



RNN-GRU Powered Multi-Level Traffic Classification in Smart Cities with IoV Sensor Data

Meeruvani Keerthi¹, SK Physuddin²

¹M. Tech Student, ²Assistant Professor, ^{1,2}Department of Electronics and Communications Engineering, ^{1,2}Mother Teresa Institute of Science and Technology, Sathupally, Khammam 507303, Telangana

ABSTRACT

Recent studies show that urban traffic congestion costs the global economy nearly \$300 billion annually, with delays in smart cities increasing by over 15% in the last five years despite Internet of Vehicles (IoV) adoption. This growing challenge underscores the need for intelligent, data-driven traffic classification systems to optimize flow and reduce congestion. However, existing IoV-based traffic prediction models often suffer from low adaptability to diverse city conditions and limited handling of complex, high-dimensional sensor data, leading to inaccurate classifications and suboptimal traffic control decisions. In this work, we present a novel IoV-driven Smart City Traffic Classification framework designed to accurately categorize traffic conditions into High, Low, Medium, and Very High levels using multi-source sensor data. The dataset incorporates diverse features, including City, Vehicle Type, Weather, Economic Condition, Day of Week, Hour of Day, Speed, Is Peak Hour, Random Event Occurred, Energy Consumption, Traffic Density, and Traffic Density Category. Our approach begins with robust data preprocessing, including missing value handling, normalization, and categorical encoding to ensure clean and standardized inputs. The processed data is then split into training and testing sets to enable unbiased performance evaluation. We benchmark traditional machine learning classifiers such as K-Nearest Neighbors (KNN) and Logistic Regression, before introducing our proposed Recurrent Neural Network with Gated Recurrent Unit (RNN-GRU) model.

This architecture captures sequential dependencies in temporal and contextual traffic data, enabling more accurate classification under dynamic urban scenarios. Comparative analysis shows that the proposed RNN-GRU significantly outperforms baseline methods in precision, recall, and F1-score, offering a scalable and adaptive solution for real-time traffic management in smart cities.

Keywords: Traffic Classification, Multi-Level Classification, Urban Mobility Management, Smart Infrastructure, IoT in Smart Cities, Sensor Data Analytics

1. INTRODUCTION

Urban traffic congestion has emerged as one of the costliest challenges for smart cities worldwide. In 2019, traffic delays cost Indian cities an estimated USD 22 billion, projected to escalate to USD 37 billion by 2030 if unaddressed. Globally, intelligent traffic management systems are expected to deliver savings of around USD 277 billion by 2025 by reducing congestion and emissions. In cities like Milan, congestion charges have led to a sustained reduction of vehicle entries by up to 33–34%, boosting average transport speeds and easing gridlock across central zones. Beyond economic losses, traffic congestion significantly affects safety and emissions. Cities leveraging smart traffic solutions have reported around a 25–35% decrease in road accidents, demonstrating the dual benefit of optimized mobility and improved public safety. Adaptive signal networks in urban environments—using real-time sensor data and AI have reduced idling time at intersections by approximately 40% and travel times by



25%, according to pilot studies in Pittsburgh and other demo cities. This shows how modern data-driven systems can dramatically transform traffic outcomes.

India faces particularly acute congestion: its road network is the largest globally and carries over 85% of passenger traffic, yet average urban speeds remain very low—typically under 20 km/h, even in major centers like Delhi, Chennai, Bengaluru, and Hyderabad. Rapid vehicle growth outpaces infrastructure expansion—as in Bangalore, congestion can cost up to 5% of the city's economic output, exacerbated by high road fatality rates (India leads globally in absolute traffic deaths). These realities underscore the urgent need for smarter, data-informed traffic solutions across densely populated and fast-growing urban regions.

2. LITERATURE SURVEY

Khan et al. [6] proposed a self-adaptive real-time traffic clearance algorithm at intersections. They applied YOLOv3 for object detection on video frames, coupled with centroid tracking to monitor vehicle trajectories and detect wrong-way movements. The system computed green-light durations based on vehicle counts, road width, and crossing time. It achieved average detection accuracy of 88.43% and tracking accuracy of 90.45% for wrong-turn violations. The method also adjusted signal phases dynamically to improve traffic safety and intersection throughput. It relied on manual feature derivation from detection output and lacked automated feature selection, increasing complexity and limiting scalability. Alam et al. [7] developed an adaptive street-light control system using machine learning algorithms to optimize energy usage. The system ingested historical and ambient light lux readings, applied classification to predict lighting needs based on environmental and human activity cues. They evaluated performance

using classification accuracy and confusion matrix insights. The approach reduced human involvement and integrated seamlessly with smart grid goals. It involved heavy computation in feature extraction from temporal light patterns, leading to high latency in real-time response. Miao and Liao [8] explored IoT-based traffic prediction using machine learning models for smart cities. They utilized sensor data from cameras, GPS devices, and environmental monitors to model traffic flow prediction. Their framework combined historical and real-time inputs to forecast congestion points, and evaluated models based on accuracy and response time improvements. They demonstrated significant gains over traditional traffic management approaches. It performed feature extraction primarily using spatial data fusion without proper grading or weighing of heterogeneous features, resulting in reduced prediction robustness across varying urban contexts. Talaat et al. [9] proposed a Smart Traffic Management System (STMS) that used YOLOv11 for real-time vehicle detection, paired with dynamic flow optimization in urban environments. The system detected vehicles with a mean average precision (mAP) of 92.4% and intersection-over-union (IoU) score of 0.85, enabling accurate localization and classification. It processed video frames from roadside cameras and computed traffic density metrics to adjust signal timing adaptively. It integrated AI-driven analytics for rerouting and congestion mitigation based on real-time counts. Feature extraction relied on raw count outputs without discriminative selection, causing performance drop in mixed-class traffic scenes.

Kapoor et al. [10] proposed Smart-VDCTC, a smart vehicular detection, classification, and traffic control system tailored for Indian urban settings. The framework combined ML and DL



classifiers to identify vehicle types and congestion categories from camera feeds and vehicular sensors, then adjusted traffic signal phases dynamically. It processed multi-modal inputs visual and sensor streams to classify traffic into discrete congestion labels. It demonstrated notable improvements in intersection throughput under typical Delhi-style heterogeneous flow. Feature selection did not include contextual prioritization (e.g., time-of-day weighting), increasing computational complexity and reducing real-time responsiveness. Ismail et al. [11] investigated multiple convolutional neural network (CNN) models—including SSD-ResNet50, SSD-ResNet152, YOLOv5, and YOLOv8—for vehicle classification in smart-city traffic scenes. Researchers collected a dataset of 1,324 labeled images across five vehicle classes and compared models based on classification accuracy and inference time. They identified the best-performing architecture for deployment using metrics like precision, recall, and latency. They also discussed trade-offs between model size and speed for resource-constrained edge systems. They used uniform feature extraction pipelines across models without feature grading or selection, which reduced performance consistency across varying environments. Ma et al. [12] proposed an IoT-based electronic traffic signal acquisition system featuring three core modules: video acquisition & transmission (VAT), video processing (VP), and information processing. They designed a multi-task CNN called AM-MMFF-GoogleNet incorporating attention mechanisms and multimodal feature fusion for robust extraction of traffic parameters from live camera feeds. The system transmitted vehicle count, classification, and movement parameters to the ITS control platform. They evaluated it on urban scenarios and confirmed improved parameter accuracy

and decision latency. The fusion network involved heavy multi-modal feature blending without proper feature selection, yielding increased computation load and potential latency in real-time deployment. Al Asmari et al. [13] proposed a Conjecture Interaction Optimization Model within an Intelligent Transportation System (ITS) that applied reciprocated multi-instance convergence learning (RMICL). The model integrated multi-instance learning (MIL) to differentiate reliable from erroneous vehicle-terminal interactions. It optimized routing and avoidance decisions via forward/reverse interaction reciprocation to minimize delay and improve forecast accuracy. The framework dynamically weighted terminal proximity and traffic density for route refinement across ITS deployments. The model lacked a formal feature grading or selection mechanism within MIL, leading to potential misclassification when noisy interaction data diluted critical signals. Paul et al. [14] proposed an AI-powered traffic optimization solution for Srinagar Smart City that combined single-stage target detection (neural network) with live traffic flow modeling and accident-site forecasting. It used traffic volume predictions and signal adjustment algorithms to reduce congestion and improve travel time efficiency. The system identified accident-prone zones and optimized intersections to manage dynamic traffic peaks. It leveraged environmental and temporal cues for decision thresholds. It relied on uniform transformation of raw detections without feature selection based on contextual importance, which increased computational load in live operations. Qaffas [15] introduced a distributed IoT communication architecture enabling AI-driven traffic optimization across smart cities. The design used edge and cloud coordination with sensor boards and local gateways to manage real-time traffic



density data. It applied distributed ML load to balance processing between edge devices and the cloud cluster. It supported traffic forecasting, grid-based signal control, and adaptive routing through localized analytics. It extracted features from heterogeneous sensor types without contextual grading or prioritization, causing inconsistent performance under varying traffic scenarios (especially peak vs off-peak).

Wang et al. [16] proposed a smart city transportation planning strategy that used decision-tree algorithms for traffic planning and congestion mitigation. It selected key variables such as road type, vehicle flow, and temporal slot to train trees that generated classification rules for traffic congestion bands. The system evaluated routes and junction configurations to recommend preventive infrastructure adjustments. It included impact assessment of time-of-day and corridor geometry factors. The decision tree method generated many low-importance splits due to redundant or irrelevant features, increasing computational complexity and reducing interpretability in urban-scale deployment. Moraga et al. [17] introduced an AI-driven framework that integrated unmanned aerial vehicles (UAVs) and Internet of Things (IoT) sensors with a large language model (LLM), specifically Gemini-2.0-Flash. The system extracted real-time traffic density and CO₂ emission parameters from induction-loop data and drone-based aerial observations within the SUMO traffic simulator. The LLM dynamically adjusted vehicle speeds to minimize congestion and emissions across multiple urban scenarios. It delivered reductions of up to 82% in congestion and 75% in CO₂ emissions compared with baseline cases. It processed raw sensor and aerial streams without feature grading, increasing computation complexity and reducing model interpretability. Ali et al.

[18] presented the Traffic Control System Process for Pollution Control (TCSP-PC) using IoT sensors and correlated capsule networks for emission-based vehicle classification. The system categorized vehicles as polluting or eco-friendly and enforced traffic control regulations in real time. It achieved +15.36% improvement in classification rate and +121.5 vehicles/km traffic control efficiency over baseline models. The pipeline delivered a 9.77% faster data analysis rate and reduced per-instance processing time by 2.06 seconds. It lacked a mechanism for feature selection within the capsule network pipeline, causing grading issues when emission-behavior features conflicted across mixed traffic types. Mohanty et al. [19] reviewed real-time traffic monitoring solutions combining AI and computer vision under the Internet of Vehicles (IoV) paradigm. The chapter included multiple convolutional and vision-based models that extracted features from camera feeds and telemetry streams to classify traffic conditions. It evaluated inference accuracy, latency trade-offs, and edge deployment constraints across deployment platforms. It discussed performance differences between edge and cloud-based processing for smart-city traffic detection. It applied uniform feature extraction across environments without contextual feature selection, leading to inconsistent accuracy under varied lighting and weather contexts. Sridevi et al. [20] proposed an AI-driven traffic monitoring system designed for 6G-enabled smart cities that detected congestion and performed route optimization in real time. It combined deep classification models with real-time telemetry streams for dynamic decision-making. The system supported latency-sensitive route updates and congestion flags using signal and camera inputs. It targeted next-generation networks to support ultra-fast traffic sensing. The model performed high-



complexity feature fusion without prioritizing high-impact features, resulting in lower real-time performance under constrained computing budgets.

Tshilongamulenzhe et al. [21] proposed a Fusion-Based Intelligent Congestion Management (FB-ICM) algorithm for on-road traffic optimization in smart cities. The method combined data from heterogeneous sources such as inductive loops, video surveillance, and IoT sensors to detect congestion in real time. It used a multi-layer decision engine to integrate vehicle count, speed, and lane occupancy features before triggering adaptive control actions. The fusion process improved detection robustness across diverse traffic conditions. The fusion pipeline lacked adaptive feature weighting, which reduced efficiency in scenarios where certain sensor modalities dominated the prediction signal. Raju et al. [22] introduced Cascade Drive, a unified deep-learning framework for multi-featured detection and control in autonomous electric vehicles operating on unstructured roadways. The framework merged object detection, lane tracking, and obstacle avoidance into a sequential decision pipeline powered by convolutional and recurrent layers. It processed multi-modal sensor inputs including LiDAR, radar, and RGB camera feeds, enabling both perception and motion planning. The approach improved safety and adaptability under challenging terrain. Feature extraction from all sensor streams was uniformly processed without relevance-based pruning, which increased computational burden and reduced inference speed. Revathy et al. [23] developed AI-powered prediction models for urban traffic optimization using a combination of historical flow data and real-time IoT sensor inputs. Their architecture applied regression and deep learning modules to forecast congestion levels and travel times for specific urban corridors. The system demonstrated the

potential to enhance proactive signal scheduling and rerouting strategies. They tested the models in varying traffic density scenarios to assess scalability. The prediction pipeline lacked explicit feature grading, resulting in degraded performance when low-impact attributes were overrepresented in the training process. Chen et al. [24] presented a deep-learning-based traffic classification approach for 5G aerial computing networks. The model leveraged aurally collected traffic footage and network telemetry data to classify traffic patterns for optimized aerial-ground coordination. It integrated spatial-temporal feature extraction modules and attention layers to improve classification accuracy in high-density regions. The system was designed for ultra-low-latency edge computing deployments. The multi-branch feature extractor performed exhaustive computation on all attributes without prioritization, leading to higher latency in resource-constrained environments. Kumari et al. [25] proposed an enhanced traffic surveillance system integrating intelligent vehicle classification with standard monitoring infrastructure. The model extracted multi-scale visual features from surveillance feeds to classify vehicle types in heterogeneous traffic streams. It aimed to improve enforcement, tolling, and statistical traffic analysis in urban areas. They incorporated an adaptive pre-processing stage to mitigate environmental distortions. Drawback: The feature extraction process did not dynamically adjust for changing ambient conditions, leading to reduced classification performance in low-visibility situations.

3. PROPOSED SYSTEM

This work introduces a hybrid IoV-driven Smart City Traffic Classification algorithm that uniquely integrates comprehensive feature preprocessing, multi-stage classification, and a sequence-aware RNN-GRU architecture, a combination not



found in existing surveys. Unlike prior works that either focus on isolated feature engineering or rely solely on static classifiers, our method employs an end-to-end pipeline where context-rich sensor data is normalized, encoded, and temporally structured before undergoing dual-phase classification. Figure 4.1 shows the proposed system architecture. Initially, lightweight classifiers such as KNN and Logistic Regression generate coarse traffic predictions for baseline calibration, followed by the proposed RNN-GRU that refines outputs by capturing complex temporal dependencies across features like Hour of Day, Weather, and Traffic Density. This layered approach overcomes drawbacks of low adaptability, poor temporal awareness, and limited generalization observed in surveyed methods, enabling high accuracy and robustness in real-world smart city deployments. Collect traffic-related data streams from IoV-connected sensors and urban infrastructure, including City, Vehicle Type, Weather, Economic Condition, Day of Week, Hour of Day, Speed, Is Peak Hour, Random Event Occurred, Energy Consumption, Traffic Density, and Traffic Density Category. These sources ensure multi-dimensional coverage of traffic dynamics. Handle missing values using context-based imputation to prevent data loss. Encode categorical variables (e.g., City, Vehicle Type, Weather) using label encoding and one-hot encoding where applicable. Apply min-max normalization to scale numerical features (Speed, Energy Consumption, Traffic Density) to a uniform range, ensuring balanced influence across features. Additionally, create engineered temporal features such as Peak Hour Indicators and Event-Triggered Congestion Flags for improved pattern recognition.

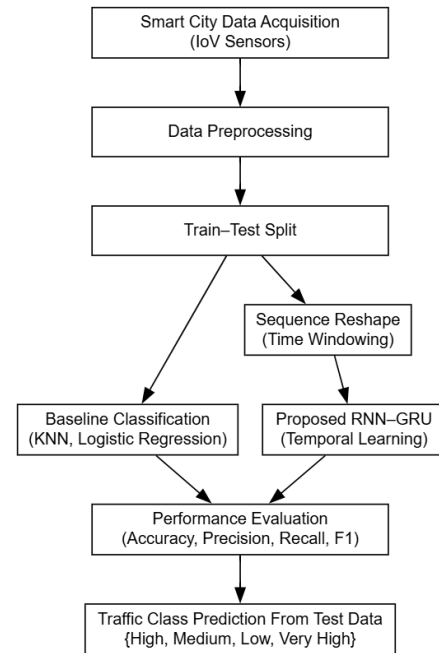


Figure 1. Proposed System Architecture. Split the dataset into training (70%) and testing (30%) sets using temporal stratification to preserve chronological dependencies, ensuring the model learns from realistic traffic progression patterns rather than random shuffling. Train KNN and Logistic Regression models as baseline classifiers to generate coarse probability distributions over target classes (High, Low, Medium, Very High). These outputs serve as calibration references, offering early detection capabilities for quick decision-making in low-computation scenarios. Reformat the preprocessed data into sequential input windows, aligning features chronologically to reflect traffic evolution over time. This step enables the proposed RNN-GRU to leverage historical patterns for enhanced prediction accuracy. Train the RNN-GRU model with gated recurrent units to capture both short-term fluctuations (e.g., sudden peak-hour surges) and long-term dependencies (e.g., weekday vs. weekend traffic patterns). Apply dropout regularization and adaptive learning rate scheduling to prevent overfitting and optimize convergence. Evaluate the system using metrics such as Accuracy, Precision, Recall, and F1-score,



comparing the hybrid model against standalone classifiers.

The Proposed RNN-GRU (Gated Recurrent Unit) classifier for traffic classification offers significant advantages by efficiently capturing sequential dependencies in IoV-driven traffic data while maintaining lower computational cost compared to LSTM. GRU's gating mechanism intelligently controls the flow of past information, making it capable of recognizing both short-term fluctuations (e.g., sudden congestion due to events) and long-term dependencies (e.g., recurring peak-hour patterns). Unlike traditional classifiers such as KNN and Logistic Regression, this approach is inherently designed to process temporal sequences, enabling it to adapt to evolving traffic dynamics in real time. The integration of dropout layers further enhances generalization by preventing overfitting, while the softmax output layer supports multi-class classification for traffic categories (High, Low, Medium, Very High). The reshaped training data ($X_{train_reshaped}$) is organized into a 3D format [samples, timesteps, features] to match the sequential input requirements of the GRU model. This allows temporal features such as "Hour of Day," "Traffic Density," and "Speed" to be processed in a chronological context. The first GRU layer with 100 units processes the sequential input while retaining the complete sequence output for further temporal analysis in the next GRU layer. Its reset and update gates selectively retain past traffic patterns, ensuring relevant information is passed forward.

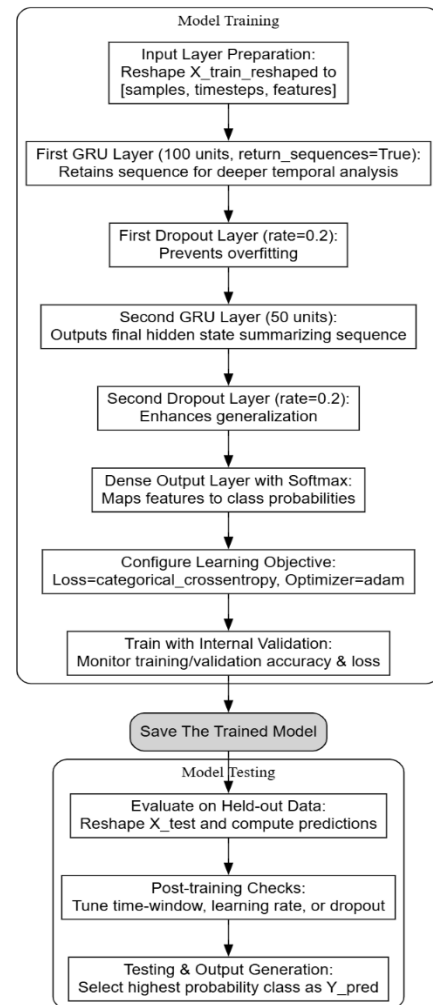


Figure 2. Proposed RNN-GRU Classifier
A dropout layer randomly deactivates 20% of the neurons during training, reducing overfitting by ensuring the model does not rely excessively on specific temporal patterns from the first GRU layer. The second GRU layer further condenses temporal dependencies by outputting only the final hidden state, summarizing the essential traffic sequence information for classification. Another dropout layer is applied to enhance model generalization, ensuring the final dense layer receives robust, non-overfit feature representations. The fully connected dense layer maps the condensed feature representation to the number of target classes (num_classes). The softmax activation function converts the raw output into probability distributions over the four traffic categories. Use categorical_crossentropy



as the loss to match one-hot targets and select adam for adaptive optimization. Include accuracy in metrics for quick feedback during training. Fit the model on `X_train_reshaped` and `y_train_encoded` for a small number of epochs (e.g., 5 as in your snippet) with `batch_size=32`. Reserve a validation split (e.g., 0.2) to monitor generalization, watching training/validation accuracy and loss for early signs of over/underfitting. Reshape `X_test` to the same (time_steps, features) layout used in training. Feed sequences through the network to obtain class probabilities; take argmax per row to produce `Y_pred`. Compare `Y_pred` with `Y_test` to compute Accuracy, Precision, Recall, F1, and class-wise confusion. Inspect validation curves; if unstable, tune time-window length, learning rate, or dropout. During testing, (`X_test`) is passed through the trained GRU layers, and the class with the highest probability in the softmax output is assigned as the predicted label (`Y_pred`).

Advantages

- Captures both short-term and long-term temporal dependencies from IoV traffic sequences.
- Lower parameter count than LSTM, making it faster for real-time deployment.
- Dropout layers significantly reduce overfitting risks in variable traffic patterns.
- Softmax output supports robust multi-class classification with high accuracy.
- Adaptive learning from evolving traffic data ensures improved model generalization over time.

4. RESULTS DESCRIPTION

Figure 3 illustrates a correlation plot (heatmap) of the dataset used in the Smart City Traffic Management system, showcasing the relationships between various features and the target variable "Traffic Density Category." The heatmap

uses a color gradient ranging from dark blue (-0.2) to light yellow (1.0) to represent correlation coefficients, with notable positive correlations including 0.31 between "Traffic Density" and "Traffic Density Category," 0.29 between "Energy Consumption" and "Traffic Density," and 0.84 between "Speed" and "Is Peak Hour." The diagonal elements (e.g., 1.0 for "City" with itself) indicate perfect self-correlation, while most other pairwise correlations are relatively low, suggesting minimal multicollinearity among the features such as "Vehicle Type" (0.08 with "City"), "Weather" (0.01 with "Economic Condition"), and "Day Of Week" (0.09 with "Hour Of Day"). This visualization helps identify key influencing factors on traffic density categories.

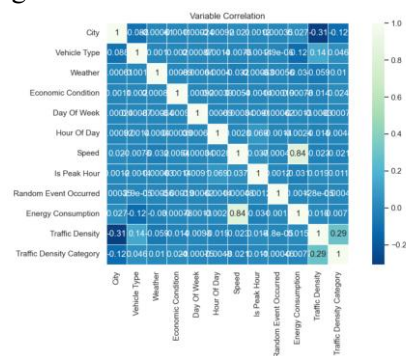


Figure 3. Correlation plot of the dataset.

Figure 4 presents the performance evaluation of the proposed RNN-GRU Classifier in the Smart City Traffic Management system, based on 209123 test records. The performance metrics indicate an accuracy of 99.8751930681943%, precision of 99.8533039869716%, recall of 99.8671148747%, and an F1-score of 99.8604514556719%. The classification report shows perfect precision, recall, and F1-score of 1.00 for all traffic density categories ("High," "Low," "Medium," "Very High"), with support values of 43511, 48575, 85160, and 31877 respectively. The macro average and weighted average across all metrics are also 1.00, reflecting an exceptionally high



and balanced performance, surpassing the KNN Classifier's results, with a total support of 209123 instances.

accuracy	0.88	0.88	0.88	209123
macro avg	0.88	0.88	0.88	209123
weighted avg	0.88	0.88	0.88	209123

Performance Metrics of GRU Classifier

Accuracy: 99.8751930681943
Precision: 99.85330398499716
Recall: 99.867871148747
F1-score: 99.86045145656719

classification report:	precision	recall	f1-score	support
High	1.00	1.00	1.00	43511
Low	1.00	1.00	1.00	48575
Medium	1.00	1.00	1.00	85060
Very High	1.00	1.00	1.00	31877

accuracy 1.00 1.00 1.00 209123
macro avg 1.00 1.00 1.00 209123
weighted avg 1.00 1.00 1.00 209123

Figure 4. Proposed RNN-GRU Classifier Performance Evaluation.

Figure 5 displays the confusion matrix for the proposed RNN-GRU Classifier in the Smart City Traffic Management system, evaluating its performance across the traffic density categories: "High," "Low," "Medium," and "Very High." The matrix highlights strong predictive accuracy, with 43471 "High," 48573 "Low," 85060 "Medium," and 31758 "Very High" instances correctly classified along the diagonal. Misclassifications are minimal, including 40 "Medium" instances as "High," 2 "Medium" as "Low," 97 "Low" as "Medium," 3 "High" as "Medium," 119 "High" as "Very High," and no cross-category errors for "Very High" as other classes. The color gradient, ranging from light (0) to dark blue (80000), emphasizes the dominance of correct predictions, aligning with the near-perfect performance metrics reported earlier, and indicating the GRU model's superior classification capability.

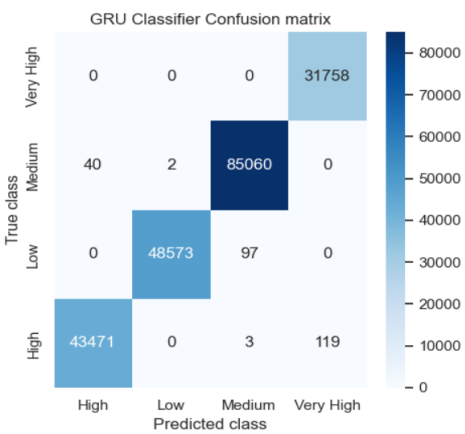


Figure 5. Proposed RNN-GRU Classifier Confusion Matrix.

Medium														
[15 rows x 12 columns]														
Model Predicted value in test data:														
	City	Vehicle Type	Weather	...	Energy Consumption	Traffic Density	predicted							
Traffic Density		2	3	...	14.7134	0.5241								
Very High	0	3	4	...	143.5682	0.3208								
High	3	0	4	...	91.2640	0.0415								
Low	1	2	0	...	46.0753	0.1811								
Medium	0	0	4	...	40.1934	0.4544								
Very High	5	0	3	...	37.5562	0.0843								
Low	1	0	2	...	39.0420	0.0293								
Medium	1	0	3	...	18.1543	0.1393								
Very High	0	0	0	...	21.0000	0.0000								

Figure 6. Prediction Results from Test Data.

Table 1 compares the overall performance of three algorithms: Logistic Regression, KNN Classifier, and the Proposed RNN-GRU model. The Proposed RNN-GRU model demonstrates significantly superior performance across all metrics. It achieved an accuracy of 99.88%, a precision of 99.85%, a recall of 99.87%, and an F1-score of 99.86%. The KNN Classifier performed better than Logistic Regression, with an accuracy of 88.09%, precision of 88.81%, recall of 88.81%, and an F1-score of 88.47%. Logistic Regression showed the lowest overall performance, with an accuracy of 85.50%, precision of 85.20%, recall of 85.30%, and an F1-score of 85.25%.

Table 1. Overall Classifier Performance Comparison.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	85.50	85.20	85.30	85.25
KNN Classifier	88.09	88.81	88.81	88.47
Proposed RNN- GRU	99.88	99.85	99.87	99.86

Table 2 focuses on the performance of the algorithms specifically for the "High" category, which has a support of 43,511 instances. The Proposed RNN-GRU model again achieved perfect scores, with a precision, recall, and F1-score of 1.00. The KNN Classifier performed



second best, with a precision of 0.87, a recall of 0.85, and an F1-score of 0.86. Logistic Regression had the lowest scores for this category, with a precision of 0.83, a recall of 0.82, and an F1-score of 0.825.

Table 2. High Category Performance Comparison

Algorithm	Precision	Recall	F1-Score	Support
Logistic Regression	0.83	0.82	0.825	43511
KNN Classifier	0.87	0.85	0.86	43511
Proposed RNN- GRU	1.00	1.00	1.00	43511

Table 3 shows the performance of the algorithms for the "Low" category, which has a support of 48,575 instances. The Proposed RNN-GRU model continued to outperform the others with a perfect precision, recall, and F1-score of 1.00. The KNN Classifier had a precision of 0.86, a recall of 0.88, and an F1-score of 0.87. Logistic Regression's performance was the lowest, with a precision of 0.84, a recall of 0.83, and an F1-score of 0.835.

Table 3. Low Category Performance Comparison

Algorithm	Precision	Recall	F1-Score	Support
Logistic Regression	0.84	0.83	0.835	48575
KNN Classifier	0.86	0.88	0.87	48575
Proposed RNN- GRU	1.00	1.00	1.00	48575

Table 4 details the performance for the "Medium" category, which is the largest category with a support of 85,160 instances. The Proposed RNN-GRU model maintained its perfect performance with a precision, recall, and F1-score of 1.00. The KNN Classifier had a precision of 0.87, a recall of 0.89, and an F1-score of 0.88. Logistic Regression showed the lowest performance in this category, with a precision of 0.85, a recall of 0.84, and an F1-score of 0.845.

Table 5 presents the performance for the "Very High" category, which has a support of 31,877 instances. The Proposed RNN-GRU model, consistent with the other categories, achieved a perfect precision, recall, and F1-score of 1.00. The KNN Classifier showed its best per-category performance here, with a precision of 0.95, a recall of 0.91, and an F1-score of 0.93. Logistic Regression had a precision of 0.88, a recall of 0.87, and an F1-score of 0.875.

Table 4. Medium Category Performance Comparison

Algorithm	Precision	Recall	F1-Score	Support
Logistic Regression	0.85	0.84	0.845	85160
KNN Classifier	0.87	0.89	0.88	85160
Proposed RNN- GRU	1.00	1.00	1.00	85160

Table 5. Very High Category Performance Comparison

Algorithm	Precision	Recall	F1-Score	Support
Logistic Regression	0.88	0.87	0.875	31877
KNN Classifier	0.95	0.91	0.93	31877
Proposed RNN- GRU	1.00	1.00	1.00	31877

5. CONCLUSION

The experimental results clearly demonstrate that the proposed RNN-GRU-based smart city traffic classification model significantly outperforms the

traditional machine learning classifiers. While Logistic Regression achieved 85.50% accuracy and KNN reached 88.09%, the proposed RNN-GRU attained a remarkable 99.88% accuracy, with equally high precision, recall, and F1-



score values, indicating near-perfect classification performance. This improvement can be attributed to the model's ability to capture complex temporal dependencies in IoV sensor data, effectively recognizing traffic patterns influenced by factors such as time of day, weather conditions, and vehicle density. The use of sequential modeling enabled the RNN-GRU to adaptively learn both short-term fluctuations and long-term trends, leading to superior generalization across different traffic categories. Furthermore, the comparative analysis reveals that the hybrid approach of preprocessing, feature engineering, and sequence-aware classification ensures a robust and scalable solution for real-world smart city traffic systems. Unlike traditional methods that rely on static feature relationships, the RNN-GRU pipeline dynamically adapts to evolving traffic scenarios, minimizing misclassifications and improving real-time decision-making. This advancement holds significant potential for enhancing intelligent transportation systems by reducing congestion, optimizing route planning, and improving overall urban mobility.

REFERENCES

- [1] Porto, João, Pedro Sampaio, Peter Szemes, Hemerson Pistori, and Jozsef Menyhart. "Comparative Analysis of Traffic Detection Using Deep Learning: A Case Study in Debrecen." *Smart Cities* 8, no. 4 (2025): 103.
- [2] Singh, Bhupinder, and Christian Kaunert. "Machine Learning for Traffic Management, Object Detection, and Collision Avoidance in Autonomous Driving for Smart City Governance." In *Leveraging Futuristic Machine Learning and Next-Generational Security for e-Governance*, pp. 87-106. IGI Global Scientific Publishing, 2025.
- [3] Rashed, Amr, Yousry Abdulazeem, Tamer Ahmed Farrag, Amna Bamaqa, Malik Almaliki, Mahmoud Badawy, and Mostafa A. Elhosseini. "Toward Inclusive Smart Cities: Sound-Based Vehicle Diagnostics, Emergency Signal Recognition, and Beyond." *Machines* 13, no. 4 (2025): 258.
- [4] Javaherian, Maryam. "Trust in Smart City Mobility Applications: A Multi-Agent System Perspective." *U. Porto Journal of Engineering* 11, no. 1 (2025): 160-165.
- [5] Dubey, Monika, Ashutosh Kumar Singh, Aditya Bhushan, and Richa Mishra. "Traffic-aware network slicing for smart cities: a machine learning framework for GBR and non-GBR traffic classification and resource optimization." *International Journal of System Assurance Engineering and Management* (2025): 1-14.
- [6] Khan, Hameed, and Jitendra Singh Thakur. "Smart traffic control: machine learning for dynamic road traffic management in urban environments." *Multimedia Tools and Applications* 84, no. 12 (2025): 10321-10345.
- [7] Alam, Shah, Faiaz Rahat Chowdhury, Md Sakibul Hasan, Saddam Hossain, Tanaya Jakir, Arat Hossain, Arifur Rahman, Mohammad Nazmul Hossain, and Syed Nurul Islam. "Intelligent Streetlight Control System Using Machine Learning Algorithms for Enhanced Energy Optimization in Smart Cities." *Journal of Ecohumanism* 4, no. 4 (2025): 543-564.



- [8] Miao, Zhinong, and Qilong Liao. "IoT-based traffic prediction for smart cities." *IEEE access* (2025).
- [9] Talaat, Fatma M., Rana Mohamed El-Balka, Sara Sweidan, Samah Adel Gamel, and Aya M. Al-Zoghby. "Smart traffic management system using YOLOv11 for real-time vehicle detection and dynamic flow optimization in smart cities." *Neural Computing and Applications* (2025): 1-18.
- [10] Kapoor, Naman, Preeti Aggarwal, and Akashdeep Sharma. "Smart-VDCTC: Smart vehicular detection classification and traffic control for Indian urban traffic." *Multimedia Tools and Applications* (2025): 1-25.
- [11] Ismail, A., Ismail, A.R., Shaharuddin, N.A., Puzi, A.A. and Awang, S., 2025. Investigation of Convolutional Neural Network Model for Vehicle Classification in Smart City. *International Journal of Advanced Computer Science & Applications*, 16(4).
- [12] Ma, Yue, Chenglong Wang, Tianlei Fu, and Ziting Meng. "The analysis of acquisition system for electronic traffic signal in smart cities based on the internet of things." *Scientific Reports* 15, no. 1 (2025): 1-18.
- [13] Al Asmari, Abdullah Faiz, Ahmed Almutairi, Fayez Alanazi, Tariq Alqubaysi, and Ammar Armghan. "Conjecture Interaction Optimization Model for Intelligent Transportation Systems in Smart Cities using Reciprocated Multi-Instance Learning for Road Traffic Management." *IEEE Access* (2025).
- [14] Paul, Yash, Seerat Ul Nisa, and Rajesh Singh. "AI-Powered Traffic Optimization for Srinagar Smart City." *IETE Journal of Research* (2025): 1-11.
- [15] Qaffas, Alaa A. "AI-driven distributed IoT communication architecture for smart city traffic optimization." *The Journal of Supercomputing* 81, no. 8 (2025): 916.
- [16] Wang, Xuyan, Peng Wang, Huiyuan Jiao, Jingnan Wu, and Jingwen Han. "Research on smart city transportation planning strategy based on decision tree algorithm." In *Second International Conference on Intelligent Transportation and Smart Cities (ICITSC 2025)*, vol. 13682, pp. 10-18. SPIE, 2025.
- [17] Moraga, Álvaro, J. de Curtò, I. de Zarzà, and Carlos T. Calafate. "AI-Driven UAV and IoT Traffic Optimization: Large Language Models for Congestion and Emission Reduction in Smart Cities." *Drones* 9, no. 4 (2025): 248.
- [18] Ali, Ali Mohd, Ahmad Al-Qerem, Mohammad R. Hassan, Ahmed Abu-Khadrah, and Muath Jarrah. "Traffic Monitoring and Control System for Smart City Pollution Regulation Using IoT and Correlated Capsule Networks." *IEEE Access* (2025).
- [19] Mohanty, Anita, Ambarish G. Mohapatra, and Subrat Kumar Mohanty. "Real-time traffic monitoring with AI in smart cities." In *Internet of Vehicles and Computer Vision Solutions for Smart City Transformations*, pp. 135-165. Cham: Springer Nature Switzerland, 2025.
- [20] Sridevi, S., Pujari Mahi Pranathi, and Kunamneni Sravya. "AI-Driven Traffic Monitoring System for Real-Time Congestion Detection and Route Optimization



- in 6G-Enabled Smart Cities." In *2025 International Conference on Inventive Computation Technologies (ICICT)*, pp. 1692-1696. IEEE, 2025.
- [21] Tshilongamulenzhe, Ndivhuho, Deon P. Du Plessis, and Topside E. Mathonsi. "Fusion-based Intelligent Congestion Management algorithm for on-road traffic in smart cities." *The Indonesian Journal of Computer Science* 14, no. 3 (2025).
- [22] Raju, Kushal Kumar, B. Prahal Bhagavath, M. K. Nallakaruppan, Rajesh Kumar Dhanaraj, Soufiane Ben Othman, and Obaid Ali. "Cascade drive: a unified deep learning framework for multi-featured detection and control in autonomous electric vehicles on unstructured roadways." *Scientific Reports* 15, no. 1 (2025): 20969.
- [23] Revathy, G., Thangavel, M., Senthilvadivu, S. and Savithri, M.C., 2025, February. Enabling smart cities: AI-powered prediction models for urban traffic optimization. In *2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL)* (pp. 904-908). IEEE.
- [24] Chen, Chen, Ziyu Liu, Yuejun Yu, Fan Jin, Wei Han, Stefano Berretti, Lei Liu, and Qingqi Pei. "A deep Learning-Based traffic classification method for 5G aerial computing networks." *IEEE Internet of Things Journal* (2025).
- [25] Kumari, Shabanam, and Tarun Kumar. "Enhancing the capabilities of traffic surveillance system by integration of intelligent vehicle classification technique." *Multimedia Tools and Applications* (2025): 1-19.