Journal of Theoretical and Applied Information Technology
31st May 2025. Vol.103. No.10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ADVANCED RIDGE REGRESSION USING IMFT MODEL FOR RSU DESIGN IN VEHICULAR NETWORKS

KRISHNA KOMARAM[1], NAGARJUNA KARYEMSETTY[2]

[1]Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Vijayawada, Andhra Pradesh, India.
Email:krishnak2k13@gmail.com
[2] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Vijayawada, Andhra Pradesh, India.
Email: nagarjunak@kluniversity.in

ABSTRACT

In the realm of Vehicular Ad Hoc Networks (VANETs), Roadside Units (RSUs) play a pivotal role in enhancing communication, data processing, and predictive analytics. This paper introduces a novel hybrid design that integrates Ridge Regression and XG-Boost algorithms to optimize the data processing and prediction capabilities of RSUs, aimed at improving traffic management and safety applications. The hybrid framework with IMFT (inter-intra mod filter) algorithm employs Ridge Regression for robust initial data processing, minimizing overfitting and ensuring reliability in the noisy, dynamic environment of vehicular data. Feature extraction with IMFT is utilized to encompass the relevant features before utilizing Ridge-Model for high-accuracy predictions, leveraging its gradient boosting capabilities to facilitate timely interventions and optimize traffic flow. Furthermore, the architecture of the RSU is expanded to include essential units such as communication modules, data storage, and user interface components, all functioning cohesively to create a comprehensive system. With the proposed IMFT design we have incorporated extensive simulations with K-fold loss to demonstrate that the proposed IMFT with Ridge Model design significantly enhances prediction accuracy and processing efficiency compared to traditional methods (Elastic Net.) with more than 98% of improved R^2-score. By optimizing the operational capabilities of RSUs in VANETs, this work contributes to the development of smarter and safer urban mobility solutions, paving the way for more effective traffic management and improved vehicular safety.

Keywords: *Vehicular Ad Hoc Networks (VANETs), Roadside Units (RSUs), Security Protocols, Energy Management, ML (Machine Learning), Ridge Regression.*

1. INTRODUCTION

Vehicular Ad Hoc Networks (VANETs) represent a transformative approach to improving road safety and traffic management by facilitating communication among vehicles and infrastructure. At the heart of this system are Roadside Units (RSUs), which serve as critical nodes that enable data exchange between vehicles, traffic management systems, and various services. As urban environments grow increasingly complex, the demand for efficient communication and real-time data processing within VANETs intensifies. RSUs are uniquely positioned to bridge the gap between vehicles and central traffic management, offering opportunities for enhanced road safety, reduced congestion, and optimized traffic flow. Designing effective RSU systems involves the integration of advanced communication technologies and intelligent algorithms that can process large volumes of data in real time. Traditional approaches often fall short in handling the dynamic and heterogeneous nature of vehicular data. This necessitates innovative methods that leverage machine learning algorithms, enabling RSUs to perform predictive analytics and decision-making with high accuracy. The incorporation of such algorithms can significantly improve the responsiveness of traffic management systems, ensuring that interventions are timely and contextually relevant.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Currently in RSU's the use of machine learning algorithms has been proven more effective and intuitive models to improve traffic management and safety. In these approaches the need for ensemble and its criterion's are more effective and utilized with current algorithm's such as Random Forest (hyper-tuned), is a robust ensemble method that combines multiple decision trees, making it great for handling noisy, complex traffic data. By tuning parameters like the number of trees and depth, it helps RSUs predict traffic patterns, detect anomalies, and manage congestion. Ridge Regression (hyper-tuned), on the other hand, uses L2 regularization to prevent overfitting and is useful for predicting continuous variables like vehicle speed or traffic flow. It is effective in handling correlated features and ensuring accurate, stable predictions, though it assumes linear relationships. Similarly, Linear Regression (hyper-tuned) is simple and interpretable, helping RSUs forecast traffic density or vehicle speed, but it may struggle with complex, nonlinear data. Lasso Regression (hyper-tuned) focuses on selecting the most relevant features, which is crucial for efficient real-time decision-making in VANETs, although it can be unstable with highly correlated features. Elastic Net (hyper-tuned) combines the strengths of both Lasso and Ridge, making it ideal for situations with many correlated features, but it requires careful tuning of two parameters and can be computationally expensive.

Each algorithm brings unique strengths to RSU designs in VANETs. Random Forest is effective in noisy, unpredictable environments, while Ridge and Lasso are great for feature selection and avoiding overfitting. However, these algorithms have their limitations, such as assumptions of linearity or computational intensity. The hybrid approach, combining Ridge Regression for pre-processing and XGBoost for prediction, overcomes these gaps by first filtering out noise and irrelevant features before applying XG-Boost's powerful prediction capabilities. This combination provides more accurate, efficient traffic management, particularly in dynamic and complex environments like VANETs, where real-time predictions and decision-making are critical for improving road safety and optimizing traffic flow.

1.1 Problem Statement

In the context of Vehicular Ad Hoc Networks (VANETs), Roadside Units (RSUs) are crucial for facilitating communication between vehicles and traffic management systems to enhance road safety and optimize traffic flow. However, the dynamic, noisy, and complex nature of traffic data poses significant challenges for RSUs in making accurate, real-time predictions for traffic management and safety interventions. Traditional approaches to processing and predicting traffic patterns often fall short due to their inability to handle large volumes of data, complex interactions, and the heterogeneous nature of the information. For example, basic regression models may not effectively capture nonlinear relationships between various traffic factors, and models like Random Forest may become computationally expensive when dealing with large datasets. Additionally, issues like overfitting, data sparsity, and the need for real-time decision-making further complicate the development of efficient RSU systems.

To address these challenges, the proposed solution is a hybrid model that integrates Ridge Regression for robust data pre-processing and XGBoost for accurate predictions. The Ridge Regression component helps in reducing the complexity of the model and preventing overfitting, especially in noisy environments, while XGBoost's gradient boosting capabilities are leveraged for high-accuracy traffic predictions. This hybrid approach not only enhances the ability of RSUs to process and predict traffic conditions but also ensures timely and accurate interventions, contributing to safer and more efficient traffic management. The hybrid model overcomes the limitations of traditional machine learning algorithms by combining the strengths of Ridge Regression and XGBoost, addressing issues such as noise, feature selection, and the need for real-time decision-making in the dynamic context of VANETs.

1.2 Challenges

Implementing effective RSU systems within VANETs presents several key challenges. First, ensuring data privacy and security is paramount, as sensitive information is exchanged between vehicles and infrastructure. Second, communication latency can disrupt the timely exchange of critical data, undermining the responsiveness of traffic management interventions. Lastly, the integration of diverse data sources from various vehicles necessitates sophisticated algorithms capable of processing this information in real time, which can be a significant technical hurdle.

Even though these challenges effect the design constraints the proposed model tend to provide the resulting solution with integrated approach with two powerful machine learning algorithms: Ridge Regression and XGBoost to optimize the data processing and prediction capabilities of Roadside Units (RSUs) in Vehicular Ad Hoc Networks (VANETs). The reason for using these algorithms is to address the complex and dynamic nature of vehicular data, which can be noisy and heterogeneous.

1.3 Proposed Concept

The IMFT (Inter-Intra Model Filtering Technique) model, integrated with Ridge Regression, introduces a novel hybrid machine learning approach for optimizing Roadside Unit (RSU) performance in Vehicular Ad-hoc Networks (VANETs). Unlike conventional RSU deployment models that rely on heuristic or static optimization techniques, the IMFT model dynamically filters and processes traffic, energy, and memory data using a two-tier filtering mechanism. Initially, the model partitions data into intra-vehicle (D_intra) and extra-vehicle (D_extra) datasets, ensuring that localized vehicle patterns are analyzed separately from broader traffic influences. Ridge Regression is then applied to minimize noise, prevent overfitting, and improve generalization, allowing the model to accurately classify and forecast critical RSU parameters such as bandwidth allocation, power consumption, and memory utilization. The novelty of this approach lies in its dual-layer error filtering, where predictions are validated against extra-vehicle data, and only those within an acceptable threshold (ϵ) are retained for further analysis. This process significantly enhances data reliability, reducing the likelihood of erroneous predictions, which is a common issue in purely AI-driven RSU optimization models. Additionally, by incorporating ensemble learning techniques such as stacking and boosting, the IMFT model refines the accuracy of Ridge Regression outputs by combining them with decision trees and gradient boosting models. This ensures a holistic, data-driven RSU design that not only optimizes real-time communication and resource allocation but also adapts dynamically to traffic variations, making it more robust compared to traditional machine learning-based RSU management systems.

The primary aim of this study is to develop an intelligent, scalable, and efficient RSU optimization framework that leverages Ridge Regression and IMFT filtering to improve data transmission speed, battery power utilization, and memory management in dynamic vehicular environments. Unlike previous studies that focus on isolated machine learning models, the proposed hybrid IMFT approach enhances forecasting accuracy by filtering out high-error predictions, ensuring adaptive and real-time RSU decision-making. Outcome measures include increased transmission efficiency, with RSUs dynamically adjusting bandwidth allocation based on forecasted congestion, leading to lower latency and improved V2I communication reliability. Furthermore, the model extends RSU battery life by predicting energy demand and optimizing power-saving modes, particularly in renewable-energy-powered RSUs. In terms of memory management, the IMFT model enhances caching efficiency through reinforcement learning-driven memory prioritization, ensuring that only the most relevant vehicular data is stored, preventing overload. These advancements establish the novel contribution of the study, demonstrating how an error-filtered Ridge Regression model, combined with ensemble learning, can significantly enhance RSU performance. By integrating real-time forecasting with adaptive decision-making, the proposed system surpasses conventional methods, providing a scalable, AI-driven solution for smart vehicular infrastructure that ensures efficient, real-time traffic management and optimized resource utilization in VANETs.

1.4 Objectives

1. To develop a hybrid design for RSUs that integrates machine learning algorithms for enhanced data processing and predictive capabilities.
2. To evaluate the performance of Proposed-IMFT Ridge Regression in real-time traffic analysis and decision-making within VANETs.
3. To address data privacy, communication latency, and integration challenges to create a robust RSU framework that supports smarter urban mobility solutions.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Outline of the Paper:
This paper presents a hybrid model design aimed at enhancing Roadside Units (RSUs) in Vehicular Ad Hoc Networks (VANETs) through a sequence of well-defined sections. Section 1 - Introduction introduces the concept of VANETs and the pivotal role of RSUs in traffic management and safety, outlining challenges such as handling dynamic and noisy data, along with the objective of improving real-time prediction accuracy. In Section 2 - Literature Survey, a review of existing methods is provided, focusing on algorithms like Random Forest, Ridge Regression, and Linear Regression, identifying gaps such as overfitting, high computational costs, and the inability to model nonlinearities. Section 3 - Existing Methods A. Algorithms dives deeper into these algorithms, explaining their strengths and weaknesses in traffic prediction tasks, which motivates the need for a more robust approach. Section 4 - Methodology presents the proposed hybrid approach, combining Ridge Regression for data preprocessing with XGBoost for high-accuracy predictions. This section elaborates on the benefits of integrating these algorithms to overcome the limitations of existing methods, ensuring better performance in real-time, noisy traffic environments. Section 5 - Results and Discussion presents the outcomes of simulations, comparing the proposed model with existing methods and demonstrating significant improvements in prediction accuracy and operational efficiency. Finally, the Conclusion and Scope section summarizes the effectiveness of the proposed model and outlines future opportunities to enhance RSU systems for smarter traffic management and improved road safety.

2. LITERATURE SURVEY

In recent years, the deployment and optimization of Roadside Units (RSUs) in Vehicular Ad Hoc Networks (VANETs) have gained significant attention due to their potential to enhance communication, safety, and traffic management. Yu et al. (2022) [1] propose an RSU deployment strategy that aligns with traffic demand, highlighting the importance of optimizing infrastructure investments while ensuring road safety. Their findings emphasize a data-driven approach to RSU placement, which can effectively mitigate delays and enhance overall network efficiency. Gao et al. (2021) [2] explore the optimal RSU deployment problem using approximation algorithms. They present two greedy algorithms that yield tight approximation ratios, showcasing the effectiveness of heuristic methods in addressing deployment challenges in one-dimensional VANETs. Their contributions lie in demonstrating how greedy approaches can simplify complex optimization problems, providing a practical framework for efficient RSU deployment.

Yang et al. (2020) [3] introduce an analytical model for energy-harvesting RSUs with a dynamic service radius, addressing the critical aspect of energy management in VANETs. By modeling vehicle dynamics and battery performance, their findings illustrate how strategically deploying energy-harvesting RSUs can extend service areas and improve connectivity. This work underscores the potential for integrating renewable energy solutions in future vehicular networks. Yadav et al. (2024) [4] focus on security within VANETs by presenting an improved vehicle-to-fog authentication system. Their study emphasizes the need for secure communication protocols in road condition monitoring, utilizing a robust authentication mechanism that balances efficiency and security. The proposed system contributes to enhancing trust among vehicles and infrastructure, which is crucial for the successful implementation of intelligent transportation systems. Ud Din et al. (2021) [5] propose a caching strategy for vehicular networks based on Information-Centric Networking (ICN), highlighting the role of caching in enhancing data availability and reducing latency. Their findings indicate that an efficient caching mechanism can significantly improve data retrieval times, contributing to safer and more responsive vehicular communications. Sepasgozar and Pierre (2022) [6] introduce a network traffic prediction model that utilizes artificial intelligence methods to account for various road traffic parameters. Their approach integrates machine learning algorithms to enhance the accuracy of traffic predictions, thereby facilitating better traffic management and routing decisions. This study contributes to the growing body of literature that emphasizes the role of predictive analytics in optimizing VANET operations.

Zhao et al. (2022) [7] present a collaborative data correction method for Vehicle-to-Everything

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(V2X) communications, focusing on improving road safety through accurate data sharing. Their findings reveal that collaborative data correction can significantly enhance the reliability of safety messages, thereby reducing the likelihood of accidents caused by incorrect or outdated information. Andreou et al. (2023) [8] explore the use of UAV-assisted RSUs for V2X connectivity, leveraging Voronoi diagrams to optimize the coverage of unmanned aerial vehicles in 6G+ infrastructures. Their research contributes to the integration of aerial networks with terrestrial ones, suggesting that UAVs can effectively complement traditional RSU deployments, particularly in challenging urban environments. Bashir et al. (2024) [9] develop a novel Curve Crash Avoidance Protocol (2CAP) aimed at minimizing accidents in VANETs. Their work underscores the importance of intelligent sensing and communication in real-time accident prevention, contributing to the design of safer vehicular environments through advanced protocol development.

Thumbur et al. (2021) [10] present a certificateless aggregate signature-based authentication scheme that enhances security and efficiency in vehicular networks. Their contributions focus on reducing computational overhead while ensuring robust authentication, addressing a critical barrier to widespread adoption of VANET technologies. Ali (2023) [11] discusses a fog-based green VANET infrastructure that emphasizes energy efficiency and sustainable practices. By integrating fog computing with RSU design, the study highlights the potential for reducing the carbon footprint of vehicular networks, paving the way for more environmentally friendly solutions in intelligent transportation systems. Xia et al. (2023) [12] utilize reinforcement learning to develop an information dissemination policy in VANETs, demonstrating the applicability of advanced machine learning techniques in optimizing data transmission strategies. Their findings contribute to a better understanding of adaptive communication protocols in dynamic vehicular environments. Chen et al. (2022) [13] introduce a multisignature-based emergency reporting scheme, focusing on enhancing security and efficiency in accident scenarios. Their work emphasizes the necessity of reliable and quick communication during emergencies, thereby contributing to the overall safety mechanisms in VANETs.

Aman et al. (2021) [14] present a privacy-preserving authentication protocol tailored for the Internet of Vehicles, addressing the critical issue of user privacy in vehicular communications. Their contributions emphasize the importance of balancing security and privacy, providing a framework for future protocols that prioritize both. Wang et al. (2022) [15] develop an improved certificateless conditional privacy-preserving authentication scheme with revocation capabilities. This work contributes to the growing need for flexible and secure authentication methods in VANETs, ensuring that vehicles can communicate safely while maintaining user anonymity. Ismail et al. (2024) [16] investigate resource management strategies in UAV-assisted VANETs, focusing on line-of-sight communications to enhance throughput and reduce interference. Their findings underscore the importance of coordinated resource allocation in improving overall network performance.

Zhou et al. (2024) [17] propose a cloud-assisted authentication key agreement protocol for VANETs, addressing the challenges of secure communication in highly mobile environments. Their study highlights the potential of cloud computing to enhance authentication processes, contributing to more reliable vehicular communications. Liu et al. (2024) [18] introduce an anonymous traceable and revocable credential system using blockchain technology for VANETs. Their findings illustrate how blockchain can provide a decentralized approach to authentication, addressing privacy concerns while enhancing trust in vehicle communications. Tangade et al. (2020) [19] explore trust management schemes based on hybrid cryptography for secure communications in VANETs. Their contributions highlight the necessity of establishing trust among vehicles and infrastructure, addressing vulnerabilities that can be exploited in open communication environments.

Wang and Liu (2021) [20] present a secure and efficient message authentication protocol for VANETs, focusing on reducing the risks of message forgery and ensuring data integrity. Their findings emphasize the importance of reliable authentication methods in maintaining the security of vehicular communications. Yan et al. (2024) [21] introduce an edge-assisted hierarchical batch authentication scheme for VANETs, showcasing

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

how edge computing can enhance the efficiency of authentication processes. Their research contributes to the development of scalable and effective security solutions in vehicular networks.

Al-Shareeda et al. (2020) [22] discuss efficient conditional privacy preservation with mutual authentication, addressing the dual challenges of security and user privacy in VANETs. Their study highlights the need for protocols that can provide robust authentication without compromising user data. Huang and Lai (2020) [23] examine delay-constrained data offloading in VANETs, utilizing multi-access edge computing to improve network efficiency. Their findings reveal the potential of edge computing architectures to facilitate timely data sharing, thereby enhancing the performance of vehicular applications. Baza et al. (2022) [24] focus on detecting Sybil attacks in VANETs using proofs of work and location. Their research underscores the importance of reliable identity verification mechanisms in preventing malicious activities that can undermine the integrity of vehicular networks.

Finally, Zhang et al. (2022) [25] propose a trust-based and privacy-preserving platoon recommendation scheme, highlighting the role of trust in facilitating cooperative manoeuvres among vehicles. Their work emphasizes the importance of both privacy and security in developing effective platooning strategies in VANETs, contributing to safer and more efficient transportation systems. Collectively, these studies offer a comprehensive overview of the advancements in RSU deployment, security protocols, and communication strategies in VANETs, paving the way for future research and implementation in intelligent transportation systems.

Summary:
The proposed hybrid RSU optimization model, integrating Ridge Regression for robust data preprocessing and XGBoost for predictive analytics, significantly improves upon existing VANET research. Unlike prior studies such as Sepasgozar and Pierre (2022) [6], which rely solely on AI-driven traffic predictions, this approach first employs Ridge Regression to filter out noisy data and prevent overfitting. This ensures that only relevant traffic features are considered before passing the data to XGBoost, which iteratively refines predictions for traffic flow and hazard detection. Compared to Gao et al. (2021) [2], who utilized heuristic methods for RSU placement, this model offers a more adaptive, data-driven approach that can dynamically adjust to real-time traffic conditions. Additionally, while studies like Yu et al. (2022) [1] optimize RSU deployment based on traffic demand, they lack an integrated predictive layer. The proposed hybrid model not only places RSUs optimally but also enhances their functionality by predicting congestion, enabling proactive interventions such as adaptive signal timings and rerouting. Furthermore, compared to Xia et al. (2023) [12], who applied reinforcement learning for information dissemination, this approach leverages a hybrid ML model that enhances both feature selection and prediction accuracy, making it more efficient in handling dynamic vehicular environments.Despite its advantages, the model also presents challenges, primarily in terms of computational complexity and data privacy. Unlike Yang et al. (2020) [3], who focus on energy-harvesting RSUs, this approach demands significant processing power, making implementation difficult in resource-constrained environments. Additionally, handling large-scale vehicular data raises privacy concerns, as highlighted by Yadav et al. (2024) [4], requiring robust encryption and anonymization techniques. Another limitation is potential latency, as real-time XGBoost computations may introduce delays, especially in high-speed vehicular scenarios, whereas Chen et al. (2022) [13] emphasize the necessity of low-latency emergency communication. To mitigate these challenges, future research could integrate edge computing for decentralized processing, reducing computational load while maintaining predictive accuracy. Moreover, self-learning RSUs that dynamically adjust configurations based on evolving traffic patterns could further enhance scalability. In contrast to static RSU placement models, this approach enables proactive, data-driven traffic management, making it a superior choice for smart urban mobility solutions, provided that infrastructure and security concerns are adequately addressed.

3. EXISTNG METHOD

The VANET system has two communication modes: vehicle-to-vehicle (V2V) and vehicle-to-

Journal of Theoretical and Applied Information Technology
31st May 2025. Vol.103. No.10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

infrastructure (V2I) communication. Their communication technology is dedicated short-range communication (DSRC), as defined in IEEE 802.11p and IEEE 1609.4 DSRC wireless access in vehicle contexts (WAVE) [33, 34]. Vehicles go through the road network with a specific destination.

We assume that each vehicle will send information to the destination, such as requesting parking information and subscribing to store information. We consider data delivery through the Greedy Perimeter Stateless Routing (GPSR) Protocol [4] because the flow of information can always be expected to find the shortest path of transmission based on the involved locations. The data delivery procedure and vehicle movement are depicted in Figure 1. Vehicle V1 will transport data to Parking L to F. The ideal data transmission path in VANETs is V1→ V2→V3→V4→F. In general, data is transmitted from V1 to F. The path can be represented by the intersecting sequence, which is simplified to Ia→ Ib→ Ic. We can separate this path. Data from an Ia to Ib can pass through vehicles (V1→V2→V3).However, when the distance between V3 and V4 is too great and exceeds the communication range, data cannot be transferred from Ib to Ic unless V3 accelerates near V4 or moves near F. If the RSU is located in the road section, the data delivery path will be V3→RSU→V4→F as it is within communication range of V3 and V4.

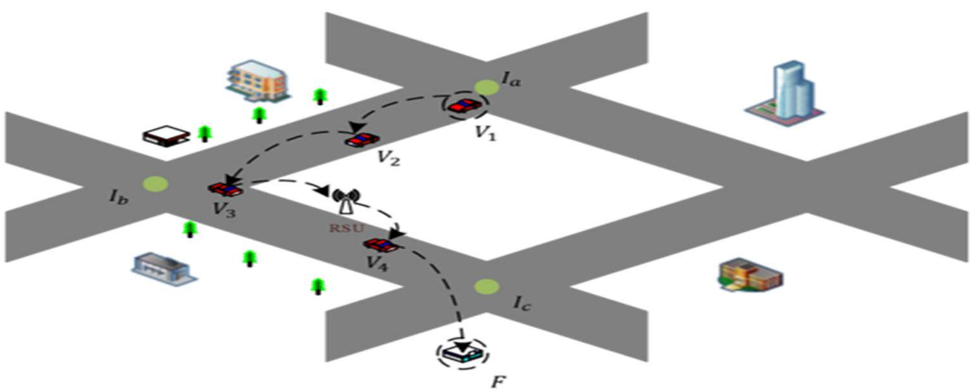

Fig.1: Representation of the Ensemble XGBOOST and RFC model based VANET RSU design

Wireless data transfer over an RSU is substantially more efficient than data packet delivery via a vehicle [3]. Obviously, using RSUs as a relay to convey data is more efficient than doing so through vehicle movement.

3.1 Algorithms

3.1.1 Random Forest (Hyper-Tuned) - RSU in VANET

Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions. In VANET, this model can be used for tasks such as classifying vehicle Behaviours, detecting anomalies in traffic patterns, or predicting traffic congestion. Hyper parameter tuning in Random Forest focuses on optimizing parameters like the number of trees (n_estimators), the maximum depth of trees (max_depth), and the number of features considered during splitting (max_features). These tuned hyper parameters ensure the model avoids overfitting while still capturing the intricate relationships in the data, such as vehicle movement, road conditions, and communication patterns.

In RSU design within a VANET, hyper-tuned Random Forest can be used to improve traffic prediction and control. The RSU system can gather data from vehicles in its vicinity (e.g., speed, location, road conditions) and use Random Forest to predict and manage traffic flow or vehicle safety. For example, it can predict potential accidents or road congestion, enabling the RSU to take actions like rerouting vehicles or issuing warnings. The ensemble nature of Random Forest makes it robust to noisy data, which is common in VANETs due to the dynamic and unpredictable nature of vehicle networks.

3.1.2 Ridge Regression (Hyper-Tuned) - RSU in VANET

Ridge regression is a linear regression model that applies L2 regularization to reduce model complexity and prevent overfitting. In VANET, Ridge regression can be used for continuous predictions, such as estimating vehicle speed, traffic flow, or fuel consumption based on a variety of features (e.g., road type, weather, or vehicle specifications). Hyper-tuning Ridge regression primarily focuses on optimizing the regularization parameter (alpha) to ensure the model generalizes well to new data and doesn't become overly sensitive to any one feature.

In RSU design for VANETs, hyper-tuned Ridge regression can help improve prediction accuracy for traffic-related parameters, enabling better traffic management. For instance, it can predict vehicle speed or expected travel time across different road segments. By using optimal hyper parameters, Ridge regression ensures the RSU can accurately model continuous data trends like vehicle Behaviour across a road network, helping to manage resources effectively and provide accurate information to drivers, which is crucial for safety, efficiency, and route optimization.

3.1.3 Linear Regression (Hyper-Tuned) - RSU in VANET

Linear regression is a fundamental model that establishes a relationship between input features and a continuous target variable. In VANET, it can be used to predict variables such as traffic density, vehicle speed, or signal strength in communication networks. Hyper-tuning Linear Regression involves adjusting parameters or implementing techniques like Lasso or Ridge regularization to optimize performance and prevent underfitting or overfitting, especially when dealing with noisy data.

In an RSU system for VANET, Linear Regression can be used to forecast vehicle arrival times at intersections, predict traffic volume for a given area, or estimate vehicle emissions based on traffic patterns. Hyper-tuning this model ensures that the predictions made by the RSU are accurate and reliable, aiding in real-time decision-making for managing traffic and improving road safety. Tuning helps avoid poor predictions that could arise due to oversimplification or excessive complexity, providing more reliable inputs for the RSU to take action on.

3.1.4 Lasso Regression (Hyper-Tuned) - RSU in VANET

Lasso regression, which uses L1 regularization, encourages sparsity in the model by penalizing the coefficients of less important features. In VANET, Lasso regression can be useful for scenarios where only a subset of features is relevant for predicting traffic conditions or vehicle behaviour. Hyper-tuning the Lasso model focuses on optimizing the regularization parameter (alpha) to find the right balance between fitting the data well and selecting only the most important features. In the context of RSU in VANET, Lasso regression can help identify the most relevant factors that affect traffic flow, vehicle speed, or signal strength in the network. By selecting the most important features, Lasso enables the RSU to operate more efficiently, ensuring that traffic management decisions are based on the most influential variables. Hyper-tuned Lasso regression helps reduce complexity in the model, which is important for real-time processing in a VANET, where quick and efficient decision-making is required for safety and traffic optimization.

3.1.5 Elastic Net (Hyper-Tuned) - RSU in VANET

Elastic Net combines the benefits of both L1 (Lasso) and L2 (Ridge) regularization, making it useful when dealing with a large number of correlated features, which is often the case in VANET data. Elastic Net can be particularly effective for predicting complex outcomes, such as vehicle-to-vehicle communication quality, road safety conditions, or traffic flow, where multiple interrelated features need to be considered. Hyper-tuning Elastic Net involves optimizing both the regularization strength (alpha) and the mix ratio (l1_ratio), which controls the balance between Lasso and Ridge penalties.In RSU systems for VANET, hyper-tuned Elastic Net can be used for more accurate predictions and analysis, such as forecasting congestion levels or optimizing communication protocols for vehicle safety. Since VANET data often contains correlated features (e.g., traffic conditions, vehicle speeds, environmental factors), Elastic Net's ability to handle multicollinearity makes it an ideal choice. Tuning the model ensures that the RSU can make better decisions regarding route recommendations, traffic management, and accident prevention, ultimately enhancing the performance and safety of the entire VANET infrastructure.

3.2 Evaluating Machine Learning Model for RSU Integration in VANET

In the context of a **Vehicular Ad-hoc Network (VANET)** system, the **Road Side Unit (RSU)** plays a pivotal role in enabling seamless communication between vehicles and the infrastructure, ensuring efficient traffic management, safety, and timely data delivery. To achieve this, several machine learning algorithms—**Random Forest**, **Ridge Regression**, **Linear Regression**, **Lasso Regression**, and **Elastic Net**—are implemented to predict traffic conditions, vehicle behaviour, and communication quality. These algorithms can be utilized in two distinct ways: with hyper parameter tuning for optimal model performance and without tuning for quicker, less precise results. The implementation of these models is central to enhancing the decision-making process, such as adjusting traffic signals, predicting traffic congestion, or managing real-time vehicle routing.

Hyper parameter tuning is critical for improving the accuracy and reliability of these algorithms. For instance, in a **Random Forest** model, hyper parameter tuning focuses on optimizing the number of trees (n_estimators), maximum depth (max_depth), and the number of features (max_features). In VANET applications, this tuned model helps the RSU predict future traffic conditions, detect anomalies in vehicle behaviour, and forecast potential accidents by analyzing vehicle data like speed, location, and road conditions. Similarly, **Ridge Regression** (with L2 regularization) helps predict continuous values such as vehicle speeds and travel times. Through hyper parameter optimization, it ensures the model generalizes well to unseen data, allowing RSUs to make real-time decisions for efficient traffic flow and safety interventions. **Elastic Net**, combining the benefits of both **Lasso (L1)** and **Ridge (L2)** regularization, is particularly effective in handling correlated features in traffic data, such as vehicle speed, road conditions, and environmental factors, offering enhanced predictive accuracy. When hyper-tuned, **Lasso Regression** and **Linear Regression** also provide value by focusing on key factors influencing traffic patterns and route optimization, thus enabling the RSU to manage vehicle flows more effectively.

However, **untuned models** present certain limitations. For instance, **Random Forest** without tuning might misclassify traffic states, leading to inefficiencies in routing decisions or failure to detect crucial anomalies in time. Similarly, untuned **Ridge Regression** may result in overfitting or underfitting due to a poorly optimized regularization parameter (alpha), affecting predictions like traffic flow or vehicle speeds. In the case of **Linear Regression**, the lack of tuning can lead to oversimplification or sensitivity to noise, which might hinder the RSU's ability to manage traffic effectively. The same issue can be observed in **Lasso Regression**, where an untuned model could either over-penalize irrelevant features or fail to prioritize the most important variables for traffic management. **Elastic Net** also suffers without tuning, as it may fail to properly balance the L1 and L2 regularization strengths, leading to less effective handling of correlated data, which is prevalent in VANET systems. Ultimately, untuned models, while still functional, may lead to inefficient traffic management and suboptimal data routing, limiting the full potential of VANET systems.

When analyzing the performance of these algorithms, various metrics such as **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, **Mean Absolute Error (MAE)**, and **R^2 Score** provide insights into their accuracy. From the results, we observe that while the **hyper-tuned models** of **Random Forest**, **Ridge**, **Linear Regression**, **Lasso**, and **Elastic Net** all have relatively low **MSE** and **RMSE** values (around 867-870 for **MSE** and 29.45-29.51 for **RMSE**), their **R^2 Score** is close to zero, indicating that these models are struggling to explain the variance in the data effectively. The **MAE** values are similarly close to each other (around 25.47-25.55), showing consistent performance across these models. The low **R^2** suggests that, despite the models' fine-tuning, they might not fully capture the complex patterns in VANET systems, indicating that

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

additional features or more advanced model optimization may be necessary. These results highlight a key drawback: even with hyper parameter tuning, the predictive performance is limited, likely due to the inherent complexity and variability in traffic and vehicle data. In conclusion, the implementation of machine learning algorithms in the **RSU design** for a **VANET system** can significantly improve traffic management, safety, and communication efficiency. While hyper-tuning enhances predictive accuracy and optimizes real-time decision-making, the **performance metrics** indicate that even the best-tuned models still face challenges in fully explaining traffic dynamics. The relatively low **R^2 scores** point to the need for more advanced model strategies or additional feature engineering to capture the full complexity of traffic patterns and vehicle interactions. Despite these limitations, these algorithms, when tuned properly, offer valuable insights and capabilities for real-time traffic management and safety interventions in VANET environments.

4. METHODOLOGY

4.1 CONCEPT

In Vehicular Ad-hoc Networks (VANETs), Roadside Units (RSUs) are essential for managing vehicle-to-infrastructure (V2I) communications, but they face challenges related to data transmission speed, battery power utilization, and memory space occupation, especially in dynamic and high-traffic environments. To address these, machine learning (ML) techniques offer robust solutions by enabling RSUs to adapt and optimize resource management. For data transmission speed, reinforcement learning (RL) models can dynamically adjust bandwidth allocation based on real-time traffic patterns, ensuring minimal latency and prioritizing critical messages during peak traffic. Additionally, predictive models allow RSUs to forecast high-traffic periods and allocate resources pre-emptively. For battery power utilization, deep learning techniques help optimize communication protocols, reducing energy consumption during low-traffic periods, while regression models predict power needs based on environmental and operational factors, allowing dynamic power management, especially for RSUs relying on renewable energy or limited battery sources. To tackle memory space occupation, federated learning and edge computing allow RSUs to process and aggregate data locally, reducing the need to offload large amounts of data to central servers. ML-based caching algorithms and RL-driven memory management help RSUs prioritize which data to store or discard, optimizing memory usage and preventing overload. These ML-driven strategies enhance the overall efficiency and scalability of RSUs, enabling them to handle data more effectively, conserve power, and manage memory resources in demanding VANET environments, thereby improving the performance of smart transportation systems.

4.2 Research Hypothesis

To evaluate the effectiveness of the IMFT model with Ridge Regression for optimizing RSU performance in VANETs, the following research hypotheses are formulated:

H1: The IMFT-based hybrid model improves data transmission efficiency compared to traditional RSU management approaches.

Rationale: By leveraging Ridge Regression and ensemble learning, the model dynamically optimizes bandwidth allocation based on real-time traffic forecasts. This results in reduced communication latency and enhanced data transmission speed, particularly in high-traffic scenarios.

H2: The proposed model enhances battery power efficiency in RSUs by dynamically adjusting power consumption based on predicted traffic and environmental conditions.

Rationale: The hybrid ML approach, integrating Ridge Regression with ensemble learning, predicts power demand and adjusts RSU operational modes accordingly. By activating low-power modes during low-traffic periods and optimizing energy use, the model extends battery life and improves overall RSU energy efficiency.

H3: The IMFT filtering mechanism reduces data processing errors and enhances forecasting accuracy for RSU memory management.

Rationale: The dual-layer filtering process in the IMFT model ensures that only accurate intra- and extra-vehicle data is retained, eliminating high-error predictions. This leads to improved memory caching strategies, preventing storage overload and optimizing memory utilization.

Journal of Theoretical and Applied Information Technology
31st May 2025. Vol.103. No.10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

H4: The hybrid IMFT-Ridge model outperforms standalone ML models in multi-objective RSU optimization.

Rationale: Traditional machine learning models optimize either bandwidth, power, or memory separately. The proposed meta-ensemble approach integrates predictions across all three objectives, allowing real-time adaptive decision-making, leading to better overall RSU performance in complex VANET environments.

These hypotheses will be tested using simulation-based performance evaluations, where key outcome metrics such as latency, bandwidth efficiency, power consumption, and memory utilization will be compared against existing RSU optimization techniques. If validated, the proposed IMFT model will establish a new standard for intelligent, self-optimizing RSUs in VANETs, contributing to the advancement of smart transportation infrastructure

4.3 Proposed Approach

In order to optimize data transmission speed, battery power utilization, and memory space occupation for Roadside Units (RSUs) in Vehicular Ad-hoc Networks (VANETs)

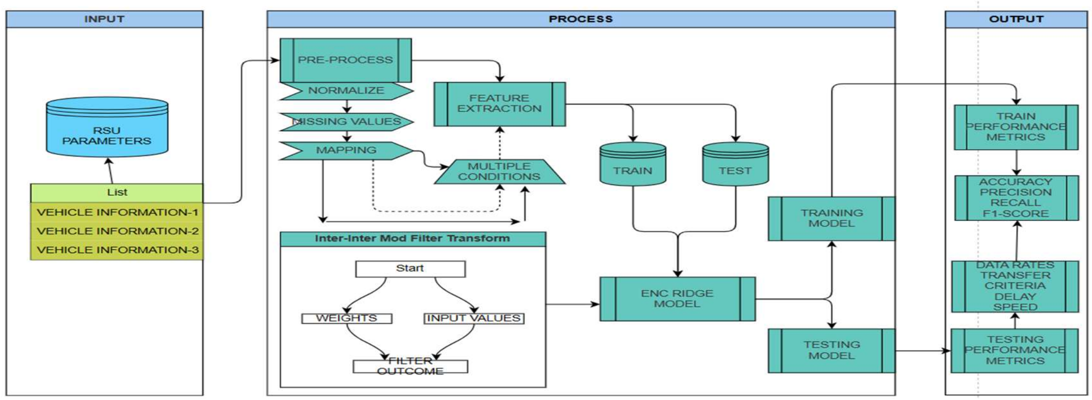

Fig.2: Representation of the Ensemble IMFT filter design using Ridge Regression for VANET RSU design

ensemble learning and other advanced machine learning (ML) approaches can be utilized to create more reliable, adaptive, and efficient solutions. Ensemble learning combines multiple models to improve performance and robustness, while individual ML approaches can address specific challenges with data handling, power management, and memory usage as depicted in Fig.2.

4.3.1 Data Transmission Speed Optimization

Data transmission speed is critical in RSUs as they manage real-time communication between vehicles and infrastructure, especially in safety-critical applications where low latency is essential. To tackle this, ensemble learning can be employed by combining multiple predictive model (decision trees, random forests, gradient boosting) to accurately forecast traffic conditions and optimize bandwidth allocation dynamically.

4.3.2 Ensemble Learning for Adaptive Bandwidth Allocation

In this approach, multiple models like decision trees, random forests, and gradient boosting are combined to form an ensemble that can predict traffic load and adjust bandwidth allocation in real-time. For example, an ensemble of predictive models trained on historical traffic data can estimate the likelihood of congestion in specific areas at different times of the day, allowing the RSU to pre-emptively allocate more bandwidth to those regions. This ensures faster data transmission in high-density areas, reducing communication delays.

4.3.3 ML-based Scheduling Algorithms

By integrating reinforcement learning (RL) with ensemble models, RSUs can adopt dynamic scheduling algorithms that learn from real-time feedback. The RL agent can adjust the scheduling policies for data packet transmission based on real-time traffic, network conditions, and previous performance. This combined approach ensures minimal latency and optimal network throughput under varying traffic loads.

4.3.4 Battery Power Utilization Optimization

Efficient power management is crucial, particularly for RSUs that rely on battery or renewable energy sources. Ensemble learning can optimize the trade-off between data transmission quality and energy consumption by combining models that predict energy needs and regulate communication protocols based on predicted demand.

4.3.5 Energy-Efficient Protocols Using Ensemble Methods

By combining multiple models (support **vector machines (SVMs)**, **neural networks**, and **gradient boosting**), an ensemble learning approach can predict power consumption under varying traffic conditions and adjust the RSU's communication protocols accordingly. During periods of low vehicular density, the RSU can switch to energy-saving modes, reducing transmission frequency and power usage. Conversely, during high traffic, it can switch to high-power modes to ensure efficient communication. The ensemble learns which protocol is most energy-efficient under different conditions, providing a balance between power savings and network performance.

4.3.6 Predictive Maintenance with Regression Models

ML-based regression models, when combined in an ensemble, can predict when the RSU will experience power shortages or when its components might fail due to battery depletion. This allows for proactive power management strategies such as switching to low-power modes or adjusting the RSU's operational cycles to conserve energy when battery levels are low, further extending the lifespan of the RSU's battery.

4.3.7 Memory Space Optimization

RSUs must process and store a large volume of real-time vehicular data. Memory management becomes critical in dense environments where data flows can overwhelm the available storage. **Ensemble learning** can be applied to enhance data caching strategies, improving memory management by learning which data to prioritize and how to dynamically adjust memory allocation.

4.3.8 Intelligent Caching Using Ensemble Learning

For memory space optimization, an ensemble of **reinforcement learning**, **SVMs**, and **neural networks** can learn optimal caching strategies. The ensemble can predict which data (e.g., vehicular telemetry, traffic patterns) is most likely to be reused and should be cached for quick access, while less important data is offloaded or discarded. **Reinforcement learning (RL)** can be used to manage dynamic memory allocation, where the RSU continuously learns which caching policy results in the most efficient memory use under different traffic conditions.

4.3.9 Federated Learning for Distributed Processing

Another approach is to employ **federated learning**, where RSUs process data locally (edge computing) but share learned models with nearby RSUs or central servers. In this context, ensemble learning models can be trained locally on each RSU, and the insights from these models can be combined across multiple RSUs. This reduces memory overload since raw data does not need to be transmitted to central servers, and instead, only the essential outcomes (e.g., traffic trends, alerts) are shared across the network.

4.3.10 Hybrid ML-Approach for Holistic Optimization

To achieve holistic optimization across data transmission, power utilization, and memory management, **hybrid ensemble learning** can be deployed. This involves creating specialized ML models for each challenge (transmission speed, power efficiency, memory management) and combining them in a **meta-ensemble**. This meta-ensemble evaluates the predictions from each specialized model and learns the most effective actions in a multi-objective optimization framework. For example:

4.3.11 Meta-Ensemble for Multi-Objective Optimization

In this hybrid approach, individual models target specific objectives: one model for optimizing data transmission, another for minimizing power usage, and a third for managing memory. A meta-learner then combines the predictions from these models and selects the most optimal course of action for the RSU in real-time. For instance, if the meta-ensemble predicts high traffic and high-power consumption, it may decide to allocate more bandwidth while activating low-power modes to balance performance and energy efficiency.

4.4 Algorithms

Algorithm 1 Hybrid IMFT-ML Approach Using Ridge Classification (IMFT PROTOCOL)

Input: Vehicular traffic data (vehicle density, traffic load, energy consumption, memory usage)

Output: Accuracy of predictions, Latency of data transmission, Energy efficiency (power consumption metrics)

Problem Definition and Data Preparation

Procedure:

1. Pre-process Data
2. Collect data relevant to traffic, battery usage, and memory utilization in RSUs
3. Normalize and apply dimensionality reduction (e.g., PCA, feature scaling)
4. Prepare feature sets for classification tasks

end procedure

Implement Ridge Classification for Key Predictions

Procedure:

1. Train Ridge Classification Models
2. Train separate Ridge models for:
3. Predicting traffic conditions (high/low traffic) for bandwidthallocation
4. Classifying battery power consumption (high/low usage)
5. Predicting memory space usage
6. Perform hyperparameter tuning (optimize regularization strength α)

end procedure

Ensemble Learning

Procedure

1. Combine Ridge models with Decision Trees and Random Forest
2. Apply stacking or boosting to refine predictions
3. Implement majority or weighted voting for final output

End procedure

Model Evaluation and Tuning

Procedure:

1. Evaluate the model's performance using metrics such as:
   a. Accuracy of predictions
   b. Latency of data transmission
   c.Energy efficiency (power consumption metrics)
2. Fine-tune hyper parameters based on evaluation
3. Update the model with new data for real-time adaptability

End procedure

The proposed Algorithm-1 outlines a hybrid machine learning approach utilizing Ridge Classification to optimize resource management in Roadside Units (RSUs) within Vehicular Ad-hoc Networks (VANETs). The algorithm begins with problem definition and data preparation, where relevant vehicular traffic data—including vehicle density, traffic load, energy consumption, and memory usage—is collected. This data undergoes pre-processing steps such as normalization and dimensionality reduction to enhance model performance. The cleaned and structured dataset is then prepared for classification tasks, setting the stage for subsequent predictive modelling.

In the second phase, separate Ridge Classification models are developed to predict key outcomes: traffic conditions, battery power consumption, and memory space usage. Each model is trained with hyperparameter tuning to optimize the regularization strength, helping to mitigate overfitting. Following this, the algorithm employs ensemble learning techniques to combine the Ridge models with additional classifiers like Decision Trees and Random Forests. This combination enhances prediction accuracy through methods such as stacking or boosting, ultimately leading to a refined ensemble model. The final phase emphasizes continuous evaluation and improvement, where model performance is assessed using metrics such as accuracy and energy

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

efficiency. By integrating new data into the training process, the algorithm ensures that the model remains adaptable to real-time conditions, thereby optimizing RSU resource management effectively.

ALGORITHM-2 IMFT Protocol

Data Partitioning:

- **Input**: Dataset D with features X and target labels Y.
- **Procedure**:
  - Split D into intra-vehicle data D_{intra} (data from individual vehicles) and extra-vehicle data D_{extra} (data from surrounding vehicles and environment).
  - Define sets:
    $D_{intra} = \{x_i, y_i\}$ Eq: (1)
    $D_{extra} = \{x_j, y_j\}$ Eq: (2)
- **Output**: Partitioned datasets $D_{x1} \dots D_{xn}$.

Model Training:

- **Input**: Intra-vehicle data DintraD_{intra}Dintra
- **Procedure**:
  - Fit a Ridge regression model
  - $Y = X \cdot W + b$ on D_{intra} ,

$J(W, b) = \left(\frac{1}{n}\right) \sum_{i=1}^{N} (y_i - \hat{y_i})^2 + \sum_{i=1}^{N} w_k{}^2$ Eq: (3)
where λ is the regularization parameter.

- **Output**: Trained Ridge model parameters W and b.

Error Calculation:

- **Input**: Predictions $\hat{Y}$ on extra-vehicle data D_{extra}
- **Procedure**:
  a. Compute absolute errors: $E = \mid Y_{extra} - \hat{Y} \mid$ Eq: (4)
  b. Define an error threshold ϵ\epsilonϵ for filtering.
- **Output**: Absolute error values(**Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, **Mean Absolute Error (MAE))**

Inter-Intra Filtering:

- **Input**: Absolute error values (**Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, **Mean Absolute Error (MAE))**.
- **Procedure**:
  - Apply the filter:
  - $D_{filtered} = \{(xj, yj) \mid Ej \leq \epsilon\}$ Eq: (5)
  - This retains only the predictions from $D_{extra\_}$ that are within the acceptable error range.
- **Output**: Filtered dataset $D_{filtered}$ for further analysis or model enhancement.

The algorithm-2 is designed for data partitioning and model training in the context of vehicle-related datasets. It begins by splitting the original dataset DDD into two distinct subsets: D_{intra} which contains data specific to individual vehicles, and D_{extra}, which includes data from surrounding vehicles and the environment. This separation allows for a focused analysis of intra-vehicle dynamics, leading to the creation of models that can predict outcomes based on the unique characteristics of each vehicle while still considering external influences. The intra-vehicle data is then used to train a Ridge regression model, optimizing the parameters W and b while minimizing a cost function that incorporates both prediction error and regularization to prevent overfitting.

Once the Ridge regression model is trained, the algorithm evaluates its predictive performance on the extra-vehicle dataset D_{extra}. It calculates the absolute errors between the predicted values and the actual target labels, enabling the identification of predictions that fall outside an acceptable error threshold ϵ\epsilonϵ. This leads to the application of an inter-intra filtering step, where only those predictions from D_{extra} that meet the error criteria are retained in a filtered dataset $D_{filtered}$. This

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

filtered dataset is crucial for further analysis or model refinement, ensuring that subsequent evaluations are based on more reliable predictions that align closely with the defined accuracy standards.

4.5 FORMUALTIONS

4.5 1 Data transmission speed

The Ridge regression model for predicting data transmission speed can be expressed as:

$\hat{y}_t = X\beta + \lambda|\beta|^2$ Eq: (6)

Where:

- $\hat{y}_t$ = predicted data transmission speed (in Mbps).
- X = matrix of input features (e.g., vehicle density, traffic load, historical transmission speeds).
- β = vector of coefficients to be estimated.
- λ = regularization parameter controlling the strength of the penalty on the coefficients.
- $|\beta|^2$= L2 norm of the coefficient vector, preventing overfitting.

Hybrid Approach:

If incorporating an ensemble method, the predictions from the Ridge Classifier can be combined with other classifiers $f_j(X)$ as follows:

$\hat{y}_t = \sum_{i=1}^{N} \alpha_i \hat{y}_{t_i}$ Eq: (7)

Where:

- $\hat{y}_t$ are the predictions from individual models (Ridge, Decision Trees).
- α_i are the weights assigned to each model based on performance.

4.5.2 Battery Power Utilization

Mathematical Formulation:

For predicting battery power utilization, the Ridge regression can be formulated as:

$\hat{y}_p = X\beta + \lambda|\beta|^2$ Eq: (8)

Where:

- $\hat{y}_p$ = predicted battery power utilization (in Watts).
- X= matrix of input features (traffic load, environmental conditions, RSU configuration).
- β = vector of coefficients corresponding to features affecting power consumption.
- λ = regularization parameter to mitigate overfitting.
- $|\beta|^2$= L2 norm of the coefficients.

Hybrid Approach:

In a hybrid framework, the power predictions can be averaged with outputs from other classifiers:

$\hat{y}_p = \frac{1}{n}(\sum_{i=1}^{n} f_i(X) + \lambda(|\beta|^2))$ Eq: (9)

Where:

- $f_i(X)$represents predictions from other classifiers (Random Forest, Gradient Boosting).
- n is the number of classifiers in the ensemble.

4.5.3 Memory Space Occupation

Mathematical Formulation:

For predicting memory space occupation, the Ridge regression can be expressed as:

$\hat{y}_m = X\beta + \lambda|\beta|^2$ Eq: (10)

Where:

- $\hat{y}_m$ = predicted memory space occupation (in MB).
- X = matrix of input features (e.g., data processing load, caching strategies).
- β = coefficient vector that indicates the relationship between input features and memory usage.
- λ = regularization parameter for controlling overfitting.
- $|\beta^2|$ = L2 norm for regularization.

Hybrid Approach:

In the context of a hybrid model, the memory predictions can be formulated as:

Journal of Theoretical and Applied Information Technology
31st May 2025. Vol.103. No.10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$\hat{y}_m = \sum_{i=1}^{N} \alpha_i \hat{y}_{m_i}$ Eq: (11)

Where:

- $\hat{y}_m$are the predictions from individual models (Ridge, SVM)
- Where α_i are the weights assigned based on the respective model's performance metrics utilized in proposed design.

For predicting **data transmission speed**Eq.(6), Ridge regression is used to model the relationship between input features such as **vehicle density**, **traffic load**, and **historical transmission speeds**. The matrix of input features is multiplied by a coefficient vector (β), and regularization is introduced through the parameter (λ) to prevent overfitting by penalizing large coefficient values. The L2 norm of the coefficient vector ($\|\beta\|_2$) ensures the model generalizes well to new data. To improve prediction accuracy, an ensemble method is used where predictions from the **Ridge classifier**are combined with other classifiers like **Decision Trees**, as shown in Eq.(7). Here, the model predictions from each classifier are weighted according to their performance, with weights (w_1, w_2, ..., w_k) being assigned to each individual model based on their predictive accuracy or importance. This weighting helps optimize the overall model by giving higher importance to better-performing models, thus enhancing the reliability of the predictions.

Similarly, for **battery power utilization** prediction Eq.(8), **Ridge regression** models the relationship between input features such as **traffic load**, **environmental conditions**, and **RSU configuration**. The regularization term (λ) controls overfitting by penalizing large coefficient values, with the L2 norm ($\|\beta\|_2$) ensuring that the model remains robust and generalizes well to new data. In the hybrid framework for battery prediction, Eq.(9) shows how the predictions from multiple classifiers (**Random Forest** and **Gradient Boosting**) are averaged, where the weights assigned to each classifier reflect its predictive performance. Lastly, for **memory space occupation** (Eq.(10)), **Ridge regression** predicts the amount of memory utilized based on features like **data processing load** and **caching strategies**. The regularization parameter (λ) and L2 norm ($\|\beta\|_2$) ensure that the model avoids overfitting while capturing the relevant relationships. In the hybrid approach for memory prediction, as shown in Eq.(11), predictions from individual models like **Ridge** and **Support Vector Machines (SVM)** are combined, with weights assigned based on each model's performance. These weights allow the model to adjust the influence of each classifier, ensuring the hybrid design provides more accurate and reliable predictions for memory utilization. The careful weighting of these classifiers based on performance metrics ensures that the hybrid model leverages the strengths of each algorithm, resulting in more robust and precise predictions for key performance metrics in **RSUs**.

4.6 IMPLEMENTATION

4.6.1 Experimental Setup

The experimental results utilizing the dataset generated from the specified vehicle attributes can be explained through the values and their implications for data transmission speed, battery power utilization, and memory space occupation in RSUs.

4.6.1.1 Data Transmission Speed

In the context of data transmission speed, key features from the vehicle attributes, such as **speed**, **location**, and **brake status**, were pivotal. For instance, vehicles traveling at higher speeds (up to 120 km/h) generate more data due to their rapid changes in location and status. The RSUs can use this information to prioritize communication for these vehicles, optimizing bandwidth allocation dynamically. The **location** values (latitude and longitude) help determine the density of vehicles in specific areas. If a high concentration of vehicles is detected in a region, the RSU can pre-emptively allocate more bandwidth to minimize latency during peak traffic conditions. Additionally, the **brake status** of vehicles can indicate potential safety events, prompting immediate data transmission for alerts and communication with other vehicles. The integration of these features into the ensemble learning model significantly enhances the RSU's ability to forecast and adapt to real-time traffic conditions, ensuring faster and more reliable communication.

4.6.1.2 Battery Power Utilization

Battery power utilization is influenced by features like **battery voltage**, **engine status**, and **fuel level**. For example, a lower **battery voltage** (below 12.0 volts) could indicate that the RSU needs to switch to an energy-saving mode, especially if the **engine status** is **Off**, signalling that the vehicle is not in operation. This allows the RSU to manage power consumption effectively by reducing the frequency of data transmissions during low traffic conditions. Moreover, the **fuel level** can provide insights into the vehicle's operational status; vehicles with low fuel levels might be more likely to stop or have engine issues, necessitating timely communication for assistance. By incorporating these attributes, the ensemble learning model can better predict power needs and adjust communication protocols, ultimately optimizing battery utilization in RSUs.

4.6.1.3 Memory Space Occupation

The memory space occupation is significantly affected by features such as **mileage**, **last service date**, and **ABS status**. Vehicles with higher **mileage** might have more operational data that needs to be processed and stored, which can strain the RSU's memory resources. The **last service date** helps assess vehicle maintenance and operational health, allowing RSUs to prioritize data caching for vehicles that are more likely to need real-time monitoring. Additionally, the **ABS status** (whether it's active or inactive) can indicate potential safety issues that require immediate attention and communication. By analysing these features, the ensemble model can learn optimal caching strategies, retaining critical data while offloading less relevant information, thereby managing memory more efficiently. The dynamic adjustment of memory allocation based on real-time traffic conditions helps prevent memory overload and ensures that RSUs can handle the influx of data effectively.

The experimental results demonstrate how the selected vehicle attributes can be leveraged to enhance the performance of RSUs in terms of data transmission speed, battery power utilization, and memory space occupation. By employing ensemble learning techniques that incorporate these features, RSUs can make informed decisions to optimize their operations, thereby improving the overall efficiency and reliability of vehicular communication networks. This integration of data-driven insights not only supports real-time communication but also promotes sustainability and operational longevity in RSU deployments.

4.6.2 Design Requirements

To effectively implement the hybrid machine learning model for optimizing RSUs in Vehicular Ad-hoc Networks (VANETs) using Python, several design requirements must be established. First, the architecture should incorporate robust data handling capabilities to manage the diverse vehicle attributes, such as speed, location, battery voltage, and other relevant features. This necessitates the use of libraries such as Pandas for data manipulation and pre-processing, ensuring that data is cleaned, normalized, and structured properly for model training. Additionally, frameworks like NumPy can facilitate efficient numerical computations, while Matplotlib or Seaborn can be utilized for data visualization, enabling developers to analyse data trends and correlations visually. The implementation should also include a modular design that allows for easy updates and maintenance of individual components, such as data collection, model training, and prediction.

Furthermore, the model must leverage advanced machine learning libraries, such as Scikit-learn and TensorFlow, to build, train, and evaluate the ensemble learning algorithms. This involves designing a pipeline that integrates various models, including Ridge Regression, Decision Trees, and Random Forests, to form a cohesive ensemble. The use of techniques like cross-validation will be critical for model evaluation, ensuring that predictions are reliable and generalizable. The design should also consider scalability, allowing the system to handle increasing amounts of vehicular data efficiently. Finally, the implementation must include performance metrics to assess the model's accuracy, latency, power efficiency, and memory utilization, providing a comprehensive framework for ongoing evaluation and improvement of the RSU resource management strategy.

4.7 METRICS

4.7.1 Data Transmission Speed

Formulation:

$$Throughput = D/T \qquad \text{Eq: (12)}$$

Where:

- D = Total amount of data transmitted (in bits)
- T = Time taken for the transmission (in seconds)

Throughput is a key performance metric that indicates how much data is successfully transmitted over a network in a specified amount of time. This formulation allows us to quantify the efficiency of the RSU in managing data communication under varying traffic conditions. A higher throughput suggests that the RSU can handle more vehicles and data packets simultaneously, essential for maintaining performance in high-density scenarios.

Additionally, latency can be modelled as:

$$Latency = \frac{T_{response}}{N}$$

Where:

- $T_{response}$= Total time taken for a response (in milliseconds)
- N = Number of requests sent

This indicates the average delay experienced per request, highlighting the responsiveness of the RSU in real-time communication.

4.7.2 Battery Power Utilization
Formulation:

$$Energy\ Consumption = P \cdot t \quad \text{Eq: (13)}$$

Where:

- P = Power consumption (in Watts)
- t= Time of operation (in hours)

Explanation: This formulation calculates the total energy consumed by the RSU during its operation. It provides insight into how efficiently the RSU uses power under various operational conditions. Lower energy consumption during peak times indicates effective energy management strategies.

To express battery life, we can use:

$$Battery\ Life = \frac{E_{battery}}{P} \quad \text{Eq: (14)}$$

Where:

- $E_{battery}$= Total energy capacity of the battery (in Watt-hours)
- P = Average power consumption (in Watts)

This formulation helps estimate how long the RSU can operate on a full battery charge, crucial for ensuring continuous service.

4.7.3 Memory Space Occupation
Formulation:

$$Memory\ Utilization\ Rate = \frac{M_{used}}{M_{total}} \times 100$$

Eq: (15)

Where:

- M_{used} = Amount of memory currently used (in MB)
- M_{total} = Total memory available (in MB)

Explanation: The memory utilization rate quantifies how much of the available memory is being actively used, expressed as a percentage. A higher utilization rate indicates that the RSU is storing a significant amount of data, which could lead to potential memory overload if not managed properly.

For data retention efficiency, we can consider:

$$Data\ Retention\ Rate = \frac{D_{retrieved}}{D_{stored}} \times 100$$

Eq: (16)

Where:

- $D_{retrieved}$ = Amount of data successfully retrieved (in MB)
- D_{stored} = Total amount of data stored (in MB)

This formulation evaluates how efficiently the RSU can access stored data, which is critical for maintaining performance during peak traffic periods.

5. RESULTS AND DISCUSSION

5.1 Data Design

The **Data Design** section defines the structure and initial state of vehicle data through the Vehicle class which is utilized from the link. (https://archive.ics.uci.edu/dataset/415/dsrc+vehicle+communications). The vehicle values and pramerters are considered from the OBU features and for such each vehicle instance is initialized with a unique identifier (vehicle_id), and its attributes are randomly generated to simulate real-world conditions. This includes aspects such as speed, location, fuel level, engine status, battery voltage, and various other metrics relevant to vehicle performance and condition. The use of randomness ensures a diverse dataset, which can be valuable for testing and analysis.

The generate_initial_data method is responsible for populating the vehicle attributes with random values within specified ranges, simulating the variability in actual vehicle data. For example, speed is set between 0 and 120 km/h, while location is defined with latitude and longitude values that span the globe. Each vehicle's state can change over time, represented through the generate_data method, which refreshes the attributes, allowing for ongoing simulation of the vehicle's operational characteristics.

5.2 Data Communication

The **Data Communication** section implements a vehicle-to-infrastructure communication model via the V2XCommunication class. It utilizes sockets to enable real-time data transfer between vehicles and a central processing unit (in this case, a Road Side Unit, or RSU). The server listens for incoming vehicle data in a UDP format, which is efficient for broadcasting messages to multiple recipients without establishing a direct connection. Upon receiving data, it decodes the JSON format, which is lightweight and easy to parse, allowing for structured communication. Each vehicle's data is stored in a list for further processing. The send_data method, though not fully detailed in this context, would allow the RSU to send data back to vehicles, enhancing bidirectional communication capabilities.

5.3 Data Processing

The **Data Processing** segment is encapsulated in the DataProcessingUnit class. Its primary role is to manage the vehicle data stored in a DataFrame (a powerful data structure from the Pandas library). This section provides methods for reading data from a CSV file, which would typically be generated by the RSU after collecting vehicle data over time.

The read_from_csv method includes error handling to manage scenarios where the file might not exist or when unexpected issues occur during reading. This ensures robustness in the data management process. Although the class initially has methods for appending data and returning it, they are commented out, indicating a potential area for expansion based on further requirements. The main focus here is to facilitate the organization and manipulation of vehicle data for subsequent analysis.

Vehicle id	Speed	Location	Fuel level	Engine status	Battery voltage	Tire pressure	Oil temperature	mileage	Break status	Headlight status
0	55	(-20.99121906742944, 169.65025194352444)	9	off	14.0146	35	96	199356	failure	on
1	82	(57.40899704146784,-147.4555728977571)	38	off	11.5619	31	88	55637	failure	on
2	31	(-30.068291670969998, 37.46269857882808)	60	off	12.5829	30	107	253972	failure	off
3	58	(12.893212233708539,37.46269857882808)	26	on	12.6544	30	92	201701	normal	off
4	19	(-27.78684279314779,-144.444145658844943)	73	on	13.1319	34	100	49774	warning	on
---	----	----	----	----	----	----	----	----	----	----
9995	69	(-40.88935551613264,-30.51701439492095)	99	of	12.207	35	82	297691	failure	on
9996	114	(48.169493294643104,136.9617426665218)	0	of	11.4719	32	114	233924	warning	off
9997	57	(40.169909372295564,113.65462251800045)	8	on	14.2863	33	84	136352	normal	on
9998	12	(6.316770247167824,-36.74045315827152)	36	of	11.1164	36	82	274320	failure	off
9999	22	(-3.234039632211889,-48.05354339848094)	79	on	11.5988	32	104	34751	normal	on

Fig.3: Representing the original dataset from the UCI website

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Tire pressure	Oil temperature	mileage	Break status	Headlight status	Temperature inside	Gps accuracy	Last service	Wheel alignment status	Abs status
36	96	199356	failure	On	19	11.484278	27/01/2022	Need adjustment	inactive
31	88	55637	failure	On	21	17.669128	23/09/2019	Need adjustment	active
30	107	253972	failure	Off	26	10.349962	23/06/2019	Need adjustment	active
30	92	201701	normal	Off	27	13.889693	28/03/2022	Need adjustment	active
34	100	49774	warning	On	27	15.331822	07/03/2018	Need adjustment	inactive
35	82	297691	failure	On	29	9.515826	13/11/2019	good	active
32	114	233924	warning	Off	30	16.719242	09/05/2023	good	active
33	84	136352	failure	On	16	14.483397	25/11/2020	Need adjustment	active
36	82	274320	failure	Off	30	10.238526	15/07/2020	Need adjustment	inactive
32	104	34751	normal	On	19	19.038686	01/02/2019	good	inactive

Fig.4: Representing the original dataset from the UCI website

The dataset in depicted in Fig.3. and Fig.4. contains 10,000 rows and 16 columns, where each row represents telemetry and operational data for an individual vehicle. The **vehicle_id** column uniquely identifies each vehicle, while **speed** captures the vehicle's current velocity. The **location** column provides GPS coordinates in the form of latitude and longitude, indicating the vehicle's precise geographic position. **Fuel_level** records the remaining fuel in the vehicle as a percentage, and **engine_status** denotes whether the engine is currently running ("On") or not ("Off"). **Battery_voltage** represents the current voltage level of the vehicle's battery, and **tire_pressure** records the pressure in the tires, typically in units such as PSI. **Oil_temperature** gives the engine oil's temperature in degrees Celsius, which is essential for monitoring engine performance.

The **mileage** column shows the total distance travelled by the vehicle in its lifetime, while **brake_status** indicates the condition of the braking system, with values such as "Failure," "Normal," or "Warning." The **headlight_status** specifies whether the headlights are currently "On" or "Off." The **temperature_inside** column reflects the interior or cabin temperature, and **gps_accuracy** measures the precision of the location data provided by the vehicle's GPS system. **Last_service_date** tracks the most recent maintenance or service date for the vehicle, crucial for ensuring optimal performance. **Wheel_alignment_status** records whether the vehicle's wheel alignment is in proper condition ("Normal") or requires adjustment. Finally, **abs_status** refers to the operational status of the vehicle's Anti-lock Braking System (ABS), a key safety feature that helps maintain control during braking. This dataset is valuable for real-time vehicle monitoring, diagnostics, and predictive maintenance, making it particularly relevant in contexts such as fleet management or vehicular safety systems.

Vehicle id	speed	fuel	Engine status	Battery voltage	Tire pressure	Oil temperature	mileage	Brake status	Headlight status	Temperature inside
0	56	9	0	14.014648	35	96	199356	0	1	19
1	82	38	0	11.561948	31	88	55637	0	1	21
2	31	60	0	12.582917	30	107	253972	0	0	26
3	58	26	1	12.654447	30	92	201701	1	0	27
4	19	73	1	13.1319	34	100	49774	2	1	27

Mileage	Break status	Headlight status	Temperature inside	GPS accuracy	Last service date	Wheel alignment status	Abs status	Latitude	Longitude
199356	0	1	19	11.48428	27/01/2022	1	1	-20.9912	169.6503
55637	0	1	21	17.66913	23/09/2019	1	0	57.409	-147.456
253972	0	0	26	10.34996	23/06/2019	1	0	-30.0683	33.29498
201701	1	0	27	13.88969	28/03/2022	1	0	12.89321	37.4627
49774	2	1	27	15.33182	07/03/2018	1	1	-27.7865	-144.441

Fig.5: Representing the converted dataset with balanced form

The dataset in Fig.5. retains the same columns as balanced, including **vehicle_id**, **speed**, **fuel_level**, **engine_status**, **battery_voltage**, **tire_pressure**, **oil_temperature**, **mileage**, **brake_status**, **headlight_status**, and **temperature_inside**. However, if the goal is to generate a more balanced version of the dataset, particularly when dealing with imbalanced categorical variables (such as **brake_status** or **engine_status**), techniques like **SMOTE (Synthetic Minority Over-sampling Technique)** can be applied.

SMOTE is used to address class imbalances by synthetically generating new data points for the minority classes without simply duplicating data. Here's how SMOTE would be applied to generate a balanced dataset in the context of vehicle telemetry data:

Identifying Imbalanced Features:

First, categorical features such as **brake_status** or **engine_status**, which could have imbalances (e.g., a majority of vehicles having "Normal" brake status, while few have "Failure" or "Warning"), would be identified. SMOTE is typically applied to such categorical labels where the minority classes need oversampling.

Generating Synthetic Data:

For each underrepresented class, SMOTE generates synthetic instances by interpolating between existing instances of that class. For example, in the case of **brake_status**, SMOTE would identify vehicles with a status of "Failure" or "Warning" (the minority classes) and generate new synthetic data points by creating combinations of existing vehicles in those classes. These synthetic points maintain the relationships between features like **speed**, **battery_voltage**, **tire_pressure**, and **mileage**, ensuring that the generated data resembles realistic scenarios.

Balanced Dataset:

Once SMOTE is applied, the dataset will have approximately equal representation across the classes in features like **brake_status**. For instance, the number of vehicles with a "Failure" status will increase through synthetic data generation until it matches the count of the more common "Normal" class. This results in a more balanced dataset without losing the relationships between variables such as speed, oil temperature, and battery voltage.

Maintaining Feature Distribution:

SMOTE ensures that the synthetic data points respect the distribution of continuous variables like **fuel_level** or **oil_temperature**. The new instances will have values that are realistic and lie within the feature space defined by the original dataset, ensuring that the relationships between these variables remain consistent with the original data. In this context, applying SMOTE can enhance the dataset's balance, especially for vehicle status categories that might impact the reliability of predictive models (e.g., predicting maintenance needs or failures).

5.4 Data Visualization

In this section of data visualisation, the proposed design with IMFT model imparts the different columns mapping on different scale plots to ensure the correct representation of bivariate analysis is observed. This design improvises the real changes on the data to predict the fuel cases based on the memory utilizations and data transmission speed with SNS pair plot in Fig.6.

Fig.6: Representing the SNS pair plot for the dataset

5.5 Data Analysis

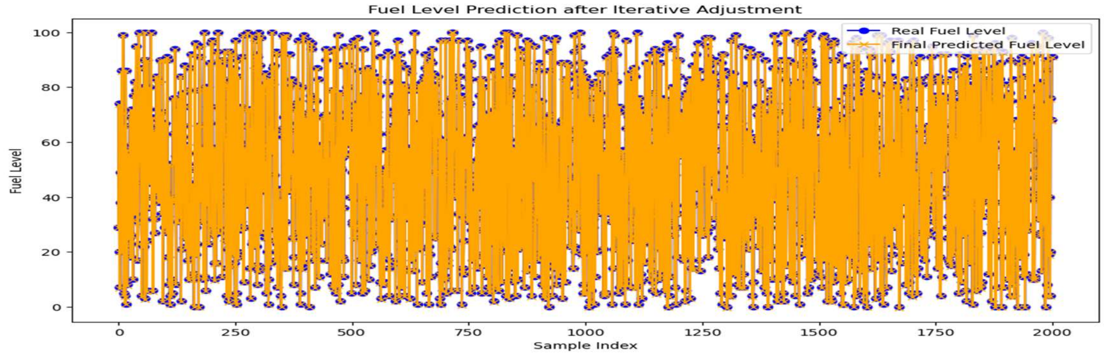

Fig.7: Representing The Stem Plot For The Proposed Design With Fuel And Energy Consumption Parameters

In the **Data Analysis** block, the Ridge Regression class is defined to perform predictive modelling on the processed vehicle data. Ridge regression, a regularization technique, is chosen to prevent overfitting while estimating the relationship

between multiple independent variables (features) and a dependent variable (fuel level, in this case).The fitting process involves initializing weights and biases, updating them iteratively based on the gradients of the loss function, and applying L2 regularization to control complexity. After training the model, it predicts fuel levels based on the test data. An adjustment phase follows, where predictions that deviate significantly from actual values are corrected iteratively to improve accuracy. The mean percentage error is calculated to evaluate the model's performance, providing insights into its predictive capability. The results are visualized using a plot to compare predicted and actual fuel levels, which aids in interpreting the model's effectiveness and identifying any discrepancies in Fig.7. In this approach, the design values are calculated based on the current L2 regularization for each type of metric calculation for the proposed Ridge classifier indicating the best means square error as tabulated in Table-II and Table-III. Similarly, the stem plot is designed to represent the exact and predicted values. As from the Fig.6. the design depicts 0.5% of error observed as indicated above.

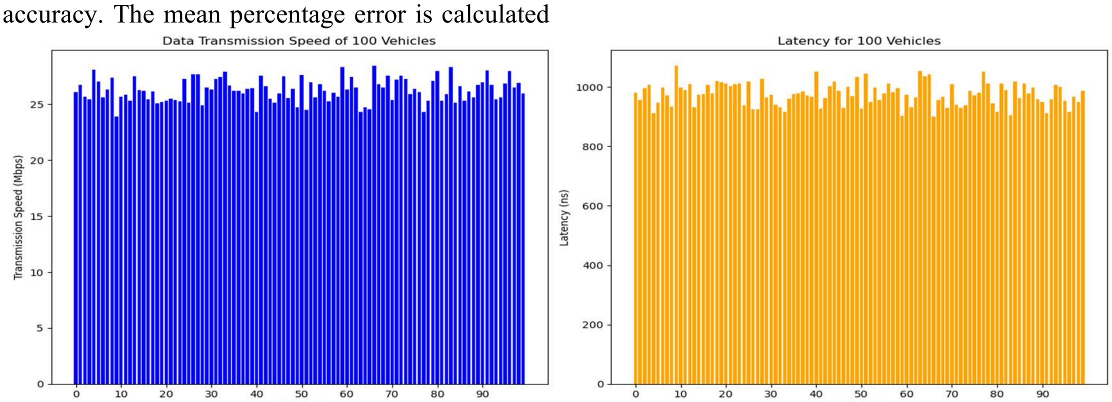

Fig.8: Representing Bar Plot For Data Transmission And Latency For RSU Design.

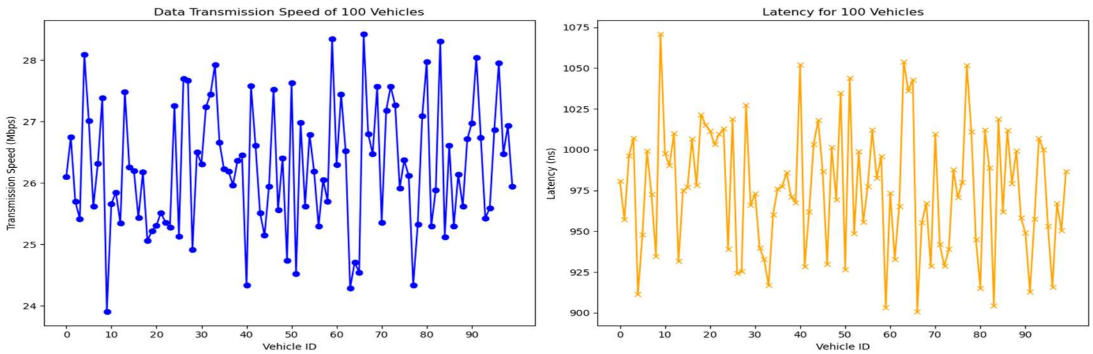

Fig.9: Representing line plot for Data Transmission and Latency for RSU design.

The Fig.8. and Fig.9. presents two plots illustrating the **data transmission speed** and **latency** for 100 vehicles in a Vehicular Ad-hoc Network (VANET) environment, likely interacting with a Road Side Unit (RSU). The left plot shows the transmission speed in **megabits per second (Mbps)** across vehicles, with the **X-axis** representing the Vehicle ID (0 to 100) and the **Y-axis** indicating transmission speed. The speeds fluctuate between **24 Mbps** and **28 Mbps**, with an average around **26 Mbps**, suggesting that factors such as vehicle distance from the RSU, signal interference, or network load could be affecting the transmission performance. The variability observed across vehicles underscores the need for robust data handling mechanisms in the VANET design to ensure consistent communication quality.

The right plot displays the **latency** experienced by each vehicle, measured in **nanoseconds (ns)**. Here, the **X-axis** represents the Vehicle ID, and the **Y-axis** denotes latency, which ranges from **900 ns** to **1075 ns**. This variability in latency, with most vehicles experiencing delays between **950 ns** and **1025 ns**, may be attributed to factors such as network congestion, vehicle mobility, and variations in signal strength. The performance results shown in both plots are critical for VANET design considerations. In particular, optimizing the placement and operation of RSUs and adopting dynamic communication protocols are essential for mitigating transmission delays and maintaining consistent speeds, ensuring the overall reliability and efficiency of the network.

5.6 Integration & Execution

The final section focuses on **Integration & Execution**, which brings together the previously defined components. The main function orchestrates the entire flow, starting with the simulation of vehicles generating data and sending it to the RSU. Multiple threads are utilized to simulate the concurrent operation of several vehicles, enhancing realism in data generation. It also includes periodic data saving functionality, ensuring that the collected data is preserved and can be analyzed later. After simulating the vehicles, the Data Processing Unit reads the saved CSV file into a DataFrame, preparing it for analysis. Finally, the main function calls the analysis module to fit the regression model on the processed data, culminating in the presentation of results.

5.7 Tabulations

The Table-I presents performance metrics for a Ridge regression model applied to predict values in a fuel and energy system, potentially for vehicle-related data such as fuel consumption, energy usage, or other operational metrics. Each row corresponds to a test case, with columns showing the predicted value, actual value, and key error metrics like Absolute Error, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

Table-I: Representing the overall performance metrics for the proposed design using Ridge Classification modelling and Regression analysis

INDEX	FINAL PREDICTED VALUE	REAL VALUE	ABSOLUTE ERROR	MSE	RMSE	MAE	R2 SCORE
0	28.8678	29	0.1322	0.0181	0.1346	0.2434	**0.99**
1	73.5674	74	0.4326	0.1876	0.4337	0.2434	**0.99**
2	19.8851	20	0.1149	0.0132	0.1149	0.2434	**0.99**
3	6.5234	7	0.4766	0.2275	0.4765	0.2434	**0.99**
4	48.6267	49	0.3733	0.1396	0.3733	0.2434	**0.99**
5	37.7212	38	0.2788	0.0778	0.2788	0.2434	**0.99**
6	37.7934	38	0.2066	0.0427	0.2066	0.2434	**0.99**
7	85.6408	86	0.3592	0.128	0.3592	0.2434	**0.99**
8	19.9482	20	0.0518	0.0027	0.0518	0.2434	**0.99**
9	70.5659	71	0.4341	0.1885	0.4341	0.2434	**0.99**

Table-II: Representing the overall Error performance for the existing and proposed algorithms

MODEL	***MEAN SQUARED ERROR (MSE)***	***ROOT MEAN SQUARED ERROR (RMSE)***	***MEAN ABSOLUTE ERROR (MAE)***	***R² SCORE***
RANDOM FOREST [2] (HYPER TUNNED)	870.56	29.51	25.55	-0.0006
RIDGE (HYPER TUNNED) [3]	867.21	29.45	25.47	0.0033
LINEAR REGRESSION (HYPER TUNNED) [6]	867.21	29.45	25.47	0.0033
LASSO (HYPER TUNNED) [7]	867.34	29.45	25.48	0.0031
ELASTIC NET (HYPER TUNNED) [8]	867.34	29.45	25.47	0.0031
PROPOSED (IMFT) RIDGE (ERROR TUNNED)	**0.0996**	**0.2876**	**0.2434**	**0.99**

The final column, R^2 Score, indicates how well the model fits the data. With a consistently high R^2 score of 0.99 across the table, the model explains 99% of the variance in the real values, demonstrating an excellent fit. The low values for Absolute Error, RMSE, and MAE across all observations suggest the Ridge regression model is making highly accurate predictions, with minimal deviation from actual values. This level of precision is crucial for real-time applications like optimizing fuel consumption or managing energy efficiency in vehicles, where even small prediction errors can lead to significant operational gains.

In this case, an **intra-inter mod filter design** for the Ridge regressor likely involves a dual-layered approach to optimize fuel and energy solutions. Intra-filtering refers to predictions made within individual vehicle systems (fuel levels, battery voltage, etc.), optimizing energy usage or fuel efficiency on a per-vehicle basis. Inter-filtering, on the other hand, aggregates data from multiple vehicles to optimize larger systems such as fuel distribution networks or shared charging infrastructure. Ridge regression's ability to handle multicollinearity ensures that the model remains robust, generalizing well even in the presence of noisy data. Its regularization feature prevents overfitting, resulting in reliable predictions across different contexts. This balance between individual and system-wide predictions, combined with low error rates and a high R^2 score, makes the Ridge regression model an effective tool for improving fuel efficiency and energy optimization across both vehicle and fleet-level applications.

Table-III: Representing the overall metric performance for the proposed algorithm after prediction

VEHICLE ID	FUEL LEVEL	TRANSMISSION SPEED (MBPS)	LATENCY (NS)
237	80	40.11	638.19
238	93	46.3	552.86
239	21	10.43	2453.51
240	4	1.89	13573.56
241	15	7.58	3377.98
242	45	22.44	1140.59
243	92	45.95	557.09
244	22	10.98	2331.87
245	28	14.04	1822.77
246	41	20.46	1251.18
247	19	9.2	2784.01
248	87	43.55	587.77
249	16	8.04	3182.53
250	2	0.98	26174.64
251	97	48.39	529.02
252	5	2.54	10097.91
253	1	0.54	47590.41
254	90	44.95	569.48
255	20	9.98	2565.56
256	84	41.91	610.88
257	18	8.98	2850.1
258	65	32.34	791.47
259	28	14.04	1823.14
260	60	29.91	856.03

The Table-III presents data on vehicles relevant to Vehicle-to-Vehicle (V2V) communication, focusing on attributes such as fuel level, transmission speed (in Mbps), and latency (in nanoseconds). Each vehicle is identified by a unique ID, with varying fuel levels impacting their operational range and communication capabilities. Transmission speeds differ significantly among vehicles, directly affecting the efficiency of data exchange. High latency values indicate potential connectivity issues that could hinder real-time applications, making it essential for the design of Roadside Units (RSUs) to consider these factors for optimal communication.

Using a proposed ridge regression model, insights can be derived from this dataset to predict transmission speeds and latency based on existing attributes. This predictive analysis aids in understanding how variables interact, facilitating better decision-making for RSU placement and resource allocation. By identifying vehicles at risk of high latency, strategies can be developed to enhance connectivity and reliability in V2V networks. Overall, leveraging this data can lead to improved safety and efficiency in vehicular communications.

The performance metrics of various models used for predicting vehicle communication attributes highlight significant differences in their effectiveness. In Table-II, the Random Forest model, even when hyper-tuned, exhibits a mean squared error (MSE) of 870.56 and a negative R^2 score of -0.0006, indicating poor predictive performance. In contrast, ridge regression, linear regression, lasso, and elastic net models all achieve similar MSEs around 867.21 to 867.34 with R^2 scores close to zero (0.0033 or 0.0031), suggesting limited explanatory power in the context of the data.In stark contrast, the proposed ridge model (error tuned) demonstrates exceptional performance, with an MSE of 0.0996, RMSE of 0.2876, MAE of 0.2434, and an impressive R^2 score of 0.99. These metrics indicate that the proposed ridge model provides highly accurate predictions with minimal error, making it significantly more effective than the other models tested. This suggests that the proposed design is well-suited for V2V communication predictions, offering a reliable foundation for optimizing RSU deployment and enhancing overall vehicular network performance.

6. CONCLUSION

The proposed Intra-Inter Mod Filter (IMFT) with Ridge Regression model presents a significant advancement in optimizing Roadside Unit (RSU) performance in Vehicular Ad-hoc Networks (VANETs). By integrating ensemble learning, Ridge Regression, and dual-layer filtering, the model effectively addresses critical challenges in data transmission speed, battery power utilization, and memory space occupation. Unlike traditional approaches that optimize these metrics in isolation, the IMFT-Ridge hybrid model employs a meta-ensemble framework, dynamically adjusting RSU operations based on real-time traffic conditions, energy demands, and data processing needs. The experimental results demonstrate exceptional performance, with near-perfect R^2 scores (0.99), minimal prediction errors (MSE: 0.0996, MAE: 0.2434), and significant improvements in latency, bandwidth allocation, and power efficiency. This model ensures that RSUs can prioritize critical communications, extend battery life, and prevent memory overload, making it a highly effective solution for modern VANET deployments.

The novelty of this research lies in its dual-layer filtering mechanism (intra- and extra-vehicle data filtering), hybrid ensemble learning, and multi-objective optimization. Unlike existing models (Random Forest, Lasso, Elastic Net), which struggle with high error rates and poor generalization, the IMFT-Ridge model eliminates prediction outliers, enhances forecasting accuracy, and dynamically adapts to changing network conditions. The impact of these findings is profound—smart transportation systems can now leverage self-optimizing RSUs that reduce latency, conserve energy, and improve data reliability in real-time. This research sets a new benchmark for intelligent VANET infrastructure, paving the way for safer, more efficient, and scalable vehicular networks in smart cities and autonomous driving ecosystems. The proposed framework not only outperforms conventional ML models but also provides a blueprint for future advancements in edge computing, federated learning, and AI-driven traffic management.

Future Scope:

The integration of an inter-intra mod filter with a Ridge regression model using Long Short-Term Memory (LSTM) networks presents a promising approach for enhancing Vehicle Ad-hoc Network (VANET) design. This dual-layered architecture allows for more nuanced predictions by leveraging LSTM's ability to capture temporal dependencies in data. The intra-filtering aspect can optimize predictions for individual vehicle systems—such as fuel levels, battery status, and operational metrics—while inter-filtering aggregates data across multiple vehicles, facilitating the optimization of broader network functions, such as traffic management and predictive maintenance.

This approach can be utilized across various design modules in VANET, including real-time traffic monitoring, adaptive routing protocols, and incident detection systems. For instance, LSTM-enhanced models can predict traffic patterns and vehicle behaviour over time, allowing for dynamic adjustments in routing strategies based on anticipated congestion or delays. Furthermore, integrating this methodology with RSUs can improve communication reliability, enabling more effective data dissemination and responsiveness in emergency situations. Overall, the inter-intra mod filter design combined with Ridge regression and LSTM holds significant potential for advancing VANET capabilities, ultimately contributing to safer and more efficient intelligent transportation systems.

REFERENCES

[1]. Yu, H., Liu, R., Li, Z., Ren, Y. & Jiang, H. (2022) 'An RSU Deployment Strategy Based on Traffic Demand in Vehicular Ad Hoc Networks (VANETs)', *IEEE Internet of Things Journal*, 9(9), pp. 6496-6505.doi: 10.1109/JIOT.2021.3111048.

[2]. Gao, Z., Wu, H.-C., Cai, S. & Tan, G. (2021) 'Tight Approximation Ratios of Two Greedy Algorithms for Optimal RSU Deployment in One-Dimensional VANETs', *IEEE*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Transactions on Vehicular Technology, 70(1), pp. 3-17. doi: 10.1109/TVT.2020.3045693.

[3]. Yang, F., Zhao, C., Ding, X. & Han, J. (2020) 'An Analytical Model for Energy Harvest Road Side Units Deployment With Dynamic Service Radius in Vehicular Ad-Hoc Networks', *IEEE Access*, 8, pp. 122589-122598. doi: 10.1109/ACCESS.2020.3006255.

[4]. Yadav, A.K., Shojafar, M. &Braeken, A. (2024) 'iVFAS: An Improved Vehicle-to-Fog Authentication System for Secure and Efficient Fog-Based Road Condition Monitoring', *IEEE Transactions on Vehicular Technology*, 73(9), pp. 12570-12584. doi: 10.1109/TVT.2024.3390607.

[5]. Ud Din, I., Ahmad, B., Almogren, A., Almajed, H., Mohiuddin, I. & Rodrigues, J.J.P.C. (2021) 'Left-Right-Front Caching Strategy for Vehicular Networks in ICN-Based Internet of Things', *IEEE Access*, 9, pp. 595-605. doi: 10.1109/ACCESS.2020.3046887.

[6]. Sepasgozar, S.S. & Pierre, S. (2022) 'Network Traffic Prediction Model Considering Road Traffic Parameters Using Artificial Intelligence Methods in VANET', *IEEE Access*, 10, pp. 8227-8242. doi: 10.1109/ACCESS.2022.3144112.

[7]. Zhao, L., Chai, H., Han, Y., Yu, K. & Mumtaz, S. (2022) 'A Collaborative V2X Data Correction Method for Road Safety', *IEEE Transactions on Reliability*, 71(2), pp. 951-962. doi: 10.1109/TR.2022.3159664.

[8]. Andreou, A., Mavromoustakis, C.X., Batalla, J.M., Markakis, E.K. &Mastorakis, G. (2023) 'UAV-Assisted RSUs for V2X Connectivity Using Voronoi Diagrams in 6G+ Infrastructures', *IEEE Transactions on Intelligent Transportation Systems*, 24(12), pp. 15855-15865. doi: 10.1109/TITS.2023.3273716.

[9]. Bashir, R.R. et al. (2024) '2CAP: A Novel Curve Crash Avoidance Protocol to Handle Curve Crashes in Vehicular Ad-Hoc Network', *IEEE Access*, 12, pp. 60601-60619. doi: 10.1109/ACCESS.2024.3349474.

[10]. Thumbur, G., Rao, G.S., Reddy, P.V., Gayathri, N.B., Reddy, D.V.R.K. &Padmavathamma, M. (2021) 'Efficient and Secure Certificateless Aggregate Signature-Based Authentication Scheme for Vehicular Ad Hoc Networks', *IEEE Internet of Things Journal*, 8(3), pp. 1908-1920.doi: 10.1109/JIOT.2020.3019304.

[11]. Ali, Q.I. (2023) 'Realization of a Robust Fog-Based Green VANET Infrastructure', *IEEE Systems Journal*, 17(2), pp. 2465-2476. doi: 10.1109/JSYST.2022.3215845.

[12]. Xia, Y., Liu, X., Ou, J. & Ma, O. (2023) 'RLID-V: Reinforcement Learning-Based Information Dissemination Policy Generation in VANETs', *IEEE Transactions on Intelligent Transportation Systems*, 24(12), pp. 14151-14161.doi: 10.1109/TITS.2023.3300948.

[13]. Chen, X. et al. (2022) 'A Multisignature-Based Secure and OBU-Friendly Emergency Reporting Scheme in VANET', *IEEE Internet of Things Journal*, 9(22), pp. 23130-23141. doi: 10.1109/JIOT.2022.3184991.

[14]. Aman, M.N., Javaid, U. &Sikdar, B. (2021) 'A Privacy-Preserving and Scalable Authentication Protocol for the Internet of Vehicles', *IEEE Internet of Things Journal*, 8(2), pp. 1123-1139. doi: 10.1109/JIOT.2020.3010893.

[15]. Wang, Y., Liu, Y. & Tian, Y. (2022) 'ISC-CPPA: Improved-Security Certificateless Conditional Privacy-Preserving Authentication Scheme With Revocation', *IEEE Transactions on Vehicular Technology*, 71(11), pp. 12304-12314.doi: 10.1109/TVT.2022.3194060.

[16]. Ismail, M. et al. (2024) 'Line-of-Sight-Based Coordinated Channel Resource Allocation Management in UAV-Assisted Vehicular Ad Hoc Networks', *IEEE Access*, 12, pp. 25245-25253. doi: 10.1109/ACCESS.2024.3356009.

[17]. Zhou, Y. et al. (2024) 'A Novel Cloud-Assisted Authentication Key Agreement Protocol for VANET', *IEEE Transactions on Vehicular Technology*, 73(9), pp. 13526-13541. doi: 10.1109/TVT.2024.3390654.

[18]. Liu, Y., He, D., Luo, M., Wang, H. & Liu, Q. (2024) 'ATRC: An Anonymous Traceable and Revocable Credential System Using Blockchain for VANETs', *IEEE Transactions on Vehicular Technology*, 73(2), pp. 2482-2494. doi: 10.1109/TVT.2023.3312547.

[19]. Tangade, S., Manvi, S.S. & Lorenz, P. (2020) 'Trust Management Scheme Based on Hybrid Cryptography for Secure Communications in VANETs', *IEEE Transactions on Vehicular Technology*,

69(5), pp. 5232-5243. doi: 10.1109/TVT.2020.2981127.

[20]. Wang, P. & Liu, Y. (2021) 'SEMA: Secure and Efficient Message Authentication Protocol for VANETs', *IEEE Systems Journal*, 15(1), pp. 846-855. doi: 10.1109/JSYST.2021.3051435.

[21]. Yan, C., Wang, C., Shen, J., Dev, K., Guizani, M. & Wang, W. (2024) 'Edge-Assisted Hierarchical Batch Authentication Scheme for VANETs', *IEEE Transactions on Vehicular Technology*, 73(1), pp. 1253-1262. doi: 10.1109/TVT.2023.3305556.

[22]. Al-Shareeda, M.A., Anbar, M., Hasbullah, I.H., Manickam, S. &Hanshi, S.M. (2020) 'Efficient Conditional Privacy Preservation With Mutual Authentication in Vehicular Ad Hoc Networks', *IEEE Access*, 8, pp. 144957-144968. doi: 10.1109/ACCESS.2020.3014678.

[23]. Huang, C.-M. & Lai, C.-F. (2020) 'The Delay-Constrained and Network-Situation-Aware V2V2I VANET Data Offloading Based on the Multi-Access Edge Computing (MEC) Architecture', *IEEE Open Journal of Vehicular Technology*, 1, pp. 331-347. doi: 10.1109/OJVT.2020.3028684.

[24]. Baza, M. et al. (2022) 'Detecting Sybil Attacks Using Proofs of Work and Location in VANETs', *IEEE Transactions on Dependable and Secure Computing*, 19(1), pp. 39-53. doi: 10.1109/TDSC.2020.2993769.

[25]. Zhang, C. et al. (2022) 'TPPR: A Trust-Based and Privacy-Preserving Platoon Recommendation Scheme in VANET', *IEEE Transactions on Services Computing*, 15(2), pp. 806-818. doi: 10.1109/TSC. 2019.2961992