on traffic-related variables, such as traffic volumes, vehicle speeds, and road occupancy, which have been taken from the data set, and then we used sophisticated prediction methodologies like Recurrent Neural Networks, Long Short-Term Memory networks, Decision Trees, AutoRegressive Integrated Moving Average, and Seasonal ARIMA. The effectiveness of these algorithms is assessed using metrics like Mean Squared Error (MSE) and Mean Absolute Error (MAE), ensuring precision in prediction.

Phase 2 presents a proactive approach to traffic optimization using metaheuristic algorithms, such as the EBAT, BAT Algorithm, Particle Swarm Optimization, Cuckoo Search, and Jaya Algorithm. The vehicle waiting times have been reduced, mitigating congestion and promoting more efficient traffic movement. The integration of these insights with the optimization efforts in Phase 2 results in an integrated framework that not only predicts congestion but also helps in development of proactive strategies to mitigate it through traffic signal timing optimization.

The successful completion of both phases depends on the precision of data, the efficacy of prediction algorithms, and the efficiency of the optimization process. The proposed technique has significant implications for contemporary contexts, and we aimed to address congestion issues using advanced machine learning techniques and conducting a comprehensive case study. The optimization targets involve minimizing total and average waiting times for all vehicles, including emergency vehicles. The optimization model incorporates time budgets, phase cycle lengths, smoothness of phase transitions, and maximum green times to ensure feasibility and safety.

There are other areas we could pursue for future work. These directions not only enhance our current research but also tackle emerging difficulties and opportunities in traffic management and urban development. As an illustration, we can focus on investigating the incorporation of real-time traffic data from IoT devices, social media, and other sources to improve the accuracy and responsiveness of our algorithm. A further avenue of research is expanding our investigation to encompass optimization for multi-modal traffic networks, encompassing pedestrians, motorcycles, and public transportation, so guaranteeing a more comprehensive approach to urban mobility. An additional improvement could involve prioritizing the enhancement of the resilience and security of the traffic management system, protecting it against cyber threats and guaranteeing its reliability.

In conclusion, this study provides a thorough investigation that forms the foundation of a complete approach to managing congestion and optimizing traffic, which has the potential to significantly transform transportation systems and rethink traffic control principles.

Chapter 4

Multilevel Learning for Traffic Congestion Prediction using Anomaly Detection and Ensemble Learning

4.1 Introduction

The persistence of traffic congestion poses a significant obstacle in contemporary transportation systems, resulting in issues such as prolonged commuting, heightened fuel use, and air pollution. Innovative prediction models based on multi-level learning approaches are required to tackle these challenges for highly precise traffic congestion forecasting strategies. The precise models in place facilitate the implementation of proactive traffic management measures to dynamically control traffic signals [98]. This method of traffic regulation involves altering the duration of signal phases at intersections to optimize traffic flow, informing signal timing based on pre-trained models, resulting in a more efficient and safe traffic system [97].

It is imperative to incorporate sophisticated Machine Learning (ML) algorithms into multi-level learning approaches, combining various tasks such as anomaly detection, data segmentation, and ensemble strategies. This integration aims to enhance prediction accuracy and cluster quality, yielding more robust models [99]. The multi-level learning approaches are founded on the premise that the combined strength of multiple methods in data analysis is better than using a single process to obtain the final predictions [100]. These approaches are essential in domains where accurate predictions are critical, such as proactive traffic management.

In this paper, Multilevel learning denotes a systematic method that encompasses several separate phases of learning, with each phase contributing a role in the final prediction model. More precisely, it encompasses:

- **Anomaly detection**: involves the identification and removing of unusual traffic patterns in order to clean the dataset and ensure robust baseline model training
- **Baseline Model Training**: Utilizing the cleaned dataset to train multiple machine learning models in order to establish initial predictive capabilities.
- **Ensemble Learning**: refers to the process of improving the overall accuracy of predictions by combining the predictions from multiple baseline models. This is achieved using techniques like stacking and voting.

This structured, multistage methodology enables gradual enhancements at each stage, leading in a final model that leverages the strengths of various machine learning techniques.

This research presents an innovative multi-level learning strategy specifically tailored to predict traffic congestion in highly populated urban regions. We employed various techniques, including time series windowing[101], anomaly detection, and multi-level ensemble-based regression, examining various settings. The windowing technique is a data preprocessing stage that involves transforming time series data of traffic flow into sequence patterns using fixed intervals. This technique facilitates the analysis of data collected from city intersections, creating more structured and systematic sequence patterns. These patterns can be used to incorporate trends and frequency routines in the data based on given window settings in the learning process, gaining valuable insights into the behavior of vehicles at city intersections, which can lead to more accurate predictions.

However, removing outliers from these collected patterns is essential for stable and robust prediction models. The unsupervised anomaly detection process involves determining abnormal traffic patterns based on a small contamination threshold as a first learning process[102]. The determined patterns are removed from the data before the second learning process, guaranteeing that the learning models are concentrated on acquiring knowledge from the dominated traffic flow patterns. The second learning process involves training a set of regression models, and the most performing ones are subject to ensemble learning as a third learning process, producing the final predictions resulting in greater forecast accuracy. The following pivotal points encapsulate the present study's primary contributions:

- A multi-level learning model is proposed for traffic congestion prediction that incorporates time series windowing, anomaly detection, and multi-level ensemble-based regression. This approach aims to enhance the precision of the prediction models.
- The study examines various windowing settings alongside dimensionality reduction based on Principal Component Analysis (PCA) in the preprocessing stage and three unsupervised anomaly detection methods in the first learning stage. The benefit of employing PCA resides in its function during the preprocessing phase to achieve dimensionality reduction. PCA is utilized to reduce the amount of features by retaining only the most significant ones, so simplifying the dataset while maintaining the necessary volatility in the data. This is especially advantageous for traffic congestion data, as it reduces computational complexity and improves the efficacy of machine learning models by removing irrelevant or redundant information. In our study, PCA enhances anomaly detection efficiency and enhances prediction accuracy by ensuring that the data input into the model is both concise and indicative of the underlying patterns. The accuracy of the prediction of the second learning stage is compared to determine the impact of these settings and methods using a real dataset[103].
- The study also examines baseline linear regression models in the second learning stage and two advanced learning methods (voting and stacking) in the final learning stage. These methods are evaluated to determine their effectiveness in enhancing the accuracy of the prediction models.

This research presents a comprehensive multi-level learning approach integrating different techniques to predict traffic congestion with high precision. The proposed approach enables proactive traffic management by controlling traffic signal delay time dynamically. This method aims to alter the duration of signal phases at intersections based on the study of traffic flow patterns and behaviors to optimize traffic flow in big cities, resulting in a more efficient and safe traffic system. The results indicate that the proposed approach outperforms traditional linear regression models, making it a promising solution for traffic congestion prediction.

4.2 Methodology

This section introduces a machine learning framework designed for forecasting traffic patterns and identifying anomalies within ITS. The framework utilizes a combination of data preprocessing approaches, anomaly detection methods, to attain resilient and reliable outcomes.

Figure 5.1 describes the multi-stage process proposed in this study to forecast traffic congestion. The procedure comprises several distinct stages:

1. **Data Acquisition and Preprocessing:** Historical traffic data is gathered from several sources like loop detectors, cameras, sensors, and GPS. Preprocessing approaches are utilized to handle probable missing values, discrepancies, and anomalies in the data, guaranteeing its appropriateness for machine learning algorithms.

2. **Windowing and Anomaly Detection:** Traffic data is partitioned into smaller time windows using windowing techniques. This enables the examination of short-term traffic patterns and the detection of anomalous events within certain time periods. Afterwards, anomaly detection techniques are used to identify any anomalies or outliers that may be present in the dataset.

3. **Multi-Level Learning:** the next step is Training Multiple Models, where a set of baseline learner models are trained. The models that exhibit the best performance on a validation set are chosen for further processing. In the next level, Ensemble Learning is employed to combine the most effective models from the previous step. Ensemble learning techniques, such as stacking or voting, are used to create a more robust model.

4. **Final Prediction:** In the Final Prediction phase, the ensemble model is used to generate forecasts with the expectation that these forecasts will be more accurate than predictions made by individual models.

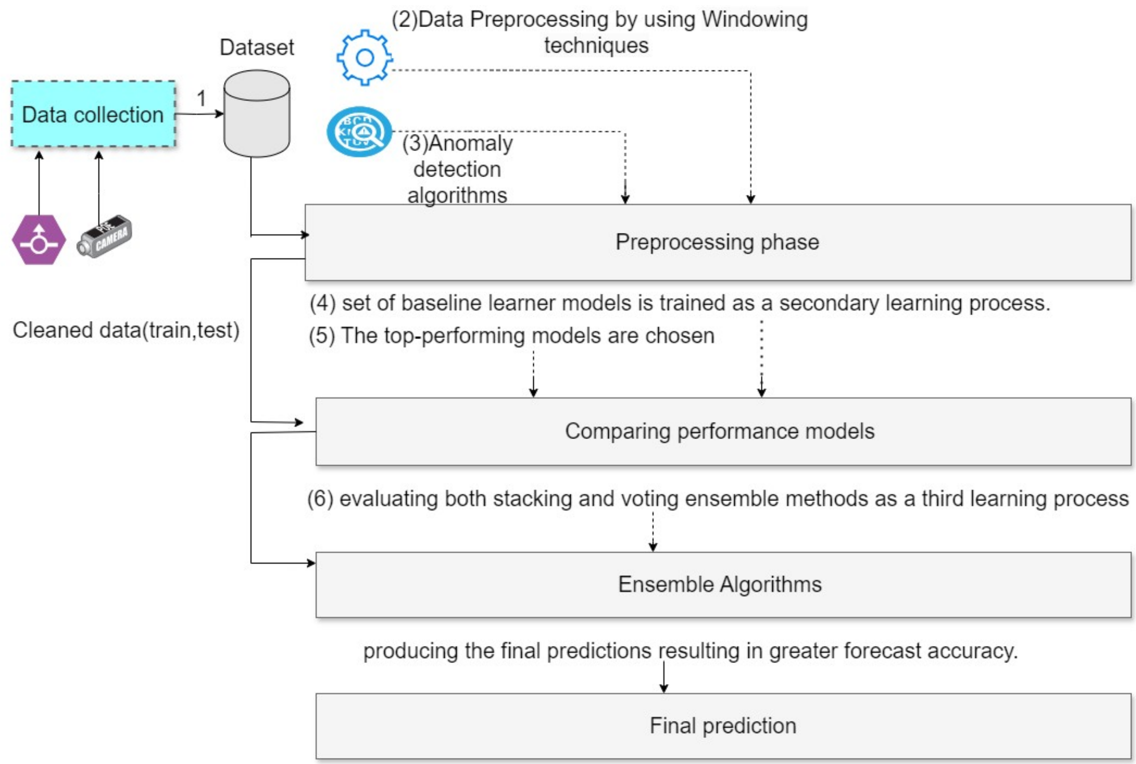


Figure 4.1: Integrated framework for traffic flow prediction: anomaly detection and ensemble learning.

Essentially, this machine learning system converts unprocessed data into a formatted and enhanced structure. Subsequently, it conducts training on several models and amalgamates them to generate a conclusive prediction model that exhibits improved accuracy.

In Algorithm 3 introduces the preparation of accurate data and the selection of appropriate baseline models for predicting traffic congestion. This will be achieved through the use of a multi-level learning method.

According to the information provided Algorithm 3 are further elaborated upon in subsequent sections:

Data Acquisition and Preprocessing Initially, we gather traffic congestion data (V) from every intersection inside the designated study area. The data is segmented into temporally consistent chunks using a sliding window approach, with a window size (W) and step size (S). In order to enhance the data, we integrate supplementary attributes such as longitude, latitude, and spatial coordinates (X, Y). A data anomaly detection technique (A) is utilized to find and remove outliers from the dataset, using a predetermined threshold (C). This guarantees that the model acquires knowledge from trustworthy data. Data preprocessing techniques, such as data transformation (T) and normalization (N), are utilized to prepare the data for machine learning models. These strategies enhance the performance of the model by rescaling the data and converting it into a suitable format for training the model.

Multi-Level Learning with Baseline Selection and Ensemble Modeling We create a baseline Multi-level models by training and testing a varied collection of machine learning models (B). The performance of each linear model is assessed using a predetermined metric (M), such as mean squared error (MSE) or R-squared. These metrics measure the degree to which the model's predictions correspond to real traffic circumstances. From the initial set (B) is used to choose models from a chosen metric (M) that surpass a given performance threshold (t). The top-performing models (B') selected.

An ensemble method (E), such as voting or stacking, is then applied to the selected models in B'. In the voting process, each model makes a prediction about the class (congested or uncongested) for a given data point. and the final forecast is determined by the majority vote among the models. Stacking is the process of training a meta-model using the outputs of individual models in B'. The selected ensemble approach (E) is subsequently employed to train the ultimate prediction model (P) for forecasting traffic congestion.

The implementation of this hierarchical ensemble technique, which incorporates anomaly detection, presents a new and efficient strategy for forecasting traffic congestion. It harnesses the combined power of several models, removes anomalies, and use data pretreatment techniques to improve the accuracy of predictions.

4.3 Experimental Setup and Evaluation

The proposed methodology, which is based on a wide range of multi-output regression models, undergoes thorough evaluation using a real dataset. This evaluation uses a number of performance metrics to carefully examine the accuracy of predictions within a specific goal timeframe. The next sections offer detailed information on the selection and use of these performance measurements,

Algorithm 3 Multilevel Learning Approach with Anomaly Detection Model for Traffic Congestion Prediction.

Input:
V : Set of traffic flow data,
W : Window size ,
S : Step size,
A : Anomaly detection algorithm ,
C : Contamination threshold ,
T : Transformation method ,
N : Normalization method ,
B : Set of base learner models ,
M : Performance metric ,
t : Threshold for selecting top models ,
E : Ensemble method ,
Output:
P : Final Congestion prediction model
1. Data Collection and Preprocessing:
1.1 Collect Congestion data from all intersections V H.
1.2 Apply sliding window technique with window size W and step size S.
1.4 Add additional information (longitude , latitude,X,Y for location information).
1.5 Apply anomaly detection algorithm A with threshold C to remove outliers.
1.6 Apply transformation T and normalization N to the data.
2. Baseline Multi-level Learning:
2.1 Train and validate a set of baseline linear models B.
2.2 Evaluate each model's performance using metric M.
2.3 Select top-performing models B' exceeding threshold t.
3. Final Ensemble Model:
3.1 Apply ensemble method E (Voting or Stacking) on models B'.
3.2 Train the final model P using the selected ensemble method.
Return: Final Congestion prediction model P.

outline the experimental setup, and provide a complete discussion and analysis of the experimental outcomes.

4.3.1 Performance Metrics

In this study, several performance metrics—such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2)—were used to evaluate the prediction accuracy of the machine learning models. These metrics provide insights into error magnitudes, model bias, and overall prediction quality. For a detailed explanation of these metrics see Section 3.5.1.

4.3.2 Experimental Setup

This section establishes the framework for an empirical evaluation of our proposed methodology. We will carefully design an experimental arrangement that includes three essential elements: dataset description, windowing setups, and model architectures.

First, we examine the attributes of the dataset used to train and test the model. A thorough explanation will be given, encompassing information about the data source, types of features, and any necessary pretreatment processes taken to ensure the data is appropriate for modeling.

Additionally, we examine the windowing setups used for segmenting data where the selection of window size and type, such as sliding window or fixed window, is of utmost importance in capturing temporal dependencies and impacting the performance of the model. We explore the reasoning behind the chosen windowing solutions and the possible alternatives that were considered for further examination.

Ultimately, we shall thoroughly outline the specific model structures utilized in our research. This involves describing the precise type of model, such as linear regression or ensemble approach, together with its hyperparameter settings and any customized configurations designed for our unique goal. By thoroughly delineating these three elements, we build a strong experimental framework for assessing the efficacy of our suggested approach.

4.3.2.1 Dataset

Our study utilizes a dataset containing information about intersections and their corresponding attributes (as shown in Table 5.1). Every intersection in the dataset is distinguished by a unique identifier called ”INT ID” (Intersection Number), ensuring unambiguous differentiation. The ”Intersection Name” field contains descriptive labels for each intersection, typically including the names of the two intersecting streets. The ”ARRONDISSEMENT” field indicates the region in which each intersection is situated, offering more context for classification purposes [104].

Moreover,the dataset incorporates geographical coordinates for every intersection. The ”Longitude” and ”Latitude” fields precisely indicate the east-west and north-south coordinates, respectively. These coordinates are essential for geographic analysis and visualization applications and congestion level field contains information regarding the number of vehicles observed at the intersection within a specified time frame. The data is gathered through regular observations of the vehicle count at the intersection, with measurements taken every five minutes.

In order to assess the ability of our model generalizing to unseen data, we utilized a comprehensive method that involved both cross-validation and a hold-out set. We adopt a 5-fold approach for cross-validation. The process involves splitting the dataset into five folds of equal size, each serving as a sub-dataset. The model underwent training using a four-fold cross-validation technique and assess thereafter on the remaining fold in a rotating manner. The method iterates five times, ensuring that each fold utilizes for validation once. Ultimately, a hold-out set including a distinct subset of the data is employed to conduct a final assessment of the model's ability to generalize.

4.3.2.2 Region-Based Analysis

As illustrated in Figure 3.4, our investigation focuses in particular on a chosen simulated region located inside the city of Montreal. We partition the simulation area into distinct zones and analyze the problem for each zone independently. Regardless of the network's size, even if it is extensive, it will be subdivided into zones, and the prediction algorithms will be applied separately to each zone. In our simulation, each zone contained 19 traffic lights, demonstrating the scalability of our proposed algorithm.

4.3.2.3 Windowing setups

This study thoroughly examines several windowing setups in combination with different anomaly detection algorithms. We provide three separate scenarios to carefully examine the effect of varying window sizes and step sizes on the predicted accuracy.

In the first scenario, we utilize a window size of 1 hour and a step size of 30 minutes. The purpose of this arrangement is to accurately capture and analyze the detailed temporal changes within the dataset, enabling a thorough examination of traffic patterns.

In the second scenario, we increase the window size to 2 hours and simultaneously increase the step size to 1 hour. This modification allows for a wider range of observation, which could potentially aid in the detection of longer-term patterns or variations in traffic behavior.

Table 4.1: Dataset Description

Field Name	Description
INT ID	A unique identifier assigned to each intersection within the dataset collected from the city of Montreal. This identifier ensures that each intersection can be distinctly recognized and referenced for analysis purposes.
Intersection Name	A descriptive label that typically includes the names of the streets that intersect at this point within Montreal. This helps in identifying the specific location within the city, facilitating easier reference and analysis.
ARRONDISSEMENT (Region Name)	The administrative region or district within Montreal where the intersection is situated. This information is crucial for regional traffic analysis and management, allowing for targeted interventions and resource allocation.
Longitude	The east-west geographic coordinate of the intersection, represented in decimal degrees. It is used in conjunction with latitude to pinpoint the exact location on a map within Montreal.
Latitude	The north-south geographic coordinate of the intersection, represented in decimal degrees. This, combined with longitude, determines the precise geographic location within the city.
Number of Vehicles	The count of vehicles observed at the intersection within a five-minute interval. This measure is repeatedly recorded to provide time-series data on traffic flow and congestion levels across Montreal.

Finally, **in the third scenario**, we increase the window size to 3 hours and the step size to 1 hour and 30 minutes. This setup seeks to achieve a compromise between collecting extensive traffic patterns and retaining computing performance by increasing the time intervals for both the window and step sizes.

The purpose of this organized investigation into windowing setups is to offer valuable information on the most efficient arrangement for effectively using anomaly detection techniques in the examination of traffic data.

4.3.2.4 Model setups

For each of the scenarios described in the Windowing setups section, we perform a thorough cleansing of data procedure as the first step in setting up the model. This entailed the application of sophisticated anomaly detection methods, specifically local outlier factor, elliptic envelope, and isolation forest. The techniques use to accurately detect and reduce outliers in the dataset, hence improving the reliability of subsequent studies.

Afterwards, we explore the construction of basic regression models by examining fundamental linear regression models. The selection of these 15 baseline models, was based on their simplicity and interpretability. They serve as a fundamental element in our analytical framework.

In addition, our research involved using an ensemble learning strategy to improve the accuracy of our predictions. In this framework, we examine the application of both the voting model with equal weights and stacking models. Ensemble learning techniques strive to improve predictive accuracy by combining the outputs of many base models. This approach allows for the utilization of the strengths of individual models while minimizing their limitations. In order to guarantee the strength and durability of our ensemble, we carefully chose the top 6 performing models from the baseline linear regression models to be included in the ensemble architecture.

All models used in this investigation are evaluated using time series cross-validatora with five folds, and are configured according to the default settings provided by scikit-learn [105], an open-source Python machine learning package. By implementing this standardized design, we guarantee uniformity and the capacity to make comparisons in our analyses. This, in turn, makes it easier to reproduce and rely on our findings.

4.3.3 Experimental Results and Analysis

This section focuses on the results and analysis of our advanced multilevel ensemble approach with anomaly detection, designed to improve the accuracy of traffic congestion prediction. We begin by explaining the prepossessing methods, which involve using sliding windowing with different configurations. Then, for each scenario, we show the results associated with various anomaly detection algorithms as a first stage in the learning process to identify abnormal events in traffic patterns.

The comparison results of our study demonstrate that the suggested multilevel learning technique considering baseline models in terms of prediction accuracy, emphasizing the significance of the multilevel learning strategy on prediction accuracy.

4.3.3.1 Scenario One (window size=1 hour and step size=30 minutes)

We explicitly define the duration of the window as 1 hour and the interval between each step as 30 minutes.

Table 4.2 compares the performance of baseline regression models. These models were trained on preprocessed data sets cleaned using various anomaly detection approaches, namely Elliptic Envelope, Isolation Forest, and Local Outlier Factor, respectively, Table 4.2 (a), Table 4.2 (b), and Table 4.2 (c) as part of the data cleaning process.

Table 4.2 shows that the Support Vector Regression (SVR) model typically demonstrates strong performance. Among the three anomaly detection approaches (Elliptic Envelope, Isolation Forest, Local Outlier Factor), SVR consistently exhibits the lowest MAE,MSE, and RMSE values. This suggests that it possesses the lowest overall margin of error compared to the models examined in this study. The SVR model has the longest duration for training. Nevertheless, SVR exhibits the most extended duration for training when compared to any other models. Consequently, although it may yield greater precision, the process of training is also more computationally demanding.

Table 4.3 presents a comparison of the performance of the top six linear regression models, evaluated using the cross-validation data. These models are selected from the baseline linear models, shown in Table 4.2.

Table 4.4 presents a comparison of the performance of two regression models: Stacking Regressor and Voting Regressor and shows that the Stacking Regressor demonstrates superior performance than the Voting Regressor on three out of the five metrics (MAE, MSE, and RMSE) for each all outlier approach (Elliptic Envelope, Isolation Forest, and Local Outlier Factor) and the difference is insignificant when considering the other two parameters, R^2 and MAPE.

Table 4.5 compares the performance of two regression models, Stacking Regressor and Voting Regressor, on the testing data set for Scenario one.

Table 4.5,shows that the Stacking Regressor demonstrates superior performance compared to the

Table 4.2: Comparison results of linear baseline models across various anomaly methods using cross-validation data for Scenario one.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE	TT (Sec)
Linear Regression	2.85403	11.87791	3.44571	0.92075	0.16917	18.176
Ridge Regression	2.85398	11.87763	3.44567	0.92075	0.16919	17.530
Least Angle Regression	2.85403	11.87791	3.44571	0.92075	0.16917	17.584
Bayesian Ridge	2.85403	11.87790	3.44570	0.92075	0.16917	17.790
Automatic Relevance Determination	2.85405	11.87810	3.44573	0.92075	0.16917	17.310
Huber Regressor	2.85351	11.89083	3.44752	0.92064	0.16645	18.518
TheilSen Regressor	2.85596	11.89681	3.44845	0.92062	0.16837	29.172
Support Vector Regression	2.83899	11.96072	3.45750	0.92019	0.15983	1148.626
Random Sample Consensus	2.93447	12.71166	3.56397	0.91530	0.17494	18.002
Passive Aggressive Regressor	3.31320	17.23769	4.11763	0.88396	0.17595	17.348
Orthogonal Matching Pursuit	4.84055	36.70200	6.05805	0.75556	0.30104	18.032
Lasso Regression	4.96265	37.89910	6.15586	0.74792	0.33949	18.000
Lasso Least Angle Regression	4.96265	37.89909	6.15586	0.74792	0.33949	17.570
Elastic Net	6.22099	57.97724	7.60912	0.61652	0.49655	17.612
Kernel Ridge	11.68482	346.24103	11.76732	-1.94001	0.58842	719.960

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE	TT (Sec)
Support Vector Regression	2.82083	11.66587	3.41490	0.92227	0.16967	1157.728
Linear Regression	2.85993	11.79312	3.43352	0.92141	0.17932	6.402
Ridge Regression	2.85988	11.79288	3.43348	0.92141	0.17935	5.552
Least Angle Regression	2.85993	11.79312	3.43352	0.92141	0.17932	5.630
Bayesian Ridge	2.85992	11.79311	3.43352	0.92141	0.17932	5.574
Automatic Relevance Determination	2.85990	11.79293	3.43349	0.92141	0.17932	5.560
Huber Regressor	2.85965	11.79826	3.43426	0.92138	0.17938	6.128
TheilSen Regressor	2.86269	11.81798	3.43712	0.92124	0.17889	18.616
Random Sample Consensus	2.94681	12.73916	3.56699	0.91520	0.18408	5.558
Passive Aggressive Regressor	3.27398	16.42730	4.01199	0.88767	0.17495	5.552
Orthogonal Matching Pursuit	4.85238	36.79045	6.06533	0.75501	0.31064	5.554
Lasso Regression	5.03461	38.94793	6.23988	0.74122	0.36271	5.634
Lasso Least Angle Regression	5.03461	38.94793	6.23988	0.74122	0.36271	5.562
Elastic Net	6.49682	63.96199	7.98748	0.57795	0.54432	5.606
Kernel Ridge	12.11152	371.54491	12.18943	-2.11169	0.60685	949.534

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE	TT (Sec)
Support Vector Regression	2.79619	11.52388	3.39383	0.92312	0.15403	1219.452
Linear Regression	2.85745	11.82155	3.43774	0.92121	0.17364	76.948
Ridge Regression	2.85741	11.82136	3.43772	0.92121	0.17367	74.848
Least Angle Regression	2.85745	11.82155	3.43774	0.92121	0.17364	76.288
Bayesian Ridge	2.85745	11.82154	3.43774	0.92121	0.17364	75.168
Automatic Relevance Determination	2.85746	11.82164	3.43776	0.92121	0.17365	76.320
TheilSen Regressor	2.85983	11.83371	3.43954	0.92115	0.17419	87.248
Huber Regressor	2.85692	11.83419	3.43956	0.92111	0.17160	75.142
Random Sample Consensus	2.92898	12.63470	3.55367	0.91558	0.16777	76.332
Passive Aggressive Regressor	3.58894	20.03340	4.40361	0.86530	0.22327	76.660
Orthogonal Matching Pursuit	4.85228	36.85105	6.07033	0.75459	0.30498	77.004
Lasso Regression	4.97293	38.08433	6.17072	0.74678	0.34606	76.372
Lasso Least Angle Regression	4.97293	38.08433	6.17072	0.74678	0.34606	76.254
Elastic Net	6.25354	58.95595	7.67187	0.61034	0.50908	76.306
Kernel Ridge	11.86210	356.71482	11.94278	-2.01305	0.59642	743.500

Table 4.3: Comparison results of selected linear baseline models across various anomaly methods using hold-out data for Scenario one.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE
Linear Regression	2.86515	11.95538	3.45766	0.92141	0.17261
Ridge Regression	2.86515	11.95536	3.45765	0.92141	0.17262
Least Angle Regression	2.86515	11.95538	3.45766	0.92141	0.17261
Bayesian Ridge	2.86515	11.95538	3.45765	0.92141	0.17261
Automatic Relevance Determination	2.86515	11.95533	3.45765	0.92141	0.17261
Huber Regressor	2.86436	11.96897	3.45962	0.92132	0.16970

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE
Support Vector Regression	2.80405	11.50042	3.39123	0.92440	0.16688
Linear Regression	2.86858	11.83845	3.44070	0.92218	0.18481
Ridge Regression	2.86857	11.83848	3.44071	0.92218	0.18482
Least Angle Regression	2.86858	11.83845	3.44070	0.92218	0.18481
Bayesian Ridge	2.86858	11.83845	3.44070	0.92218	0.18481
Automatic Relevance Determination	2.86853	11.83812	3.44066	0.92218	0.18480

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE
Support Vector Regression	2.77918	11.34930	3.36887	0.92539	0.15314
Linear Regression	2.85881	11.78880	3.43348	0.92251	0.17751
Ridge Regression	2.85880	11.78885	3.43349	0.92251	0.17752
Least Angle Regression	2.85881	11.78880	3.43348	0.92251	0.17751
Bayesian Ridge	2.85881	11.78880	3.43348	0.92251	0.17751
Automatic Relevance Determination	2.85881	11.78880	3.43348	0.92251	0.17751

Table 4.4: Performance results of Stacking Regressor versus Voting Regressor using cross-validation data for Scenario one.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE
Voting Regressor	2.85382	11.87872	3.44581	0.92074	0.16871
Stacking Regressor	2.85967	11.93050	3.45339	0.92042	0.17265

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.81562	11.55922	3.39927	0.92297	0.17110
Voting Regressor	2.84077	11.65015	3.41265	0.92236	0.17697

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.79562	11.46311	3.38491	0.92353	0.15765
Voting Regressor	2.83580	11.66908	3.41547	0.92221	0.16959

Table 4.5: Performance results of Stacking Regressor versus Voting Regressor on the testing data set for Scenario one.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE
Voting Regressor	2.86489	11.95625	3.45778	0.92140	0.17211
Stacking Regressor	2.86556	11.96055	3.45840	0.92138	0.17260

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.80609	11.46212	3.38558	0.92465	0.17110
Voting Regressor	2.84505	11.65776	3.41435	0.92337	0.18102

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.78152	11.30747	3.36266	0.92567	0.15847
Voting Regressor	2.83324	11.60496	3.40661	0.92371	0.17255

Voting Regressor. The Stacking Regressor demonstrates superior performance than the Voting Regressor in three out of the five metrics (MAE, MSE, and RMSE) for Isolation Forest and Local Outlier Factor approaches (Elliptic Envelope, Isolation Forest, and Local Outlier Factor).Interestingly, this trend does not hold true for the Elliptic Envelope method, where the Voting Regressor appears to have a slight advantage. This suggests that the choice of outlier detection technique may influence the optimal regression model selection.

Figure 4.2 presents a comparison of the performance of two distinct regression models, namely the Stacking Regressor and the Voting Regressor, in the context of a traffic flow prediction job. The performance is evaluated using the R^2 metric. Both the Stacking Regressor and Voting Regressor yield a significantly high R^2 value, approaching 0.92, when used to all three anomaly detection techniques (Elliptic Envelope, Isolation Forest, Local Outlier Factor). This indicates that both models well represent the general pattern of traffic flow, irrespective of the strategy employed to identify outliers. The Stacking Regressor consistently attains a marginally superior R^2 value in comparison to the Voting Regressor for every outlier identification approach. The discrepancy is negligible (about 0.001-0.003), but consistently present in all three approaches.

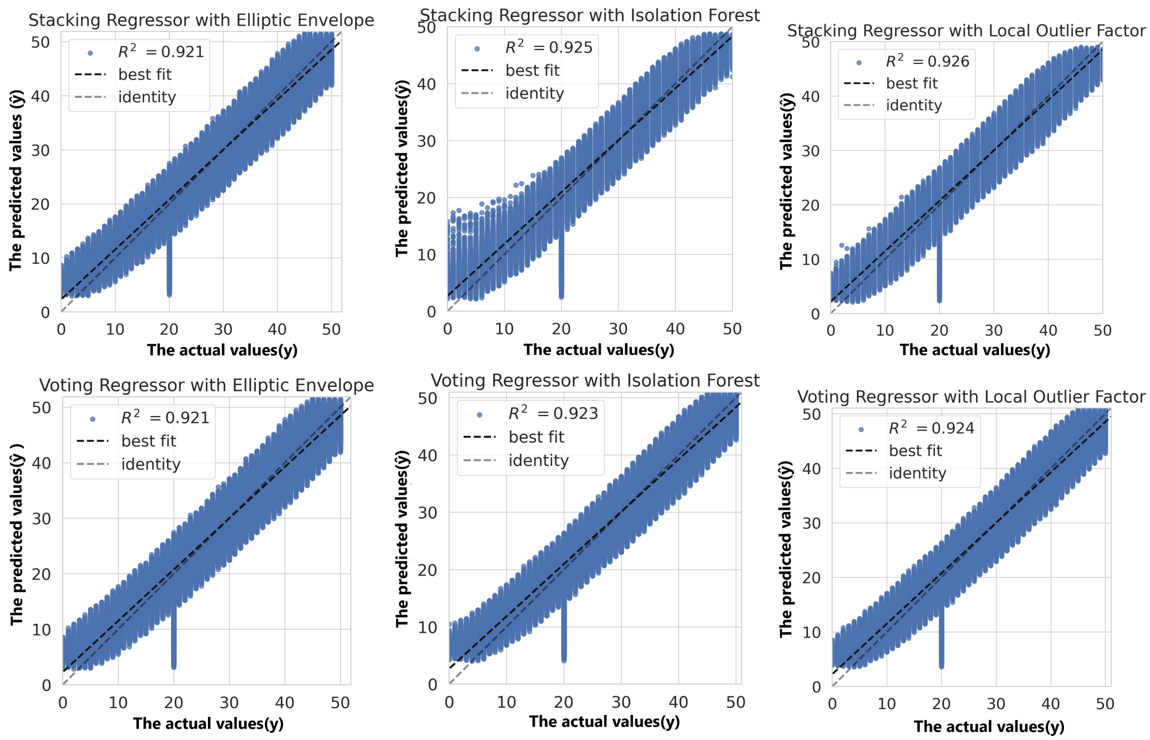


Figure 4.2: Evaluating Compression Efficiency of Stacking and Voting Ensemble Methods using Three Different Anomaly Detection Algorithms for Scenario One

4.3.3.2 Scenario Two (window size=2 hours and step size=1 hour)

In the scenario two,we once again utilize the sliding window technique, but with a different configuration. the window duration is specifically specified as two hours, with a step interval of one hour.

Table 4.6 evaluates the effectiveness of baseline regression models. The models were trained using a preprocessed dataset. data sets cleaned using various anomaly detection approaches, namely Elliptic Envelope, Isolation Forest, and Local Outlier Factor, respectively, Table 4.6 (a), Table 4.6 (b), and Table 4.6 (c) as part of the data cleaning process.

Table 4.6, shows that **(OMP, Elastic Net, Kernel Ridge)** models typically exhibit elevated levels of MAE, MSE, and RMSE in all evaluations. This occurs because they give priority to minimizing variance by reducing the coefficients of the model, which might potentially generate bias.

Table 4.6, shows that SVR model regularly exhibits slower performance, as indicated by greater

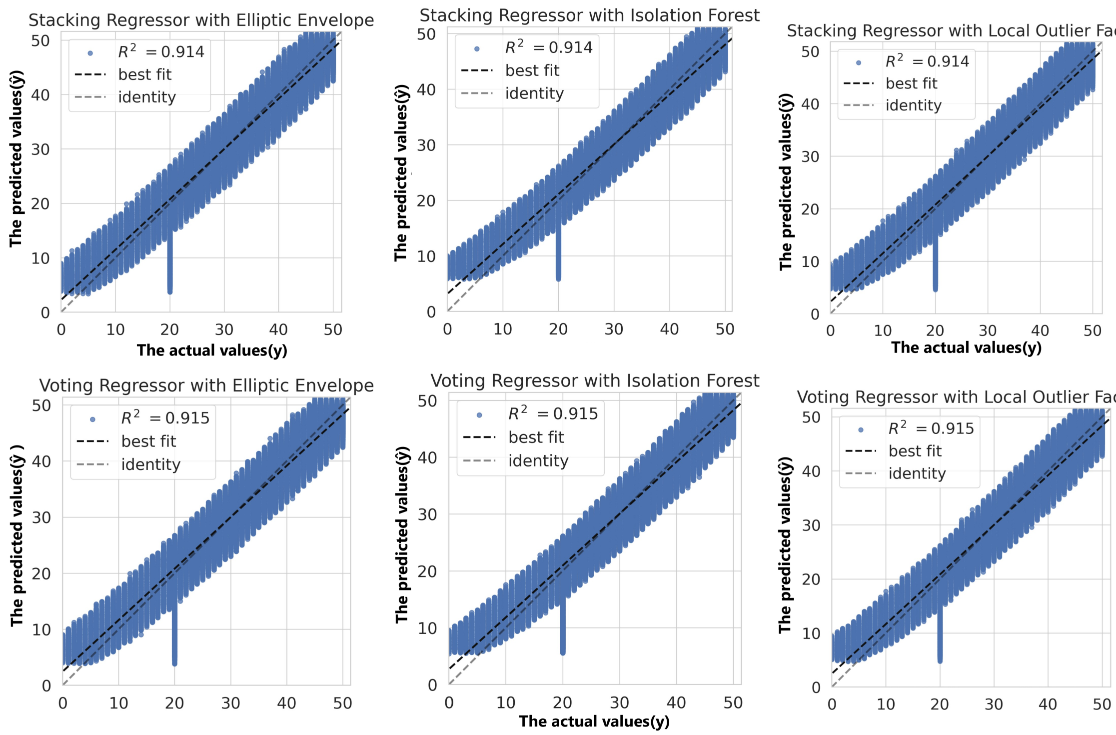


Figure 4.3: Evaluating Compression Efficiency of Stacking and Voting Ensemble Methods using Three Different Anomaly Detection Algorithms for Scenario Two

Total Time (TT) values, in comparison to other models. The reason for this is that it is a more intricate model that necessitates additional computational resources for training. Table 4.6, shows that Ridge Regression model consistently attains the lowest MAE, MSE, and RMSE compared to various anomaly detection methods. Therefore, it is an excellent option when prioritizing accuracy. Linear Regression model Similar to Ridge Regression, it demonstrates strong performance with low error in the majority of scenarios. If interpretability is a crucial factor alongside accuracy, this model should be considered due to its simplicity and ease of understanding. Based on our investigation, the optimal choice for anomaly detection model is based on your specific goals. Ridge Regression places a higher emphasis on accuracy, while Random Sample Consensus(RANSAC) places a higher emphasis on speed. Simpler models such as Ridge Regression or Linear Regression favor interpretability.

Table 4.7 displays a comparison of the performance of the top six linear regression models, which were chosen from a potentially larger set of models as depicted in Table 4.6. This selection ensures that we are evaluating models that exhibit a high level of initial performance.

Table 4.6: Comparison results of linear baseline models across various anomaly methods using cross-validation data for Scenario two.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE	TT (Sec)
Ridge Regression	2.93874	12.99353	3.60270	0.91151	0.17805	13.422
Linear Regression	2.93877	12.99407	3.60277	0.91150	0.17800	13.188
Least Angle Regression	2.93877	12.99407	3.60277	0.91150	0.17800	13.076
Bayesian Ridge	2.93877	12.99400	3.60276	0.91150	0.17800	13.382
Automatic Relevance Determination	2.93878	12.99412	3.60278	0.91150	0.17800	13.066
TheilSen Regressor	2.93793	13.00569	3.60445	0.91141	0.17548	19.358
Huber Regressor	2.93608	13.04076	3.60879	0.91110	0.17185	13.534
Support Vector Regression	2.93509	13.21567	3.63289	0.90994	0.16879	321.224
Random Sample Consensus	3.00916	13.88877	3.72477	0.90541	0.16784	12.992
Passive Aggressive Regressor	3.46844	18.41031	4.28031	0.87496	0.22362	13.094
Lasso Regression	5.05996	39.12001	6.25380	0.73532	0.35276	13.302
Lasso Least Angle Regression	5.05996	39.12001	6.25380	0.73532	0.35276	13.296
Elastic Net	6.19130	57.97262	7.61051	0.60907	0.47138	13.052
Orthogonal Matching Pursuit	6.15558	59.11820	7.68856	0.59927	0.40409	13.230
Kernel Ridge	29.67189	893.51732	29.88938	-5.06796	1.48609	1497.224

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE	TT (Sec)
Linear Regression	2.95033	12.84300	3.58228	0.91270	0.18941	4.594
Ridge Regression	2.95026	12.84240	3.58220	0.91270	0.18947	3.788
Least Angle Regression	2.95033	12.84300	3.58228	0.91270	0.18941	3.820
Bayesian Ridge	2.95032	12.84293	3.58227	0.91270	0.18942	3.812
Automatic Relevance Determination	2.95031	12.84277	3.58225	0.91270	0.18941	3.774
Huber Regressor	2.94917	12.84979	3.58316	0.91263	0.18812	4.252
TheilSen Regressor	2.95550	12.89661	3.58978	0.91233	0.18986	9.704
Support Vector Regression	2.93130	12.94596	3.59640	0.91201	0.18303	306.270
Random Sample Consensus	3.04543	13.94481	3.73184	0.90536	0.18093	3.804
Passive Aggressive Regressor	3.67285	21.07906	4.54010	0.85877	0.23911	3.762
Lasso Regression	5.05972	39.21306	6.26039	0.73508	0.36993	3.868
Lasso Least Angle Regression	5.05972	39.21305	6.26039	0.73508	0.36993	3.782
Orthogonal Matching Pursuit	6.17468	59.46657	7.71093	0.59714	0.41674	3.778
Elastic Net	6.33541	61.39914	7.82842	0.58707	0.50868	3.784
Kernel Ridge	30.94403	970.47458	31.15011	-5.59319	1.54067	1698.738

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE	TT (Sec)
Ridge Regression	2.95151	12.97082	3.59990	0.91174	0.18381	22.312
Automatic Relevance Determination	2.95157	12.97138	3.59997	0.91174	0.18376	21.932
Linear Regression	2.95157	12.97156	3.60000	0.91173	0.18376	22.796
Least Angle Regression	2.95157	12.97156	3.60000	0.91173	0.18376	22.440
Bayesian Ridge	2.95157	12.97147	3.59999	0.91173	0.18376	22.234
Huber Regressor	2.94992	13.00226	3.60403	0.91147	0.17973	22.950
TheilSen Regressor	2.95372	13.01618	3.60611	0.91140	0.18164	29.518
Support Vector Regression	2.91494	13.02575	3.60652	0.91122	0.16617	314.924
Random Sample Consensus	2.99686	13.54716	3.67878	0.90779	0.17754	22.156
Passive Aggressive Regressor	3.40930	18.61324	4.27944	0.87160	0.18101	22.342
Lasso Regression	5.04770	39.01150	6.24412	0.73625	0.35980	22.612
Lasso Least Angle Regression	5.04770	39.01150	6.24412	0.73625	0.35980	22.684
Elastic Net	6.22489	58.99409	7.67528	0.60269	0.48643	22.228
Orthogonal Matching Pursuit	6.17328	59.45611	7.71043	0.59709	0.41236	22.448
Kernel Ridge	30.28623	930.61963	30.49891	-5.32469	1.51230	1934.114

Table 4.7: Comparison results of selected linear baseline models across various anomaly methods using hold-out data Scenario two.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE
Ridge Regression	2.92966	12.78439	3.57553	0.91457	0.17924
Linear Regression	2.92966	12.78438	3.57552	0.91457	0.17921
Least Angle Regression	2.92966	12.78438	3.57552	0.91457	0.17921
Bayesian Ridge	2.92966	12.78438	3.57553	0.91457	0.17922
Automatic Relevance Determination	2.92968	12.78454	3.57555	0.91457	0.17922
TheilSen Regressor	2.92967	12.83779	3.58299	0.91422	0.17595

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE
Linear Regression	2.94933	12.73257	3.56827	0.91492	0.19193
Ridge Regression	2.94933	12.73265	3.56828	0.91492	0.19196
Least Angle Regression	2.94933	12.73257	3.56827	0.91492	0.19193
Bayesian Ridge	2.94933	12.73258	3.56827	0.91492	0.19194
Automatic Relevance Determination	2.94932	12.73257	3.56827	0.91492	0.19193
Huber Regressor	2.94747	12.73231	3.56824	0.91492	0.19092

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE
Ridge Regression	2.94513	12.78511	3.57563	0.91457	0.18646
Automatic Relevance Determination	2.94516	12.78524	3.57565	0.91457	0.18644
Linear Regression	2.94514	12.78515	3.57563	0.91457	0.18644
Least Angle Regression	2.94514	12.78515	3.57563	0.91457	0.18644
Bayesian Ridge	2.94514	12.78514	3.57563	0.91457	0.18644
Huber Regressor	2.94254	12.80621	3.57858	0.91443	0.18298

Table 4.8, shows that the Voting Regressor performs better than Stacking Regressor in this specific scenario, especially when assessing metrics like MAE, MSE, and RMSE. Figure 5.6 Compares the performance of two different regression models, the Stacking Regressor and the Voting Regressor, in the specific task of predicting traffic flow. The performance is assessed using the R^2 metric. Both the Stacking Regressor and Voting Regressor produce a remarkably high R^2 value, nearing 0.91, when used to all three anomaly detection approaches (Elliptic Envelope, Isolation Forest, Local Outlier Factor). Both models accurately depict the overall traffic flow pattern, regardless of the method used to detect unusual data points. The Voting Regressor consistently achieves a slightly higher R^2 value compared to the Stacking Regressor for each outlier identification method.

Table 4.8: Performance results of Stacking Regressor versus Voting Regressor using cross-validation data Scenario two.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE
Voting Regressor	2.93816	12.99084	3.60233	0.91152	0.17756
Stacking Regressor	2.94112	13.02265	3.60664	0.91130	0.17789

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.95228	12.88261	3.58758	0.91240	0.18774
Voting Regressor	2.95006	12.84333	3.58231	0.91269	0.18920

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.95789	13.07890	3.61441	0.91095	0.18250
Voting Regressor	2.95104	12.97380	3.60027	0.91171	0.18308

Table 4.9: Performance results of Stacking Regressor versus Voting Regressor on the testing data set Scenario two.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE
Voting Regressor	2.92915	12.78755	3.57597	0.91455	0.17865
Stacking Regressor	2.93112	12.84044	3.58336	0.91420	0.17627

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.96087	12.80848	3.57889	0.91441	0.19818
Voting Regressor	2.94897	12.73203	3.56820	0.91492	0.19177

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.94511	12.81544	3.57987	0.91436	0.18322
Voting Regressor	2.94455	12.78684	3.57587	0.91456	0.18586

4.3.3.3 Scenario Three (window size=3 hours and step size=1.5 hours)

For scenario three, we employ the sliding window technique once more, but with a distinct configuration. The window duration is precisely defined at three hours, with a step interval of 1.5 hours. This configuration differs from scenario one and two, maybe to capture specific traffic patterns with a specific amount of time detail.

Table 4.10 explicitly investigates the efficacy of various linear baseline models in identifying

Table 4.10: Comparison results of linear baseline models across various anomaly methods using cross-validation data for Scenario three.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE	TT (Sec)
Linear Regression	2.75661	10.50387	3.24092	0.92916	0.15715	11.328
Ridge Regression	2.75640	10.50371	3.24090	0.92916	0.15722	11.494
Least Angle Regression	2.75661	10.50387	3.24092	0.92916	0.15715	11.494
Bayesian Ridge	2.75659	10.50383	3.24092	0.92916	0.15716	11.508
Automatic Relevance Determination	2.75657	10.50391	3.24093	0.92916	0.15715	11.402
Huber Regressor	2.75566	10.50784	3.24153	0.92913	0.15774	11.484
Support Vector Regression	2.74671	10.52528	3.24401	0.92907	0.15313	155.228
TheilSen Regressor	2.76157	10.54600	3.24742	0.92885	0.15663	16.346
Random Sample Consensus	2.90379	12.05698	3.47098	0.91874	0.16718	11.406
Passive Aggressive Regressor	3.35135	16.77588	4.07523	0.88756	0.16979	11.224
Lasso Regression	5.34997	43.69405	6.60972	0.70585	0.36929	11.264
Lasso Least Angle Regression	5.34997	43.69404	6.60972	0.70585	0.36929	11.454
Orthogonal Matching Pursuit	5.35961	44.77540	6.69122	0.69796	0.32990	11.288
Elastic Net	6.37847	60.82949	7.79727	0.59174	0.48132	11.334
Kernel Ridge	29.71303	893.47341	29.88915	-5.02635	1.51215	502.048

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE	TT (Sec)
Linear Regression	2.77088	10.61480	3.25783	0.92851	0.17080	4.104
Ridge Regression	2.77077	10.61515	3.25788	0.92851	0.17089	3.274
Least Angle Regression	2.77088	10.61480	3.25783	0.92851	0.17080	3.228
Bayesian Ridge	2.77086	10.61483	3.25784	0.92851	0.17081	3.312
Automatic Relevance Determination	2.77087	10.61490	3.25785	0.92851	0.17082	3.274
Huber Regressor	2.77162	10.64099	3.26179	0.92836	0.17253	3.698
TheilSen Regressor	2.77474	10.64179	3.26200	0.92832	0.16949	8.164
Support Vector Regression	2.76453	10.69677	3.27011	0.92808	0.16951	155.200
Random Sample Consensus	2.95620	12.55492	3.53978	0.91550	0.17864	3.314
Passive Aggressive Regressor	3.05193	13.56227	3.68018	0.90853	0.15761	3.288
Lasso Regression	5.41637	44.71503	6.68478	0.69991	0.40294	3.348
Lasso Least Angle Regression	5.41637	44.71503	6.68478	0.69991	0.40294	3.264
Orthogonal Matching Pursuit	5.42169	45.57345	6.75036	0.69294	0.35310	3.270
Elastic Net	6.57768	65.31575	8.07517	0.56321	0.52781	3.278
Kernel Ridge	30.87519	963.97754	31.04612	-5.50358	1.55795	491.776

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE	TT (Sec)
Support Vector Regression	2.73046	10.36492	3.21916	0.93018	0.15282	161.780
Linear Regression	2.76764	10.59154	3.25432	0.92863	0.16374	12.654
Ridge Regression	2.76745	10.59151	3.25431	0.92863	0.16381	12.570
Least Angle Regression	2.76764	10.59154	3.25432	0.92863	0.16374	12.536
Bayesian Ridge	2.76761	10.59152	3.25431	0.92863	0.16375	11.804
Automatic Relevance Determination	2.76762	10.59180	3.25436	0.92863	0.16375	12.030
Huber Regressor	2.76704	10.60139	3.25581	0.92858	0.16473	12.410
TheilSen Regressor	2.77108	10.63317	3.26070	0.92835	0.16416	17.064
Random Sample Consensus	2.86350	11.62139	3.40803	0.92181	0.16405	12.080
Passive Aggressive Regressor	3.54857	19.70329	4.32617	0.86381	0.18286	12.048
Lasso Regression	5.30694	42.93366	6.55085	0.71147	0.38207	12.824
Lasso Least Angle Regression	5.30694	42.93366	6.55085	0.71147	0.38207	12.202
Orthogonal Matching Pursuit	5.39909	45.27710	6.72852	0.69481	0.34432	12.048
Elastic Net	6.43171	62.14299	7.87882	0.58373	0.50145	12.796
Kernel Ridge	30.16363	920.80056	30.33853	-5.20941	1.52847	491.076

Table 4.11: Comparison results of selected linear baseline models across various anomaly methods using hold-out data for Scenario three.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE
Linear Regression	2.76482	10.58453	3.25339	0.93000	0.16291
Ridge Regression	2.76478	10.58454	3.25339	0.93000	0.16294
Least Angle Regression	2.76482	10.58453	3.25339	0.93000	0.16291
Bayesian Ridge	2.76481	10.58453	3.25339	0.93000	0.16291
Automatic Relevance Determination	2.76479	10.58437	3.25336	0.93000	0.16291
Huber Regressor	2.76441	10.59233	3.25459	0.92995	0.16363

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE
Linear Regression	2.78049	10.70021	3.27112	0.92923	0.17869
Ridge Regression	2.78047	10.70048	3.27116	0.92923	0.17873
Least Angle Regression	2.78049	10.70021	3.27112	0.92923	0.17869
Bayesian Ridge	2.78048	10.70025	3.27112	0.92923	0.17870
Automatic Relevance Determination	2.78048	10.70021	3.27112	0.92923	0.17870
Huber Regressor	2.78263	10.73770	3.27684	0.92899	0.18081

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE
Support Vector Regression	2.71617	10.19717	3.19330	0.93256	0.15414
Linear Regression	2.78248	10.74726	3.27830	0.92892	0.17297
Ridge Regression	2.78246	10.74735	3.27831	0.92892	0.17301
Least Angle Regression	2.78248	10.74726	3.27830	0.92892	0.17297
Bayesian Ridge	2.78248	10.74727	3.27830	0.92892	0.17298
Automatic Relevance Determination	2.78247	10.74718	3.27829	0.92892	0.17298

anomalies in scatter plots. The models are evaluated based on their performance using different anomaly detection techniques, including Elliptic Envelope, Isolation Forest, and Local Outlier Factor.4.10, we have noted the following observations:

The efficacy of the models varies according on the anomaly detection technique employed. There is no singular optimal model that outperforms all measurements and methods for anomaly identification. in Table 4.10 (a) Ridge Regression exhibits the lowest MAE, MSE, and RMSE. while in Table 4.10 (b) shows that SVR exhibits the lowest MAE, whereas Linear Regression achieves the lowest . Mean squared error (MSE) and root mean squared error (RMSE). In addition 4.10 (c) SVR achieves the lowest MAE, MSE, and RMSE.

Table 4.11 presents a comparison of the performance of the top six linear regression models. These models were selected from a possibly larger group of models, as shown in Table 4.10. This

Table 4.12: Performance results of Stacking Regressor versus Voting Regressor using cross-validation data for Scenario three.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE
Voting Regressor	2.75636	10.50396	3.24094	0.92916	0.15726
Stacking Regressor	2.75695	10.50980	3.24184	0.92911	0.15744

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.77095	10.60964	3.25708	0.92853	0.16988
Voting Regressor	2.77089	10.61834	3.25837	0.92849	0.17111

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.71133	10.13095	3.18288	0.93169	0.15171
Voting Regressor	2.74448	10.39842	3.22459	0.92992	0.16103

Table 4.13: Performance results of Stacking Regressor versus Voting Regressor on the testing data set for Scenario three.

(a) Outliers Method: Elliptic Envelope

Model	MAE	MSE	RMSE	R2	MAPE
Voting Regressor	2.76468	10.58522	3.25349	0.93000	0.16303
Stacking Regressor	2.76499	10.58646	3.25368	0.92999	0.16333

(b) Outliers Method: Isolation Forest

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.78079	10.69987	3.27106	0.92924	0.17859
Voting Regressor	2.78073	10.70549	3.27192	0.92920	0.17905

(c) Outliers Method: Local Outlier Factor

Model	MAE	MSE	RMSE	R2	MAPE
Stacking Regressor	2.70823	10.10745	3.17922	0.93316	0.15541
Voting Regressor	2.75500	10.50485	3.24112	0.93053	0.16890

selection guarantees that we are assessing models that demonstrate a high level of initial performance.

Table 4.12 (a) shows that the Voting Regressor performs better than the Stacking Regressor in terms of MAE, MSE, and RMSE when the Elliptic Envelope outlier identification approach is utilized. Nevertheless, in Tables 4.12 (b) and Table 4.12 (c) exhibit a contrasting pattern, indicating that the Stacking Regressor may be superior in such situations.

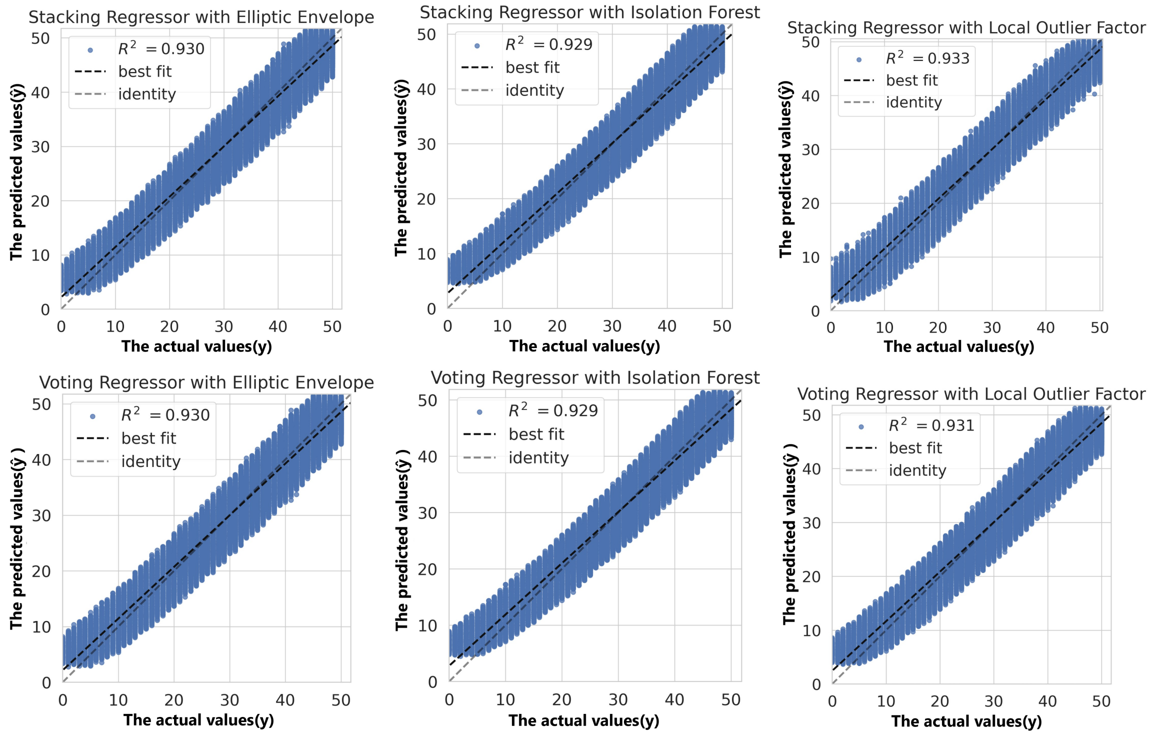


Figure 4.4: Evaluating Compression Efficiency of Stacking and Voting Ensemble Methods using Three Different Anomaly Detection Algorithms for Scenario Three

Table 4.13 presents a comparison of the performance of two regression models, namely Stacking Regressor and Voting Regressor, using the testing data set for Scenario Three. It shows that the Stacking Regressor has a higher level of performance in comparison to the Voting Regressor. The Stacking Regressor outperforms the Voting Regressor in three out of the five metrics (MAE, MSE, and RMSE) for the Isolation Forest and Local Outlier Factor techniques.Curiously, the Elliptic Envelope technique does not exhibit this pattern, as the Voting Regressor seems to possess a little edge. This implies that the selection of an outlier detection technique can impact the optimal selection of a regression model.

Figure 5.7 presents a comparison of the performance of two distinct regression models, namely the Stacking Regressor and the Voting Regressor, in the specific job of predicting traffic flow. The performance is evaluated using the R^2 metric.

Both the Stacking Regressor and Voting Regressor yield a significantly high R^2 value, approaching 0.93, when applied to all three anomaly detection methods (Elliptic Envelope, Isolation Forest, Local Outlier Factor). Both models effectively represent the general traffic flow pattern, irrespective of the approach employed to identify anomalous data points.

The Voting Regressor consistently achieves an identical R^2 value compared to the Stacking Regressor for each outlier identification method.

4.4 Summary

Our study investigates the effectiveness of a multilevel ensemble strategy combined with anomaly detection for predicting traffic flow. We conduct experiments in three different scenarios, each using a distinct sliding window configuration. Our investigation of comparisons yielded numerous significant insights, summarized as follows.

The selection of appropriate models is greatly influenced by the choice of anomaly detection technology. During Scenario 1,SVR demonstrates the lowest error rate for all techniques, but it also had the longest training time. The Stacking Regressor has proven to be a formidable option for ensemble learning in this particular scenario.

In Scenario 2, where the importance of interpretability was taken into account, simpler models such as Ridge or Linear Regression considers as potential alternatives to Ridge Regression, which placed a higher emphasis on accuracy. While, the Voting Regressor performed better than the Stacking Regressor in this particular scenario.

Ultimately, Scenario 3 demonstrates that the ideal ensemble model relies heavily on the particular anomaly detection approach used. The Stacking Regressor yield superior results when combined with the Isolation Forest and Local Outlier Factor methods, while the Voting Regressor shows greater performance when paired with the Elliptic Envelope method.

Although there are variations particular to each circumstance, a similar theme is evident in all arrangements. Both Stacking and Voting Regressors consistently achieve high R^2 values (around 0.91-0.93) for all anomaly detection methodologies. This demonstrates their efficacy in capturing the whole traffic flow pattern, irrespective of the selected anomaly detection technique. These findings emphasize the need of taking into account the interaction between anomaly detection and model selection to achieve the most effective traffic flow forecast.

Chapter 5

Clustering Based Approach for Enhanced Characterization of Anomalies in Traffic Flows

5.1 Introduction

Effective traffic management is a fundamental aspect of urban planning, with direct implications for the daily experiences of commuters and the economic prosperity of cities. Traffic anomalies, characterized by substantial variations from normal traffic patterns, pose significant difficulties in ensuring smooth traffic flow. These anomalies can manifest as either sudden increases in traffic volume, resulting in congestion and surpassing the capacity of the route, or sudden decreases in traffic volume, which could indicate occurrences like accidents, road closures, or faults in traffic infrastructure[106].

Precise identification and analysis of traffic anomalies are crucial for minimizing their disruptive impacts and enhancing overall traffic management. By identifying these abnormalities at an early stage, traffic management systems can promptly implement interventions to reduce traffic congestion and solve any safety concerns. Furthermore, comprehending the nature and distribution of these anomalies might offer valuable perspectives for long-term urban planning and infrastructure development[51], [107].

This work builds on our prior research where we utilized three sophisticated anomaly detection methods—Elliptic Envelope, Isolation Forest, and Local Outlier Factor—to discover abnormal traffic patterns. The aforementioned strategies were utilized on a carefully prepared dataset, which was further improved using several windowing methods in order to optimize the effectiveness of anomaly identification. After completing the detection phase, we employe clustering methods in this research, notably K-Means and Hierarchical Clustering, to divide the discovered anomalies into segments. The utilization of these clustering algorithms played a crucial role in identifying the most suitable number of clusters, enabling a thorough description of each anomalous cluster through comprehensive visualization[108].

Our approach not only facilitates the efficient detection of traffic abnormalities but also offers a comprehensive comprehension of their spatial and temporal patterns. Accurate identification and analysis of anomalies in the transportation system are essential for developing effective strategies to control traffic and make informed decisions about urban development. This process ultimately improves the efficiency and safety of transportation networks.

The importance of employing various anomaly detection techniques resides in their complementary strengths. The Elliptic Envelope approach is based on the premise that the normal data is distributed according to a multivariate Gaussian distribution. It aims to detect locations that vary considerably from this assumption. Unlike profiling normal data, Isolation Forest focuses on isolating anomalies, making it highly effective for discovering anomalies in datasets with complex distributions. The Local Outlier Factor algorithm measures the extent to which a certain data point differs in density from its neighboring points. This makes it effective for detecting anomalies in datasets that have different densities. Through the integration of different methodologies, our research guarantees a strong and all-encompassing identification of anomalies.

Utilizing windowing techniques to preprocess the dataset significantly improved the accuracy of anomaly identification. Various window sizes and configurations were experimented with to capture temporal dependencies and patterns in the traffic data, ensuring that both short-term fluctuations and long-term trends were accurately depicted. This step was important in converting unprocessed traffic data into a format that is appropriate for efficient anomaly detection.

After detecting anomalies, the application of K-Means and Hierarchical Clustering allows for a more in-depth analysis of their characteristics. The K-Means clustering algorithm, with its iterative refinement procedure, efficiently classified anomalies into separate groups according to their distinctive characteristics. Hierarchical Clustering, a method that constructs a tree-like structure of clusters, enables us to comprehend the intricate arrangement of anomalies and their relationships. By employing this dual methodology for clustering, we were able to ascertain the most suitable quantity of clusters and obtain a comprehensive visualization of the anomalies[109].

The characterization of these clusters through visualization and mapping reveals distinct geographical and temporal patterns. For example, particular clusters may represent regular traffic congestion that occurs during certain times of the day or week, while other clusters may reflect occasional instances. Gaining insight into these patterns enables traffic management authorities to effectively address present anomalies and proactively prepare for forthcoming disruptions. This knowledge is crucial for raising the efficiency of traffic, improving road safety, and optimizing the infrastructure of urban transportation

In conclusion, our comprehensive approach builds on our prior research paper in which we utilized Elliptic Envelope, Isolation Forest, and Local Outlier Factor for identifying anomalies. In this study, we enhance our analysis by employing clustering techniques, namely K-Means and Hierarchical Clustering, to further investigate the attributes of the found anomalies. By utilizing sophisticated windowing approaches, we achieved reliable anomaly identification, and by subsequently applying clustering, we obtain comprehensive understanding of their geographical and temporal patterns.

5.2 Motivation and Preliminaries

5.3 Methodology

Figure 5.1 depicts a thorough procedure for managing and examining data, with a particular emphasis on identifying unusual patterns and clustering them. The procedure is segmented into multiple discrete phases: Data Collection, Preprocessing, Clustering, Characterization, and Insights. Every phase signifies a crucial stage in converting unprocessed data into important understandings, highlighting the methodical approach necessary for efficient data analysis.

The process initiates with Data Collection, when unprocessed data is acquired from several sources, represented by camera ,sensors and other resources. The unprocessed dataset is subsequently subjected to the Preprocessing Phase. In our prior study, we utilized sliding windowing methods to preprocess data. We employed three distinct scenarios: a window size of 1 hour with a step size of 30 minutes, a window size of 2 hours with a step size of 1 hour, and a window size of 3 hours with a step size of 1.5 hour. In addition, we utilized three anomaly detection algorithms: Local Outlier Factor, Elliptic Envelope, and Isolation Forest [1]. By employing these techniques, the anomalies were successfully identified and isolated, leading to a refined dataset that is now prepared for additional study. After preprocessing phase, we obtain two datasets for the training process: the cleaned dataset, which contains only the normal data without any anomalies, and the dataset specifically for the discovered anomalies. The dataset containing the anomalies is processed in the Clustering Phase, during which the data is sorted into several groups or clusters, labeled for example Cluster 1, Cluster 2, Cluster 3, and so on, up to Cluster k. This clustering technique employs two algorithms, namely K-Means and Hierarchical Clustering, to categorize data points with comparable characteristics or criteria. Organizing the datasets in this manner is crucial for facilitating subsequent analysis, as it enhances manageability and meaning. After the data has been clustered into clusters, the next step is the characterization and analysis of these clusters. During this phase, a thorough examination is conducted on each cluster to gain a comprehensive understanding of the data's characteristics and the underlying patterns within it. This step entails employing statistical analysis, pattern recognition, or other analytical tools to identify common features or underlying factors. In addition, we provide a visual representation of the clusters on a map.The objective is to acquire a more profound comprehension of the data and the elements that contribute to its arrangement and behavior inside each cluster.

The last phase of the process is Insights. By doing a thorough study and characterization of the clusters, we are able to extract practical and useful insights. These insights can assist in making well-informed decisions, spotting possible problems, and adopting measures to optimize processes

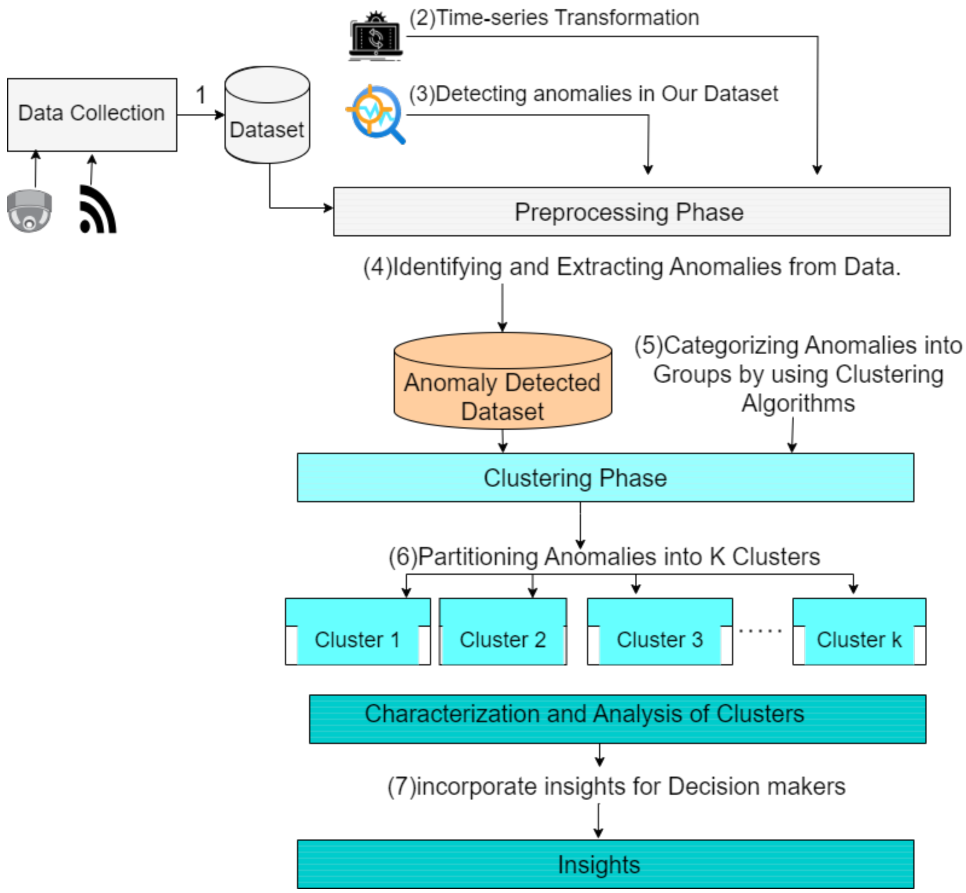


Figure 5.1: Proposed Model for Traffic Flow Anomaly Detection and Cluster Characterization.

or resolve observed patterns. The complete procedure, starting from the gathering of data to the extraction of meaningful conclusions, emphasizes the significance of employing a methodical approach in data analysis to convert unprocessed data into usable information.

5.4 Experimental Setup and Evaluation

The performance of clustering is crucial as clustered data are usually assessed manually and subjectively to ascertain their significance. When the true clustering data labels are not known, other intrinsic measures can be employed to assess the efficacy of the clustering technique. In this paper, we employed the most widely used metrics, which are described as follows [110].

5.4.1 Silhouette coefficient

The Silhouette coefficient is a quantitative metric utilized to assess the quality of a clustering method. This analysis considers both the cohesion and separation of clusters, offering an understanding of the degree of similarity between an object and its own cluster in comparison to other clusters. The calculation of the Silhouette coefficient for a single data point comprises two key components: a and b.

The value a is the mean distance from a point to all other points within the same cluster, indicating the level of cohesion within the cluster. Alternatively,b represents the mean distance from a point to all points in the closest cluster that the point does not belong to. This metric quantifies the separation between clusters [111].

The Silhouette coefficient s for a data point is computed using the following formula:

$$s = \frac{b - a}{\max(a, b)}$$

The coefficient varies between -1 and 1, with a value near to 1 indicating a strong match between the data point and its own cluster, but a poor match with neighboring clusters. A value that is approximately equal to 0 indicates that the data point is located precisely on or in close proximity to the decision boundary between two adjacent clusters. On the other hand, a value close to -1 suggests that the data point might have been incorrectly assigned to the cluster.

By averaging the Silhouette coefficients of all data points in a dataset, This helps in evaluating the appropriateness of the data clustering and provides guidance for enhancing the clustering strategy if needed.

5.4.2 Calinski–Harabasz Index

The Calinski-Harabasz Index, also known as the Variance Ratio Criterion, is a quantitative metric utilized to assess the effectiveness of clustering algorithms by measuring the dispersion of the clusters. The term refers to the proportion of the total dispersion between clusters to the total dispersion within clusters. This index facilitates the assessment of the degree of separation and compactness of the clusters.

Mathematically, the Calinski-Harabasz Index CH for a clustering result is calculated using the following formula:

$$CH = \frac{\text{trace}(B_k)/(k-1)}{\text{trace}(W_k)/(n-k)}$$

In this formula, $\text{trace}(B_k)$ denotes the trace of the dispersion matrix for between-group differences, whereas $\text{trace}(W_k)$ denotes the trace of the dispersion matrix for within-group differences. In this context, k represents the number of clusters, while n denotes the total number of data points.

The between-group dispersion matrix B_k quantifies the extent to which the cluster centroids deviate from the overall centroid of the data. This indicates the level of dissimilarity between the groups. Conversely, the within-group dispersion matrix W_k quantifies the dispersion of data points within each cluster, providing information about the tightness of the clusters [112].

Higher values of the Calinski-Harabasz Index indicate better-defined clusters, characterized by higher between-cluster dispersion and lower within-cluster dispersion. This index is very valuable for evaluating distinct clustering results and determining the most suitable number of clusters.

5.4.3 Davies–Bouldin Index

The Davies-Bouldin Index is a quantitative metric utilized to assess the effectiveness of a clustering algorithm. The metric calculates the average similarity ratio between each cluster and its most similar cluster, which indicates the level of separation between the clusters. A lower Davies-Bouldin Index value indicates superior clustering results, as it signifies that the clusters are compact and well-separated.

In order to calculate the Davies-Bouldin Index DB for a clustering result, several steps must be undertaken. Firstly, the average distance between each point in cluster i and the centroid of the same cluster is computed. The measure S_i indicates the scatter of the cluster. Next, for each pair of clusters i and j, the distance d_{ij} between the centroids of the clusters is computed [113].

Then, for each cluster i, the cluster j that maximizes the ratio

$$R_{ij} = \frac{S_i + S_j}{d_{ij}}$$

has been identified. The ratio R_{ij} quantifies the degree of similarity between cluster i and cluster j. The Davies-Bouldin Index is calculated by taking the average of the maximal ratios R_i across all of the clusters.

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} \left(\frac{S_i + S_j}{d_{ij}} \right)$$

where k is the number of clusters, S_i and S_j are the average distances within clusters i and j, respectively, and d_{ij} is the distance between the centroids of clusters i and j.

In summary, the Davies-Bouldin Index quantifies the average similarity between each cluster and its closest similar cluster. Lower values of this index indicate higher clustering quality, with clusters being denser and more distinct from one other.

5.4.4 Dataset

The study used a dataset that includes information about intersections and their associated attributes (as displayed in Table 5.1). Each intersection in the dataset is identified by a unique identifier known as "INT ID" (Intersection Number), which guarantees unambiguous differentiation. The "Intersection Name" field contains informative labels for each intersection, usually comprising the names of the two streets that intersect. The "ARRONDISSEMENT" field provides information on the specific region where each intersection is located, which helps with the clustering process [104].

Furthermore, the dataset includes geographical coordinates for each intersection. The "Longitude" and "Latitude" fields precisely indicate the horizontal and vertical coordinates, respectively. The coordinates are crucial for geographic analysis and visualization applications. The congestion level field provides information about the number of vehicles seen at the intersection over a specific time period. The data is collected through systematic monitoring of the number of vehicles at the intersection, with measurements recorded at five-minute intervals.

Our research utilizes clustering algorithms, notably K-Means and Hierarchical Clustering, to leverage detected anomalies. These anomalies were obtained from a dataset in our earlier research [1]. This dataset exclusively consists of frequency pattern features, which were utilized for the purpose of segmenting these abnormalities.

Table 5.1: Dataset Description

Field Name	**Description**
INT ID (Intersection Number)	Unique identifier for each intersection.
Intersection Name	Descriptive label for the intersection.
ARRONDISSEMENT (Region Name)	Region where the intersection is located.
Longitude	East-west geographic coordinate.
Latitude	North-south geographic coordinate.
Congestion Level (Number of Vehicles)	Number of vehicles observed at the intersection measured every 5 minutes.

Our study focuses on a selected simulated region inside the city of Montreal, as depicted in the Figure 3.4. We divide the simulation area into several zones and examine our challenge for each zone autonomously. despite the scale of the network, whether it is large or widespread, it will be separated into zones, and the clustering algorithms will be applied individually to each zone. Our simulation included 19 traffic lights in each zone, showcasing the scalability of our system.

5.4.5 Clustering pipeline setups

Our research utilizes two clustering techniques, namely K-Means and Agglomerative Clustering, to evaluate datasets that contain identified anomalies. The justification for utilizing these strategies is based on their mutually reinforcing advantages and the valuable perspectives they offer.

The K-Means clustering algorithm is selected due to its high efficiency and simplicity in dividing a dataset into several clusters. This approach operates by reducing variance within each cluster, rendering it particularly efficient for clusters that have a spherical shape. K-Means is computationally efficient, making it particularly suitable for handling large datasets. Additionally, it facilitates for straightforward implementation and interpretation of results.

Agglomerative Clustering, a kind of hierarchical clustering, is chosen for its capacity to manage clusters with diverse shapes and sizes. This approach builds a hierarchical structure of clusters by repeatedly combining or dividing existing based clusters on a chosen linkage criterion(such as single, complete, or average linkage). Agglomerative Clustering's flexibility enables it to uncover complex cluster structures that K-Means may overlook.

Both clustering approaches are implemented using the Scikit-learn library [114], a powerful and widely-used machine learning library in Python. Scikit-learn offers efficient and optimized versions of K-Means and Agglomerative Clustering algorithms, along by a range of hyperparameters that may be adjusted to maximize performance for our particular datasets.

5.4.6 Determining the optimal number of clusters

One way to evaluate the quality of clustering is by computing performance metrics and then comparing the results to determine the best option. Data patterns within the same group will probably have similar characteristics, while data patterns in other groups are expected to possess significantly diverse properties and features. However, determining the ideal number of clusters is one of the most difficult components of the clustering process.

The elbow method can determine the optimal number of clusters for a given data pattern by executing the clustering algorithm with several initial cluster assignments (ranging from 2 to 18) and evaluating the distortion score for each cluster assignment. The results are then plotted for analysis.

The inflection point or the elbow of a curve signifies the stage where the marginal benefits of adding another cluster no longer outweigh the diminishing returns. The Figures 5.2,5.3,5.4 display elbow plots of distortion scores for KMeans and Agglomerative clustering. The plots are generated using three distinct anomaly detection algorithms—Elliptic Envelope, Isolation Forest, and Local Outlier Factor. The Figures correspond to Scenarios One, Two, and Three. Each plot displays the distortion score in relationship with the number of clusters (k), with a dotted line indicating the elbow point that signifies the optimal number of clusters. For example, the Figure 5.2, the ideal value of k for KMeans clustering is determined to be 6 for Elliptic Envelope, 5 for Isolation Forest, and 6 for Local Outlier Factor. The corresponding distortion scores are 5396.535, 4185.090, and 5233.685, respectively. The ideal number of clusters (k) for Agglomerative clustering is 6 when using Elliptic Envelope, 5 when using Isolation Forest, and 5 when using Local Outlier Factor. The corresponding distortion scores are 5505.354, 4763.874, and 1320.200, respectively. The plots include green lines that show the run time for clustering. The elbow plots are crucial to determining the optimal number of clusters, balancing distortion score minimization and computational efficiency for each anomaly detection method

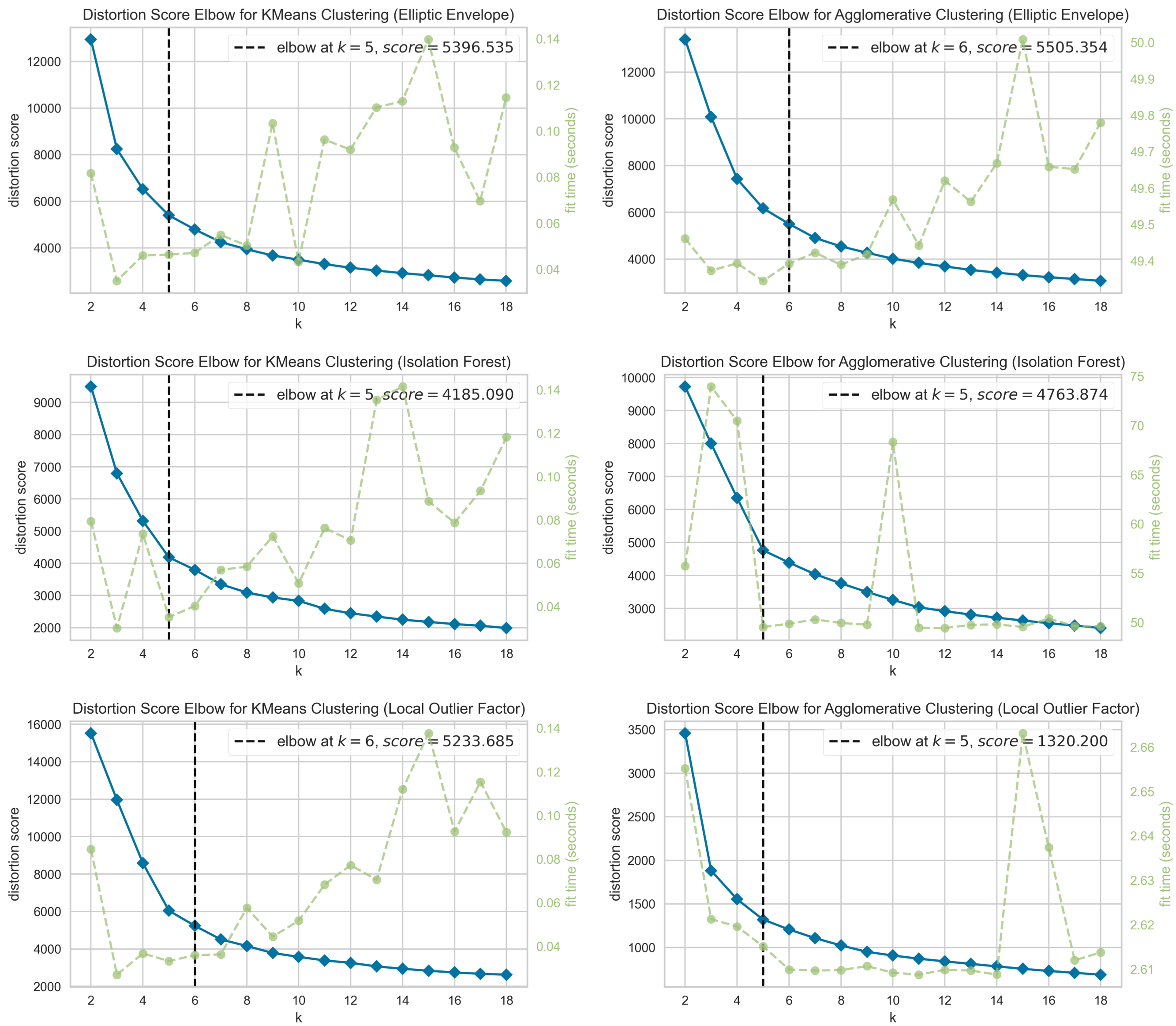


Figure 5.2: Distortion Score Elbow Plots for K-Means and Agglomerative Clustering Using Three Anomaly Detection Algorithms for Scenario One(window size=1 hour and step size=30 minutes)

5.4.7 Evaluation Results

In this section, we present the evaluation results of applying K-Means clustering to datasets that have been identified for anomalies using three distinct outlier detection techniques: Elliptic Envelope, Isolation Forest, and Local Outlier Factor, across three different scenarios with varying configuration settings (window sizes of 1 hour, 2 hours, and 3 hours).

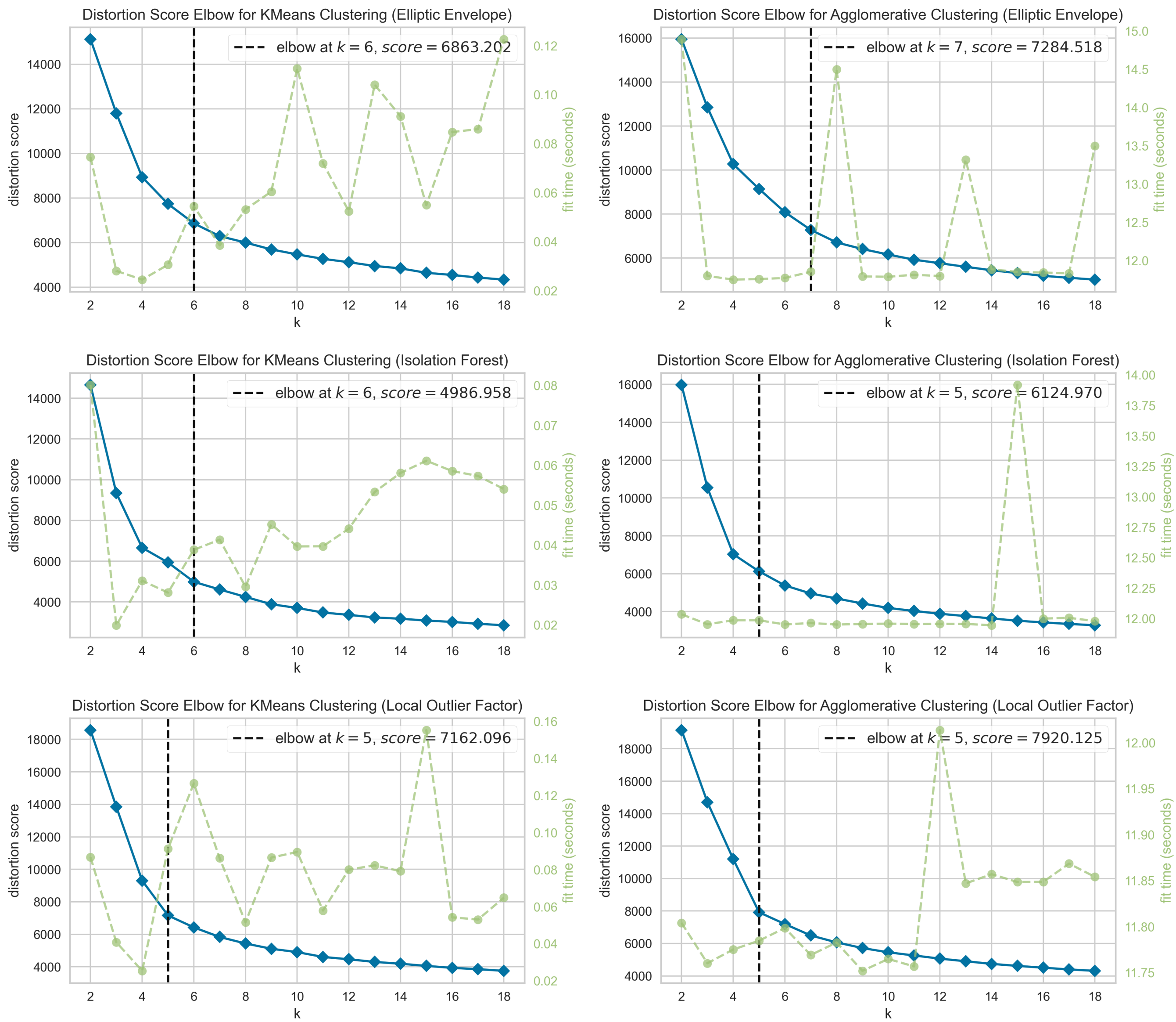


Figure 5.3: Distortion Score Elbow Plots for K-Means and Agglomerative Clustering Using Three Anomaly Detection Algorithms for Scenario Two(window size=2 hour and step size=1 hour

The Silhouette analysis, depicted in the Figures 5.5, 5.6, and 5.7, provides a thorough assessment of the clustering performance under various scenarios utilizing K-Means clustering in combination with three outlier detection techniques: Elliptic Envelope, Isolation Forest, and Local Outlier Factor. The Silhouette coefficient quantifies the degree of similarity between an object and its own cluster relative to other clusters, with values ranging from -1 to 1. A high average Silhouette score

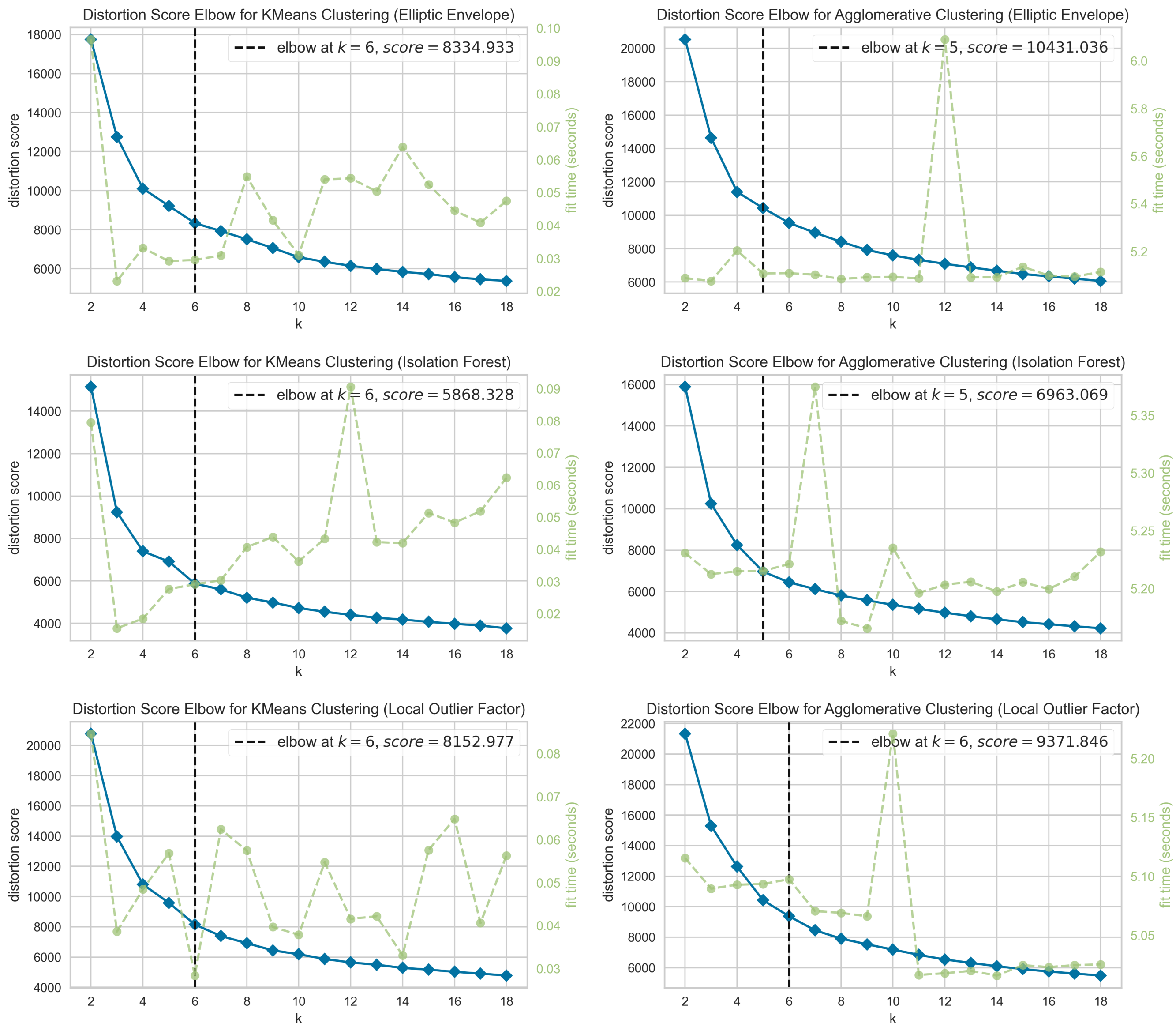


Figure 5.4: Distortion Score Elbow Plots for K-Means and Agglomerative Clustering Using Three Anomaly Detection Algorithms for Scenario Three(window size=3 hours and step size=1.5 hours

shows that the clusters have a significant degree of separation and cohesion, meaning they are well-separated and internally cohesive. Conversely, a low or negative score suggests that the clusters either overlap or lack clear definition.

Figure 5.5, showing Scenario One, demonstrates that the clustering performance is influenced by the choice of outlier detection method, as indicated by the Silhouette plots. The Elliptic Envelope approach yields a moderately high average Silhouette score, indicating a satisfactory balance between cluster cohesion and separation. The Isolation Forest approach achieves a little superior average Silhouette score, denoting more distinct clusters with minimal overlap among them. The Local Outlier Factor approach has robust performance, exhibiting a high average Silhouette score, indicating the presence of clearly defined and well-isolated clusters.

Figure 5.6 displays the Silhouette analysis for Scenario Two. In this case, the Elliptic Envelope approach remains effective, although the average Silhouette score is somewhat lower compared to Scenario One, suggesting a slight decrease in the quality of clustering. Comparatively, the Isolation Forest approach exhibits a decrease in the average Silhouette score when compared to Scenario One, indicating a minor decrease in the quality of clustering The Local Outlier Factor approach regularly achieves a high average Silhouette score, demonstrating its robust clustering ability in different scenarios.

Figure 5.7, which represents scenario Three, demonstrates that the Elliptic Envelope approach exhibits another decrease in the average Silhouette score. This indicates that the quality of clustering continues to decrease in this situation. In contrast to Scenarios One and Two, the Isolation Forest approach exhibits a decrease in the average Silhouette score, suggesting that it is less effective at producing distinct and cohesive clusters in this particular scenario. Although the Local Outlier Factor method continues to perform well, there is a little decrease in the average Silhouette score compared to earlier instances. Nevertheless, it remains an excellent way for clustering.

Upon comparing the Silhouette analysis over the three scenarios, it is evident that the Isolation Forest method, although initially effective in Scenario One, experiences a decrease in average Silhouette scores by Scenario Three. This implies that the effectiveness of Isolation Forest is contingent upon the specific qualities of the data, and it is not consistently superior in all scenarios. The Local Outlier Factor approach consistently performs well, maintaining high Silhouette scores in all scenarios, which makes it a reliable choice for clustering. The Elliptic Envelope approach, although successful, has more variability in performance across different scenarios, suggesting that it may be more sensitive to the underlying data characteristics.

In summary, the Isolation Forest method proves to be a robust solution for detecting outliers in Scenario One, especially when combined with K-Means clustering. Nevertheless, its efficacy decreases in subsequent scenarios, indicating that although it is efficient in certain scenarios, its

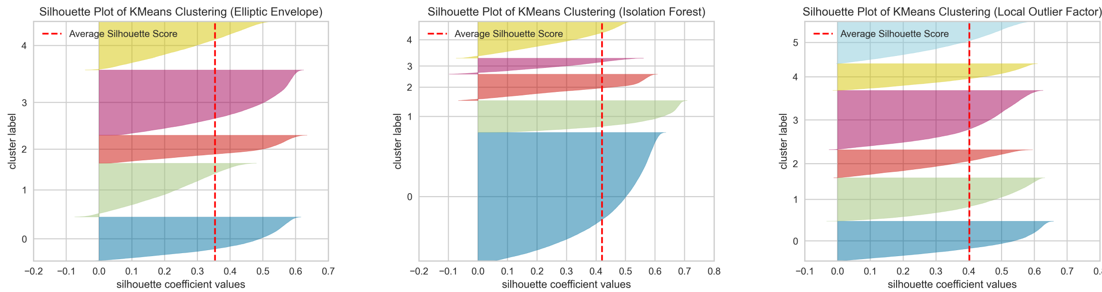


Figure 5.5: Silhouette Analysis of K-Means Clustering with Different Outlier Detection Methods for Scenario One(window size=1 hour and step size=30 minutes)

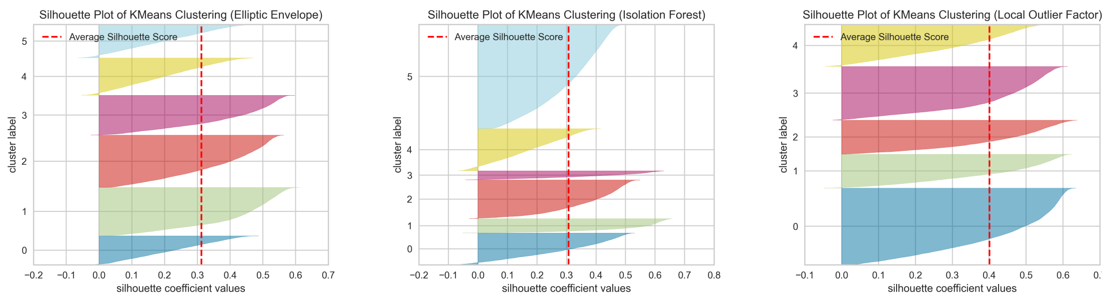


Figure 5.6: Silhouette Analysis of K-Means Clustering with Different Outlier Detection Methods for Scenario Two(window size=2 hours and step size=1 hour

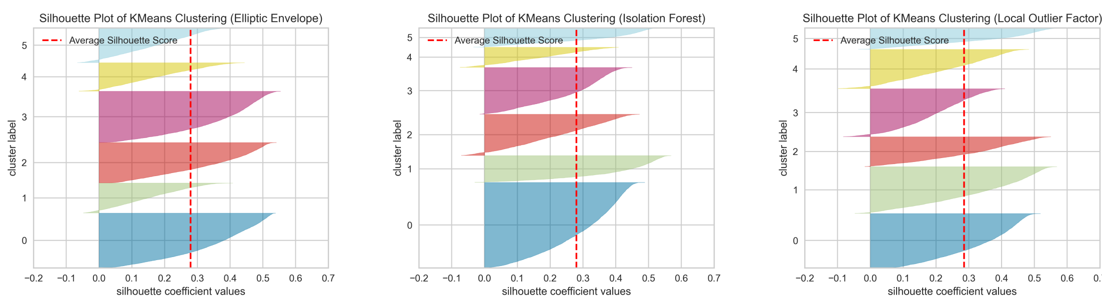


Figure 5.7: Silhouette Analysis of K-Means Clustering with Different Outlier Detection Methods for Scenario Three(window size=3 hour and step size=1.5 hours

usefulness may fluctuate depending on the characteristics of the data. The Local Outlier Factor approach consistently demonstrates superior performance in all scenarios, making it a reliable choice for clustering in many contexts.

5.4.8 Discussion

Table 3 presents a comparison of the performance of two clustering algorithms, namely K-Means and Agglomerative, in Scenario 1. The comparison is performed under three different outlier detection methods: Elliptic Envelope, Isolation Forest, and Local Outlier Factor. The evaluation relies on three metrics: Silhouette Coefficient, Calinski-Harabasz Index, and Davies-Bouldin Index. Table 3 demonstrates that the Isolation Forest method consistently produces highest values for both the Silhouette Coefficient and the Calinski-Harabasz Index. This indicates that it effectively detects outliers and enhances the quality of clusters for both K-Means and Agglomerative clustering algorithms. Moreover, the Elliptic Envelope and Local Outlier Factor exhibit diverse outcomes, which are dependent upon the clustering algorithm's effectiveness. K-Means consistently surpasses Agglomerative clustering in terms of Silhouette Coefficient ,Calinski-Harabasz and Davies-Bouldin Index for all outlier detection methods. This suggests that K-Means algorithm has a tendency to generate more compact and well-separated clusters in the Scenario 1.

Table 4 shows a comparison of the performance of two clustering algorithms in Scenario 2. Table 4 demonstrates that the Local Outlier Factor method consistently produces highest values for all the Silhouette Coefficient and the Calinski-Harabasz Index. This indicates that it effectively detects outliers and enhances the quality of clusters for both K-Means and Agglomerative clustering algorithms. Moreover, Isolation Forest method produces higher values for all the Silhouette Coefficient and the Calinski-Harabasz Index than the Elliptic Envelope method . K-Means consistently surpasses Agglomerative clustering in terms of Silhouette Coefficient , Calinski-Harabasz and Davies-Bouldin Index for both outlier detection methods (Elliptic Envelope, Local Outlier Factor). This suggests that K-Means algorithm has a tendency to generate more compact and well-separated clusters in the Scenario 2.

Table 5 presents a comparison of the performance of two clustering algorithms in Scenario 3. Table 5 present that the Local Outlier Factor consistently produces the highest values for the Silhouette Coefficient and Calinski-Harabasz Index with K-Means clustering method. On the other hand, with Agglomerative clustering method , the solation Forest strategy, yields highest values for these metrics.

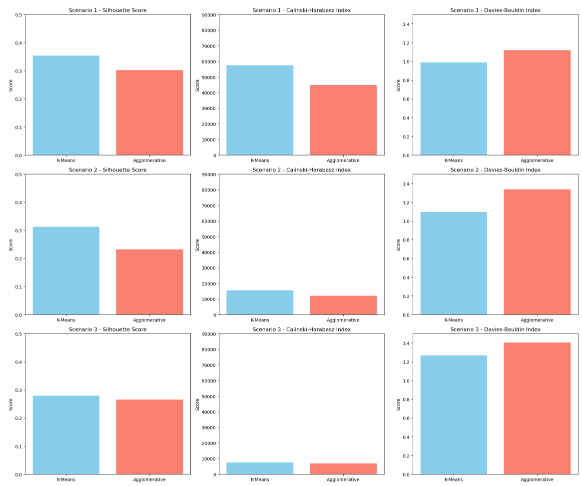


Figure 5.8: Comparative Analysis of Clustering Performance Across Three Scenarios

Across all evaluation criteria (Silhouette Coefficient, Calinski-Harabasz Index, and Davies-Bouldin Index), K-Means consistently performs better than Agglomerative clustering for both Elliptic Envelope and Local Outlier Factor outlier identification approaches. These findings indicate that K-Means is more effective at producing compact and well-separated clusters within the context of Scenario 3.

As shown in Figure 5.8 the comparative analysis across three scenarios, show that the K-Means clustering algorithm constantly surpasses Agglomerative clusteringparticularly when paired with the Isolation Forest method in Scenario One. This combination proves to be the most effective for producing well-defined clusters that are both compact and well-separated, making it the recommended approach for clustering the analyzed data, as indicated by the Silhouette Coefficient, Calinski-Harabasz , and Davies-Bouldin Index. In summary, the combination of K-Means and Local Outlier Factor yields the most promising results for clustering the analyzed data.

5.4.9 Cluster Characterization and Interpretation

Characterizing clusters in the context of traffic anomalies is essential for several reasons. Firstly, it offers a more profound comprehension of the inherent characteristics and regularities of traffic anomalies, which have the potential to cause severe disruptions in transportation systems. Traffic anomalies, which refer to deviations from normal traffic patterns, can lead to congestion, delays, and potentially accidents. Through the examination of these anomalies, authorities responsible for

Table 5.2: Comparison of Clustering Methods for Scenario one.

(a) Outliers Method: Elliptic Envelope

	KMeans	Agglomerative
Silhouette	0.354000	0.301600
Calinski-Harabasz	57347.150700	44815.396500
Davies-Bouldin	0.987100	1.118100

(b) Outliers Method: Isolation Forest

	KMeans	Agglomerative
Silhouette	0.419700	0.365600
Calinski-Harabasz	83620.841100	72275.403300
Davies-Bouldin	0.888300	1.003500

(c) Outliers Method: Local Outlier Factor

	KMeans	Agglomerative
Silhouette	0.401300	0.334600
Calinski-Harasz	43948.212900	37548.952300
Davies-Bouldin	0.827900	0.908700

Table 5.3: Comparison of Clustering Methods for Scenario Two.

(a) Outliers Method: Elliptic Envelope

	KMeans	Agglomerative
Silhouette	0.312000	0.231700
Calinski-Harabasz	15536.666700	12011.187500
Davies-Bouldin	1.094300	1.336500

(b) Outliers Method: Isolation Forest

	KMeans	Agglomerative
Silhouette	0.306400	0.386800
Calinski-Harabasz	17311.568000	16720.658400
Davies-Bouldin	1.044800	0.845100

(c) Outliers Method: Local Outlier Factor

	KMeans	Agglomerative
Silhouette	0.400400	0.363100
Calinski-Harabasz	20162.350900	17769.464800
Davies-Bouldin	0.855800	0.902500

Table 5.4: Comparison of Clustering Methods for Scenario Three.

(a) Outliers Method: Elliptic Envelope

	KMeans	Agglomerative
Silhouette	0.279200	0.265300
Calinski-Harabasz	7488.032000	6837.001600
Davies-Bouldin	1.266300	1.404600

(b) Outliers Method: Isolation Forest

	KMeans	Agglomerative
Silhouette	0.279900	0.282900
Calinski-Harabasz	7857.835400	7775.797400
Davies-Bouldin	1.198500	1.138400

(c) Outliers Method: Local Outlier Factor

	KMeans	Agglomerative
Silhouette	0.285300	0.241100
Calinski-Harabasz	8724.303100	7256.889200
Davies-Bouldin	1.118500	1.235100

traffic management can devise more efficient measures to minimize their influence and improve the overall effectiveness and safety of transportation networks.

Cluster characterisation helps in the identification of particular locations (ARRONDISSEMENTS) that are prone to specific types of traffic anomalies, as well as the measurement of the frequency and severity of these anomalies. This information is crucial for urban planners and traffic management to make well-informed decisions on the allocation of resources, improvements to infrastructure, and planning for emergency response.

This study utilized a systematic approach to analyze and describe the clusters generated by applying clustering algorithms to the identified traffic anomalies. The techniques and criteria employed for characterization encompass:

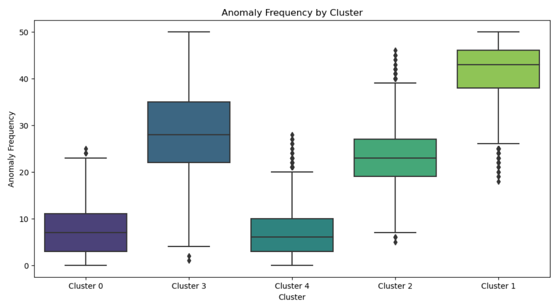
Anomaly Frequency Analysis: We performed an analysis of the frequency of anomalies within each cluster in order to gain insight into the rate at which these deviations occur. This helps in the identification of clusters exhibiting high anomaly rates, potentially indicating regions experiencing significant traffic issues.

Geographical Distribution: We analyzed the spatial distribution of traffic abnormalities by mapping the clusters geographically. This representation facilitates the identification of patterns related to certain regions and better comprehension of the geographical distribution of traffic problems.

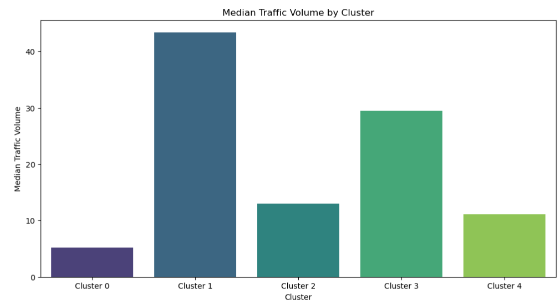
ARRONDISSEMENT Distribution: We analyzed the distribution of clusters across several ARRONDISSEMENTS to determine the administrative regions that are most impacted by traffic anomalies. This analysis is essential for the management and planning of traffic in the region.

Traffic Volume Analysis: We analyzed the median and standard deviation of traffic volumes within each cluster to get insight into the average traffic conditions and their variability. This helps in differentiating between clusters exhibiting significant fluctuations in traffic and those displaying more consistent traffic patterns. Figure 5.9a plot provides a comprehensive overview of the distribution of anomaly frequencies among various clusters. Clusters 1 and 3 show the highest median frequencies of anomalies, suggesting that intersections inside these clusters encounter traffic abnormalities most frequently. These clusters also exhibit the widest ranges and highest number of outliers, indicating substantial heterogeneity in the occurrence of anomalies within these clusters.

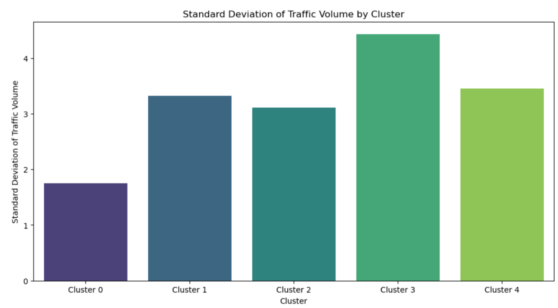
Clusters 0, 2, and 4 have lowered median frequencies, with Cluster 0 displaying the lowest median and the narrowest range, indicating a higher level of consistency and lower frequency of anomalies. The existence of outliers inside each cluster indicates that although the majority of intersections conform to the typical pattern of the cluster, there are specific places that exhibit abnormally high or low frequencies of anomalies. These points may require more investigation.



(a) Anomaly Frequency for each Cluster



(b) Median Traffic Volume by Cluster



(c) Standard Deviation of Traffic Volume by Cluster

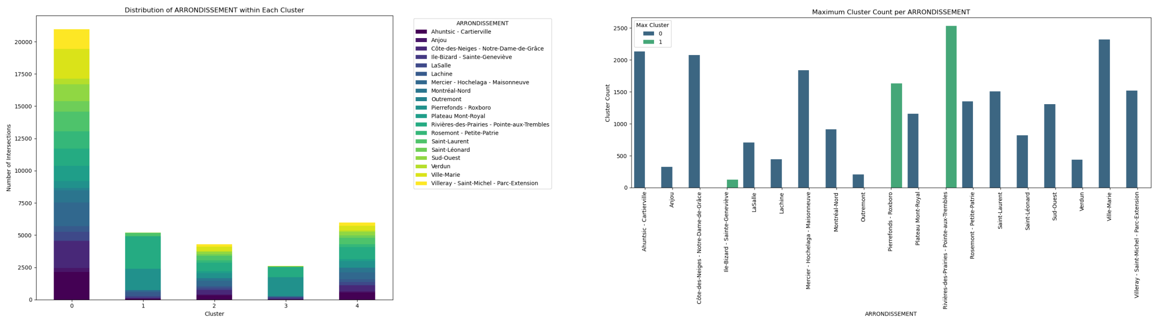
Understanding these distributions helps in identifying critical areas that require attention for traffic management and planning. High-frequency anomaly clusters might indicate problematic regions needing targeted interventions to improve traffic flow and safety.

Figure 5.9b shows significant differences in median traffic volumes among five clusters. Cluster 1 demonstrates the most highest median traffic volume, approximately 45, indicating regions with congested traffic movement, most likely urban areas or significant roadways. Clusters 2 and 3 have median volumes ranging from 15 to 30, indicating the presence of mixed-use or secondary roadways. Clusters 0 and 4 have the lowest median traffic levels, just above 5 and 10, indicating quieter places that may be residential or suburban.

Figure 5.9c Shows the variation in traffic volumes among five Clusters, highlighting significant differences in traffic consistency. Cluster 0 exhibits the smallest standard deviation, indicating a high level of consistency and predictability in traffic patterns. This is typically observed in residential or low-traffic regions. Clusters 1, 2, and 4 demonstrate a moderate level of variability, indicating that these locations include a combination of residential and commercial uses and experience moderate fluctuations in traffic. Cluster 3 exhibits the highest standard deviation, indicating notable variations in traffic volume, most likely occurring in prominent roadways or commercial regions with diverse traffic patterns.

Figure 5.10a shows the number of intersections in each cluster, organized by ARRONDISSEMENT. Cluster 0 comprises more than 20,000 intersections and encompasses many ARRONDISSEMENTS such as Villeray - Saint-Michel - Parc-Extension, Ville-Marie, and Saint-Laurent. This suggests a combination of residential and less densely populated urban regions. Cluster 1, consisting of almost 5,000 intersections, is primarily characterized by regions such as Rivières-des-Prairies - Pointe-aux-Trembles, indicating locations with significant traffic flow and commercial presence. Cluster 2, with around 3,000 intersections, and Cluster 4, consisting of roughly 7,500 intersections, are characterized as mixed-use regions with moderate levels of traffic. Cluster 3, characterized by a relatively low number of intersections (about 2,500), mostly encompasses Montréal-Nord and Saint-Léonard, suggesting locations with significant fluctuations in traffic volume. The Figure 5.10a provides an overview of the varied representation and traffic characteristics in each cluster. Cluster 0 demonstrates consistent conditions, Cluster 1 indicates significant traffic volumes, Clusters 2 and 4 represent areas with a balanced mix of uses, and Cluster 3 necessitates focused traffic management due to its high variability.The spatial distribution of these clusters is further illustrated in the Figure5.11 where it illustrates the spatial distribution of several traffic clusters inside a specific geographic region. Each colored dot represents a distinct cluster type, indicating the locations where particular traffic patterns or abnormalities are more common. The clustering analysis reveals distinct traffic patterns in different locations of the city, where certain places are predominantly characterized by specific clusters.

Figure 5.10b depicts the type of traffic cluster that occurs in each ARRONDISSEMENT. Figure illustrates the relative occurrence of two distinct clusters (presumably denoted as Max Cluster 0 and Max Cluster 1) throughout different ARRONDISSEMENTS, indicating the dominant cluster in each location. Based on this Figure, it is evident that certain arrondissements constantly exhibit a dominant cluster type. For instance, the locations of "Ahuntsic - Cartierville," "Côte-des-Neiges - Notre-Dame-de-Grâce," and "Ville-Marie" have a notably greater occurrence of one cluster type, suggesting that Cluster 0 is the most prevalent in these regions. Conversely, the arrondissements "Île-Bizard - Sainte-Geneviève" and "Pierrefonds - Roxboro" exhibit a greater number of



(a) Distribution of ARRONDISSEMENT within Each Cluster

(b) Maximum Cluster Count per ARRONDISSEMENT

Figure 5.10: Cluster distribution and counts in each ARRONDISSEMENT.

occurrences for Cluster 1. This statistic is essential for discerning the prevailing traffic patterns or anomalies inside each arrondissement, aiding in comprehending the particular traffic circumstances or problems that are most prevalent in distinct areas of the city. These insights can provide guidance for managing traffic and building infrastructure in a way that specifically addresses the most common or troublesome traffic patterns in each area.

In the conclusion, this research uncovers distinct traffic patterns and abnormal distributions among various clusters, offering crucial information for managing traffic and planning urban areas. Figure 5.9a shows that Cluster 1 has the highest incidence of traffic anomalies, indicating regions with frequent disruptions that require concentrated attention. Cluster 0, in comparison, demonstrates the lowest frequency of anomalies, suggesting a higher level of stability in traffic circumstances. The Figure5.9b presents further evidence by illustrating that Cluster 1 has the largest median traffic volume, suggesting the presence of vibrant urban hubs or significant transportation routes. In contrast, Cluster 0 exhibits the lowest median traffic volume, indicating calm residential areas.Cluster 0, however, exhibits negligible change, which further reinforces its categorization as a stable and low-traffic cluster. These findings underscore the need of adopting tailored traffic control measures. Clusters exhibiting elevated levels of anomalies and variability would derive advantages from using dynamic traffic control techniques, while clusters characterized by low traffic and variability can require routine monitoring to maintain their stability.

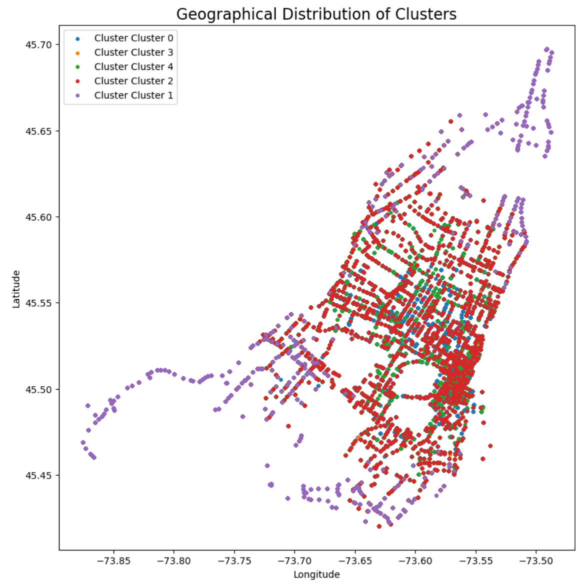


Figure 5.11: Geographical Distribution of Clusters

5.5 Summary

This study emphasizes the crucial importance of identifying and describing anomalies in traffic flow in order to improve the effectiveness and safety of transportation networks. By utilizing three different anomaly detection methods Elliptic Envelope, Isolation Forest, and Local Outlier Factor on pre-processed datasets with different window configurations, we were able to accurately detect traffic anomalies. By utilizing clustering algorithms, notably K-Means and Hierarchical Clustering, we were able to segment these anomalies into distinct groups and identify the most suitable number of clusters for detailed characterization.

The evaluation results clearly show that The Local Outlier Factor approach consistently produces the highest quality of clustering. This is evident from its superior average silhouette scores across different scenarios. the Isolation Forest also performs well, especially in maintaining clearly defined and evenly distributed clusters. Although the Elliptic Envelope exhibits a moderate level of clustering quality, its performance is less robust when compared to the other approaches.

The comparative analysis, shows that the K-Means clustering method consistently performs better than the Agglomerative clustering algorithm to generate clusters that are both compact and well-separated. This conclusion is verified by metrics such as the Silhouette Coefficient, Calinski-Harabasz Index, and Davies-Bouldin Index. Local Outlier Factor is the most successful among the anomaly detection methods, followed by Isolation Forest and Elliptic Envelope. However, the efficiency of these approaches may vary depending on the scenario.

In conclusion, combining K-Means clustering with Local Outlier Factor anomaly detection is the most promising method for efficiently handling and comprehending traffic anomalies. This combination not only helps to identify traffic disruptions but also offers vital information about where and when they occur. This information is important for urban planners and traffic management authorities. These findings emphasize the significance of customized traffic management solutions to tackle specific anomalies and enhance the overall operation of transportation systems.

Chapter 6

Conclusions and Future Works

In this thesis,we present a unified and comprehensive strategy for addressing urban transportation congestion. Chapter 3 presents a sophisticated system aimed at enhancing traffic light timings through the application of machine learning and metaheuristic techniques. The principal objective is to reduce vehicle wait times and mitigate congestion in urban environments. This method utilizes predictive algorithms developed from actual traffic data to anticipate congestion patterns. These forecasts are subsequently employed to dynamically modify traffic signal timings. Metaheuristic techniques, such the Enhanced Bat Algorithm (EBAT), which demonstrated superior performance, and Particle Swarm Optimization (PSO), are utilized to identify near-optimal solutions for traffic light regulation. The outcome is a more effective traffic management system capable of real-time responses to congestion, hence enhancing traffic flow and minimizing delays.

Chapter 4 elaborates on the core concepts established in Chapter 3 by examining the complexities of real-world traffic data, which frequently included anomalies such as accidents or sudden road closures. This chapter presents a multilevel learning approach that combines anomaly detection and ensemble learning to improve the accuracy of congestion prediction. The application of Isolation Forest for anomaly detection facilitates data cleansing by finding and eliminating unusual traffic patterns. Upon data cleansing, various prediction models are trained, and their outputs are combined through ensemble learning methods such as stacking, yielding the most robust results. This multilevel methodology yields more reliable and accurate congestion predictions, which can subsequently guide the traffic signal optimization procedure outlined in Chapter 3.

Chapter 5 concentrates on the comprehensive characterization of traffic anomalies identified in earlier chapters. This chapter employs clustering algorithms, specifically K-Means, which proved most effective, and Hierarchical Clustering, to classify and examine traffic abnormalities, building upon the prediction and optimization methodologies discussed in Chapters 3 and 4. The objective is to understand the spatial and temporal distribution of these abnormalities, yielding enhanced insights into their causes and impacts on traffic flow. By clustering the anomalies, the system can identify patterns and trends essential for sustainable traffic management and urban planning. This improved characterization helps traffic authorities in making more informed decisions about where and when interventions are needed.

The chapters are tightly interconnected, forming a unified structure for addressing urban traffic congestion. Chapter 3 establishes a framework for optimizing traffic light timings with predictive models. Chapter 4 enhances the accuracy of predictions using multilevel learning, enabling the traffic management system to effectively process complex, real-world data. The anomaly detection and data cleansing mechanisms in Chapter 4 directly inform the optimization techniques in Chapter 3, ensuring that the system operates on high-quality data. Chapter 5 enhances the preceding chapters by presenting a comprehensive analysis of traffic anomalies via clustering, which facilitates immediate congestion management and yields significant insights for future planning.

Together, these three chapters offer a comprehensive solution to urban traffic congestion The predicted accuracy achieved in Chapter 4 improves the efficacy of the optimization methods in Chapter 3, while the clustering and anomaly characterization in Chapter 5 provide a deeper understanding of traffic patterns. The integration of prediction, optimization, and characterization guarantees a scalable and efficient traffic management system capable of adapting to the dynamic characteristics of urban traffic, hence enhancing mobility and alleviating congestion.

For future work, several promising research directions can be considered to extend the contributions of thesis:

Integration of Advanced Machine Learning Models and Real-Time Data: To further enhance traffic anomaly detection and prediction accuracy, future research could explore integrating more sophisticated machine learning models that can capture complex spatial and temporal patterns in traffic data. Additionally, incorporating real-time data from emerging technologies, such as IoT devices and connected vehicle sensors, could provide more granular and up-to-date information for predictions. This integration would allow the system to adapt more dynamically to changing traffic conditions and better anticipate anomalies in real time, improving the overall accuracy and responsiveness of the proposed framework.

Scalability and Generalization Across Different Urban Environments: Scalability and Generalization Across Different Urban Environments: While this study focuses on the Montreal region, future work could explore the scalability and generalization of the proposed methodologies to different cities with varying traffic conditions and infrastructures. This would involve testing the optimization and multilevel learning frameworks on datasets from other urban environments. Comparative analyses across diverse urban settings would provide insights into the robustness and adaptability of the methods.Additionally, factoring in long-term urban growth and changes in transportation infrastructure could improve the models' ability to predict traffic flow and anomalies over extended periods, aiding urban planners in different contexts.

Incorporation of Transportation Engineering Principles: Our future work expands the framework to include transportation engineering viewpoints. This could entail simulating the effects of traffic management interventions such signal timing changes, road extensions, and lane reconfigurations on congestion patterns. Incorporating traffic flow theories, queuing models, and transportation network optimization approaches might help researchers gain a better understanding of how proposed strategies interact with actual traffic systems. Collaborations with transportation experts could help enhance the model to better reflect the physical and operational characteristics of urban traffic networks. This addition would increase the framework's practical applicability and help to generate concrete solutions for traffic management and urban planning.

Appendix A: Dataset for Traffic Light Congestion Analysis

This appendix references a detailed dataset containing information about traffic lights and congestion metrics across various intersections in Montreal. The dataset includes:

- **INT_NO**: Intersection Number.
- **RUE_1, RUE_2**: Streets intersecting at the traffic light.
- **ARRONDISSEMENT**: The arrondissement of the traffic light.
- **Longitude, Latitude**: Location coordinates.

For more details about the dataset, please visit the following link:
`https://donnees.montreal.ca/dataset/feux-tous`

Sample of the Dataset (First 19 Rows)

INT_NO	RUE_1	RUE_2	ARRONDISSEMENT	Longitude	Latitude
97	Décarie	Ferrier Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6582	45.4972
98	Décarie	Isabella Est	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6322	45.4853
99	Décarie	Isabella Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6325	45.4849
100	Décarie	Jean-Talon Est	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6528	45.4946
101	Décarie	Jean-Talon Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6534	45.4942
102	Décarie	Snowdon Est	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6266	45.4827
103	Décarie	Snowdon Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6270	45.4825
104	Décarie	Paré Est	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6554	45.4961
105	Décarie	Paré Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6559	45.4957
106	Décarie	Plamondon Est	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6452	45.4911
107	Décarie	Plamondon Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6455	45.4907
108	Décarie	Queen-Mary Est	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6290	45.4838
109	Décarie	Queen-Mary Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6292	45.4835
110	Décarie	Royalmount Est	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6601	45.4991
111	Décarie	Royalmount Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6607	45.4988
112	Décarie	Van Horne Est	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6421	45.4897
113	Décarie	Van Horne Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6424	45.4893
114	Décarie	Vézina Est	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6476	45.4921
115	Décarie	Vézina Ouest	CÃ´te-des-Neiges - Notre-Dame-de-GrÃ¢ce	-73.6481	45.4918

Bibliography

[1] M. Daraghmeh, A. Awasthi, and A. Agarwal, *Multilevel learning for enhanced traffic congestion prediction using anomaly detection and ensemble learning.* Jun. 2024. DOI: 10.21203/rs.3.rs-4660420/v1.

[2] X.-Y. Liu, M. Zhu, S. Borst, and A. Walid, *Deep reinforcement learning for traffic light control in intelligent transportation systems*, Feb. 2023. DOI: 10.48550/arXiv.2302.03669.

[3] R. Suryawanshi, O. Rachalwar, A. Hedaoo, A. Thakur, and T. Rane, *A review of traffic light control system with reinforcement learning*, Jan. 2023.

[4] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graphs," in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Sep. 2008, pp. 656–671, ISBN: 978-3-540-87478-2. DOI: 10.1007/978-3-540-87479-9_61.

[5] E. van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *Proceedings of the Deep Reinforcement Learning Symposium at the 30th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:267818750.

[6] M. Wiering, "Multi-agent reinforcement learning for traffic light control.," in *Proceedings of the 17th International Conference on Machine Learning (ICML)*, Jan. 2000, pp. 1151–1158.

[7] Shen, C. Zhan, W.-n. Chen, Y.-J. Gong, J. Zhang, and Y. Li, "Bi-velocity discrete particle swarm optimization and its application to multicast routing problem in communication networks," *IEEE Transactions on Industrial Electronics*, vol. 61, Mar. 2014. DOI: 10.1109/TIE.2014.2314075.

[8] N. Taherkhani and S. Pierre, "Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 1–11, Apr. 2016. DOI: 10.1109/TITS.2016.2546555.

[9] V. Kumar and S. Kumar, "Energy balanced position-based routing for lifetime maximization of wireless sensor networks," *Ad Hoc Networks*, vol. 52, Aug. 2016. DOI: 10.1016/j.adhoc.2016.08.006.

[10] H. Wang, X. Meng, S. Li, and H. Xu, "A tree-based particle swarm optimization for multicast routing," *Computer Networks*, vol. 54, no. 15, pp. 2775–2786, 2010, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2010.05.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128610001441.

[11] A. Haghighat, K. Faez, A. Mowlaei, and Y. Ghahremani, "Ga-based heuristic algorithms for qos based multicast routing.," *Knowl.-Based Syst.*, vol. 16, pp. 305–312, Jan. 2003.

[12] M. Yazdani, M. Sarvi, S. Asadi Bagloee, N. Nassir, J. Price, and H. Parineh, "Intelligent vehicle pedestrian light (ivpl): A deep reinforcement learning approach for traffic signal control," *Transportation Research Part C: Emerging Technologies*, vol. 149, p. 103 991, 2023, ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2022.103991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X22004041.

[13] B. Liu and Z. Ding, "A distributed deep reinforcement learning method for traffic light control," *Neurocomputing*, vol. 490, pp. 390–399, 2022, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.11.106. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092523122101818X.

[14] N. Kumar, S. Rahman, and N. Dhakad, "Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–10, Apr. 2020. DOI: 10.1109/TITS.2020.2984033.

[15] N. Kumar and S. Rahman, "Deep reinforcement learning with vehicle heterogeneity based traffic light control for intelligent transportation system," in *Proceedings of the 2019 IEEE International Conference on Industrial Internet (ICII)*, Nov. 2019, pp. 28–33. DOI: 10.1109/ICII.2019.00016.

[16] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and M. Royle, "The scoot on-line traffic signal optimisation technique," *Traffic engineering and control*, vol. 23, 1982. [Online]. Available: https://api.semanticscholar.org/CorpusID:109965942.

[17] N. Gartner, "Opac: A demand-responsive strategy for traffic signal control," *Transportation Research Record Journal of the Transportation Research Board*, vol. No. 906, pp. 75–81, Jan. 1983.

[18] J. Henry, J.-L. Farges, and J. Tuffal, "The prodyn real time traffic algorithm," in *Traffic Flow Theory and Characteristics*, Elsevier, Aug. 2013, ISBN: 9780080293653. DOI: 10.1016/B978-0-08-029365-3.50048-1.

[19] F. Boillot, J.-M. Blosseville, J. B. Lesort, V. Motyka, M. Papageorgiou, and S. Sellam, "Optimal signal control of urban traffic networks," in *Proceedings of the 11th International Symposium on Transportation and Traffic Theory (ISTTT)*, 1992. [Online]. Available: https://api.semanticscholar.org/CorpusID:106620806.

[20] S. Sen and L. Head, "Controlled optimization of phases at an intersection," *Transportation Science*, vol. 31, pp. 5–17, Feb. 1997. DOI: 10.1287/trsc.31.1.5.

[21] J. Sánchez, M. J. Galán, and E. Rubio, "Genetic algorithms and cellular automata for traffic light cycles optimization. scalability study," in *Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS)*. [Online]. Available: https://api.semanticscholar.org/CorpusID:16733587.

[22] J. García-Nieto, E. Alba, and A. Olivera, "Swarm intelligence for traffic light scheduling: Application to real urban areas," *Eng. Appl. of AI*, vol. 25, pp. 274–283, Mar. 2012. DOI: 10.1016/j.engappai.2011.04.011.

[23] J. García-Nieto, A. Olivera, and E. Alba, "Optimal cycle program of traffic lights with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, Apr. 2013. DOI: 10.1109/TEVC.2013.2260755.

[24] S. Goettlich, M. Herty, and U. Ziegler, "Modeling and optimizing traffic light settings in road networks," *Computers & Operations Research*, vol. 55, Oct. 2014. DOI: 10.1016/j.cor.2014.10.001.

[25] J. Cheng, W. Wu, J. Cao, and K. Li, "Fuzzy group based intersection control via vehicular networks for smart transportations," *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 1–1, Jan. 2016. DOI: 10.1109/TII.2016.2590302.

[26] A. Jamal, M. T. Rahman, A. Hasan, I. Ullah, and M. Zahid Khattak, "Intelligent intersection control for delay optimization: Using meta-heuristic search algorithms," *Sustainability*, vol. 12, pp. 1–25, Mar. 2020. DOI: 10.3390/su12051896.

[27] A. Shahkar, S. Oruc, and A. Yelghi, *Traffic signal prediction based on anfis and meta-heuristic algorithms applied to a vissim-based simulated intersection*, Jun. 2023. DOI: 10.21203/rs.3.rs-3057609/v1.

[28] C. Shirke, N. Sabar, E. Chung, and A. Bhaskar, "Metaheuristic approach for designing robust traffic signal timings to effectively serve varying traffic demand," English, *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 2021, Publisher Copyright: © 2021 Taylor & Francis Group, LLC., ISSN: 1547-2450. DOI: 10.1080/15472450.2021.1878892.

[29] M. Al-Turki, A. Jamal, H. M. Al-Ahmadi, M. A. Al-Sughaiyer, and M. Zahid, "On the potential impacts of smart traffic control for delay, fuel energy consumption, and emissions: An nsga-ii-based optimization case study from dhahran, saudi arabia," *Sustainability*, vol. 12, no. 18, p. 7394, 2020. DOI: 10.3390/su12187394. [Online]. Available: https://doi.org/10.3390/su12187394.

[30] L. Szatmari, L. Prodan, A. Iovanovici, and D. Avramoni, "Road intersection optimization with resource-constrained metaheuristic: A case study," in *Proceedings of the 2022 IEEE 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, Sep. 2022, pp. 000 261–000 266. DOI: 10.1109/SISY56759.2022.10036312.

[31] A. Jamal, A. Hasan, M. Farhan, M. Almoshaogeh, S. Ali, and M. Iqbal, "Metaheuristics for traffic control and optimization: Current challenges and prospects," in *Metaheuristics - Theories and Applications*. IntechOpen, Sep. 2021, ISBN: 978-1-83969-086-0. DOI: 10.5772/intechopen.99395.

[32] S. Gupta, Y. Zhang, and R. Su, "Urban traffic light scheduling for pedestrian–vehicle mixed-flow networks using discrete sine–cosine algorithm and its variants," *Applied Soft Computing*, vol. 120, p. 108 656, 2022, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2022.108656. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494622001351.

[33] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International journal of forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[34] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*, IEEE, 2015, pp. 153–158.

[35] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "Lstm network: A deep learning approach for short-term traffic forecast," *IET intelligent transport systems*, vol. 11, no. 2, pp. 68–75, 2017.

[36] A. Belhadi, Y. Djenouri, D. Djenouri, and J. C.-W. Lin, "A recurrent neural network for urban long-term traffic flow forecasting," *Applied Intelligence*, vol. 50, pp. 3252–3265, 2020.

[37] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, 2017.

[38] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[39] A. Nejadettehad, H. Mahini, and B. Bahrak, "Short-term demand forecasting for online car-hailing services using recurrent neural networks," *Applied Artificial Intelligence*, vol. 34, no. 9, pp. 674–689, 2020.

[40] A. Azhar, S. Rubab, M. M. Khan, *et al.*, "Detection and prediction of traffic accidents using deep learning techniques," *Cluster Computing*, vol. 26, pp. 477–493, 1 2022. DOI: 10.1007/s10586-021-03502-1.

[41] D. K. Jain, X. Zhao, S. Garcia, and S. Neelakandan, "Robust multi-modal pedestrian detection using deep convolutional neural network with ensemble learning model," *Expert Systems with Applications*, vol. 249, p. 123 527, 2024, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2024.123527. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424003920.

[42] K. Tang, T. Guo, F. Shao, Y. Ma, and A. J. Khattak, "Spatial-temporal traffic performance collaborative forecast in urban road network based on dynamic factor model," *Expert Systems with Applications*, vol. 225, p. 120 090, 2023, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.120090. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423005924.

[43] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic, "Detecting and quantifying causal associations in large nonlinear time series datasets," *Science advances*, vol. 5, no. 11, eaau4996, 2019.

[44] C.-I. Chang, Y. Wang, and S.-Y. Chen, "Anomaly detection using causal sliding windows," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 7, pp. 3260–3270, 2015.

[45] G.-L. Huang, K. Deng, Y. Ren, and J. Li, "Root cause analysis of traffic anomalies using uneven diffusion model," *IEEE Access*, vol. 7, pp. 16 206–16 216, 2019.

[46] Q. Cheng, G. Dai, B. Ru, *et al.*, "Traffic flow outlier detection for smart mobility using gaussian process regression assisted stochastic differential equations," *Transportation Research Part E: Logistics and Transportation Review*, vol. 193, p. 103 840, 2025, ISSN: 1366-5545. DOI: https://doi.org/10.1016/j.tre.2024.103840. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554524004319.

[47] Z. Peng, "Multi-level spatial-temporal fusion neural network for traffic flow prediction," *Cluster Computing*, 2024. DOI: 10.1007/s10586-024-04296-8.

[48] L. Zhao, Y. Song, C. Zhang, *et al.*, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[49] J. Xu, "A coupled generative graph convolution network by capturing dynamic relationship of regional flow for traffic prediction," *Cluster Computing*, 2024. DOI: 10.1007/s10586-024-04323-8.

[50] W. Jiang, Y. Xiao, Y. Liu, Q. Liu, Z. Li, *et al.*, "Bi-grcn: A spatio-temporal traffic flow prediction model based on graph neural network," *Journal of Advanced Transportation*, vol. 2022, 2022.

[51] X. Peng, Y. Lin, Q. Cao, Y. Cen, H. Zhuang, and Z. Lin, "Traffic anomaly detection in intelligent transport applications with time series data using informer," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2022, pp. 3309–3314.

[52] N. Wang and X. Zhao, "Time-series prediction based on double pyramid bidirectional feature fusion mechanism," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 106, no. 6, pp. 886–895, 2023.

[53] Y. Djenouri, A. Belhadi, J. C.-W. Lin, and A. Cano, "Adapted k-nearest neighbors for detecting anomalies on spatio–temporal traffic flow," *Ieee Access*, vol. 7, pp. 10 015–10 027, 2019.

[54] X. Chen, J. Lu, J. Zhao, Z. Qu, Y. Yang, and J. Xian, "Traffic flow prediction at varied time scales via ensemble empirical mode decomposition and artificial neural network," *Sustainability*, vol. 12, no. 9, p. 3678, 2020.

[55] H. Zhan, G. Gomes, X. S. Li, K. Madduri, A. Sim, and K. Wu, "Consensus ensemble system for traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3903–3914, 2018.

[56] A. Gregory and C. J. Cotter, "A seamless multilevel ensemble transform particle filter," *SIAM Journal on Scientific Computing*, vol. 39, no. 6, A2684–A2701, 2017.

[57] M. Siino, F. J. Rodríguez-Cortés, J. Mateu, and G. Adelfio, "Spatio-temporal classification in point patterns under the presence of clutter," *Environmetrics*, vol. 31, no. 2, e2599, 2020.

[58] A. A. Awan, A. Majid, R. Riaz, S. S. Rizvi, and S. J. Kwon, "A novel deep stacking-based ensemble approach for short-term traffic speed prediction," *IEEE Access*, 2024.

[59] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, pp. 241–258, 2020.

[60] Y. Huo, Y. Cao, Z. Wang, Y. Yan, Z. Ge, and Y. Yang, "Traffic anomaly detection method based on improved gru and efms-kmeans clustering," *Computer Modeling in Engineering & Sciences*, vol. 126, pp. 1053–1091, Jan. 2021. DOI: `10.32604/cmes.2021.013045`.

[61] A. Daranda and G. Dzemyda, "Navigation decision support: Discover of vessel traffic anomaly according to the historic marine data," *International Journal of Computers Communications & Control*, vol. 15, Apr. 2020. DOI: `10.15837/ijccc.2020.3.3864`.

[62] G. Shu Fuhnwi, J. Agbaje, K. Oshinubi, and O. Peter, "An empirical study on anomaly detection using density-based and representative-based clustering algorithms," *Journal of Nigerian Society of Physical Sciences*, vol. 5, p. 1364, Apr. 2023. DOI: `10.46481/jnsps.2023.1364`.

[63] Z. Ren, X. Li, J. Peng, *et al.*, "Graph autoencoder with mirror temporal convolutional networks for traffic anomaly detection," *Scientific Reports*, vol. 14, 1 2024. DOI: `10.1038/s41598-024-51374-3`.

[64] H. Zhang, S. Zhao, R. Liu, W. Wang, and R. Hu, "Automatic traffic anomaly detection on the road network with spatial-temporal graph neural network representation learning," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–12, Jun. 2022. DOI: 10.1155/2022/4222827.

[65] Y. Huang, H. Zhen, and J. J. Yang, "Cluster-guided denoising graph auto-encoder for enhanced traffic data imputation and fault detection," *Expert Systems with Applications*, vol. 261, p. 125 531, 2025, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2024.125531. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424023984.

[66] Z. Zhang, F. He, Y. Wang, and M. Li, "High-performance traffic speed forecasting based on spatiotemporal clustering of road segments," *IET Intelligent Transport Systems*, vol. 15, Dec. 2020. DOI: 10.1049/itr2.12016.

[67] Z. Sun, Y. Hu, W. Li, S. Feng, and L. Pei, "Prediction model for short-term traffic flow based on a k-means-gated recurrent unit combination," *IET Intelligent Transport Systems*, vol. 16, Jan. 2022. DOI: 10.1049/itr2.12165.

[68] H. Shaji, A. Tangirala, and L. Vanajakshi, "Joint clustering and prediction approach for travel time prediction," *PLOS ONE*, vol. 17, e0275030, Sep. 2022. DOI: 10.1371/journal.pone.0275030.

[69] E. Vanem and A. Brandsater, "Cluster-based anomaly detection in condition monitoring of a marine engine system," Oct. 2018, pp. 20–31. DOI: 10.1109/PHM-Chongqing.2018.00011.

[70] Y. Wu, H. Dai, and H. Tang, "Graph neural networks for anomaly detection in industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, pp. 9214–9231, 12 2022. DOI: 10.1109/jiot.2021.3094295.

[71] Y. Feng, W. Cai, H. Yue, Y. Lin, J. Chen, and Z. Hu, "An improved x-means and isolation forest based methodology for network traffic anomaly detection," *Plos One*, vol. 17, e0263423, 1 2022. DOI: 10.1371/journal.pone.0263423.

[72] X. Wei, H. Huang, S. Tian, X. Yang, and B. Xu, "An online adaptive network anomaly detection model," May 2009, pp. 365–368. DOI: 10.1109/CSO.2009.97.

[73] Y. Yue, H. Hu, X. Wei, and C. Jiang-Hang, "Network traffic anomaly detection method based on a feature of catastrophe theory," *Chinese Physics Letters*, vol. 27, p. 060 501, Jun. 2010. DOI: 10.1088/0256-307X/27/6/060501.

[74] M. Balega, W. Farag, X. Wu, S. Ezekiel, and Z. Good, "Enhancing iot security: Optimizing anomaly detection through machine learning," *Electronics*, vol. 13, p. 2148, 11 2024. DOI: 10.3390/electronics13112148.

[75] T. J. Chengula, J. Mwakalonge, G. Comert, M. Sulle, S. Siuhi, and E. Osei, "Enhancing advanced driver assistance systems through explainable artificial intelligence for driver anomaly detection," *Machine Learning with Applications*, vol. 17, p. 100 580, 2024, ISSN: 2666-8270. DOI: https://doi.org/10.1016/j.mlwa.2024.100580. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827024000562.

[76] Y. Vanrompay, A.-U.-H. Yasar, D. Preuveneers, and Y. Berbers, "Context-aware optimized information dissemination in large scale vehicular networks," Jun. 2010. DOI: 10.1145/1839294.1839344.

[77] H. Gonzalez, J. Han, Y. Ouyang, and S. Seith, "Multidimensional data mining of traffic anomalies on large-scale road networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2215, pp. 75–84, Dec. 2011. DOI: 10.3141/2215-08.

[78] F. Guerrini, "Traffic congestion costs americans $ 124 billion a year, report says," *Forbes, October*, vol. 14, 2014.

[79] T. Economist, "The cost of traffic jams," *The Economist: London, UK*, 2014.

[80] M. K. Tan, H. S. E. Chuo, R. K. Y. Chin, K. B. Yeo, and K. T. K. Teo, "Optimization of urban traffic network signalization using genetic algorithm," in *2016 IEEE Conference on Open Systems (ICOS)*, IEEE, 2016, pp. 87–92.

[81] K. Gao, Y. Zhang, Y. Zhang, R. Su, and P. N. Suganthan, "Meta-heuristics for bi-objective urban traffic light scheduling problems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2618–2629, 2019. DOI: 10.1109/TITS.2018.2868728.

[82] A. J. Miller, "Settings for fixed-cycle traffic signals," *Journal of the Operational Research Society*, vol. 14, no. 4, pp. 373–386, 1963.

[83] F. V. Webster, "Traffic signal settings," Tech. Rep., 1958.

[84] S.-B. Cools, C. Gershenson, and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," *Advances in applied self-organizing systems*, pp. 45–55, 2013.

[85] I. Porche and S. Lafortune, "Adaptive look-ahead optimization of traffic signals," *Journal of Intelligent Transportation System*, vol. 4, no. 3-4, pp. 209–254, 1999.

[86] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/-CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[87] F.-Q. Lu, M. Huang, W.-K. Ching, and T. K. Siu, "Credit portfolio management using two-level particle swarm optimization," *Information Sciences*, vol. 237, pp. 162–175, 2013, Prediction, Control and Diagnosis using Advanced Neural Computations, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2013.03.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002002551300193X.

[88] T. Yan, F. Lu, S. Wang, L. Wang, and H. Bi, "A hybrid metaheuristic algorithm for the multi-objective location-routing problem in the early post-disaster stage," *Journal of Industrial and Management Optimization*, vol. 19, Jan. 2022. DOI: 10.3934/jimo.2022145.

[89] H. Wen, S. X. Wang, F. Q. Lu, *et al.*, "Colony search optimization algorithm using global optimization," *Journal of Supercomputing*, vol. 78, no. 5, pp. 6567–6611, Oct. 2021. DOI: 10.1007/s11227-021-04127-2. [Online]. Available: https://doi.org/10.1007/s11227-021-04127-2.

[90] M. Huang, L. Ren, L. H. Lee, and X. Wang, "4pl routing optimization under emergency conditions," *Know.-Based Syst.*, vol. 89, no. C, pp. 126–133, Nov. 2015, ISSN: 0950-7051.

DOI: 10.1016/j.knosys.2015.06.023. [Online]. Available: https://doi.org/10.1016/j.knosys.2015.06.023.

[91] C.-W. Tsai, T.-C. Teng, J.-T. Liao, and M.-C. Chiang, "An effective hybrid-heuristic algorithm for urban traffic light scheduling," *Neural Computing and Applications*, vol. 33, Dec. 2021. DOI: 10.1007/s00521-021-06341-8.

[92] I. Tunc and M. T. Soylemez, "Fuzzy logic and deep q learning based control for traffic lights," *Alexandria Engineering Journal*, vol. 67, pp. 343–359, 2023.

[93] M. B. Younes, A. Boukerche, and F. De Rango, "Smartlight: A smart efficient traffic light scheduling algorithm for green road intersections," *Ad Hoc Networks*, vol. 140, p. 103 061, 2023.

[94] D. Jutury, N. Kumar, A. Sachan, Y. Daultani, and N. Dhakad, "Adaptive neuro-fuzzy enabled multi-mode traffic light control system for urban transport network," *Applied Intelligence*, vol. 53, no. 6, pp. 7132–7153, 2023.

[95] M. A. Khasawneh and A. Awasthi, "Intelligent traffic light control," in *Proceedings of the 2nd African International Conference on Industrial Engineering and Operations Management (IEOM), Harare, Zimbabwe*, 2020, pp. 7–10.

[96] A. Botchkarev, "A new typology design of performance metrics to measure errors in machine learning regression algorithms," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 14, pp. 045–076, 2019.

[97] M. A. Khasawneh and A. Awasthi, "Intelligent meta-heuristic-based optimization of traffic light timing using artificial intelligence techniques," *Electronics*, vol. 12, no. 24, p. 4968, 2023.

[98] M. Tang, X. Fu, H. Wu, Q. Huang, and Q. Zhao, "Traffic flow anomaly detection based on robust ridge regression with particle swarm optimization algorithm," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, 2020.

[99] M. Daraghmeh, A. Agarwal, and Y. Jararweh, "Incorporating data preparation and clustering techniques for workload segmentation in large-scale cloud data centers," in *2023 Fourth International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, IEEE, 2023, pp. 7–14.

[100] M. Daraghmeh, A. Agarwal, and Y. Jararweh, "An ensemble clustering approach for modeling hidden categorization perspectives for cloud workloads," *Cluster Computing*, pp. 1–25, 2023.

[101] H. Robert *et al.*, *Time series analysis and its applications with r examples second edition*, 2006.

[102] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[103] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20 150 202, 2016.

[104] V. de Montréal, *Feux tous*, `https://donnees.montreal.ca/dataset/feux-tous`, Data set, n.d. [Online]. Available: `https://donnees.montreal.ca/dataset/feux-tous`.

[105] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[106] R. Ravish and S. R. Swamy, "Intelligent traffic management: A review of challenges, solutions, and future perspectives," *Transport and Telecommunication Journal*, vol. 22, no. 2, pp. 163–182, 2021.

[107] Y. Djenouri, A. Belhadi, J. C.-W. Lin, D. Djenouri, and A. Cano, "A survey on urban traffic anomalies detection algorithms," *IEEE Access*, vol. 7, pp. 12 192–12 205, 2019. DOI: `10.1109/ACCESS.2019.2893124`.

[108] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*, IEEE, 2008, pp. 413–422.

[109] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *Gi/itg workshop mmbnet*, vol. 7, 2007.

[110] M. Daraghmeh, A. Agarwal, and Y. Jararweh, "Cloud workload categorization using various data preprocessing and clustering techniques," in *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*, 2023, pp. 1–10.

[111] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987, ISSN: 0377-0427. DOI: https://doi.org/10.1016/0377-0427(87)90125-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042787901257.

[112] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[113] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979. DOI: 10.1109/TPAMI.1979.4766909.

[114] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.