FSTS Fog-Based Smart Transportation System a Novel Approach: Navigating towards Future of Urban Mobility

Deepa Arora and Oshin Sharma*

Department of Computer Science and Engineering, SRM Institute of Science and Technology,
Delhi-NCR Campus, Ghaziabad, India
Email: csedeepaarora@gmail.com (D.A.); oshins1@srmist.edu.in (O.S.)
*Corresponding author

***Abstract*—The number of lives and possessions lost because of ineffective and dangerous traffic management has risen significantly in recent years. Innovative solutions are required for effective transportation management due to the accelerated urbanisation and rising demands on the transportation infrastructure. We propose a Fog-Based Smart Transportation System (FSTS) that utilizes fog computing to enhance traffic management, environmental monitoring, and road safety. In contrast to traditional centralized cloud solutions, the FSTS combines fog nodes, proxy servers, and cloud servers to facilitate low-latency, energy-efficient, and dynamic execution of data. The proposed framework was validated by iFogSim2, simulating up to 60 fog nodes together with various smart cameras and sensors. A comparison is conducted by utilizing two parameters, namely latency and energy consumption. The results indicated that the FSTS decreases latency by as much as 82% and energy consumption by approximately 75% in comparison to leading methodologies like Scalable Microservices Placement (SMP) and Reliable Resource Allocation and Management (R2AM). These enhancements underscore the system's capacity to reduce congestion, facilitate real-time accident identification, and optimize energy use. The suggested architecture prioritizes data security and privacy through the integration of encryption and safe communication between fog nodes and the cloud. Therefore, experimental results validate that the proposed FSTS approach outperforms alternatives by reducing latency, energy consumption, and optimized traffic efficiency.**

***Keywords*—fog commuting, smart transportation system, iFogSim, smart city, cloud computing, Internet of Things (IoT)**

I. Introduction

A major problem for the world's constantly expanding metropolitan centres is the effective management of transport networks. Today's urban transportation systems encounter previously unreachable challenges because of declining infrastructure, increasing vehicle density, and fast population increase [1]. Chronic traffic congestion, frequent accidents, ineffective traffic flow, and declining environmental quality are all clear signs of the effects. Significant time and fuel losses are caused by congestion alone, and greenhouse gas emissions are increased by extended waiting at junctions. At the same time, lives are at risk due to the incapacity to promptly identify accidents or road hazards [2]. Conventional cloud-based Transportation Systems have facilitated centralized data processing for traffic management. Yet cloud-exclusive systems encounter numerous constraints: enhanced latency in decision-making, bandwidth congestion due to extensive sensor data transmission, exposure to singular points of failure, and challenges in scaling for large-scale implementations. The limitations make centralized techniques inadequate for real-time, safety-critical, and environmentally sustainable transportation management. Fog computing is now seen as a potential paradigm that reduces latency and enhances system responsiveness by moving processing and data storage near to the data source [3]. Fog-based solutions have been found to enhance response time and throughput in isolated situations. Several studies prioritize latency reduction or authentication over energy efficiency, scalability, and robustness under dynamic mobility patterns.

To bridge these gaps, this paper proposes the Fog-Based Smart Transportation System (FSTS), a three-tier architecture that integrates Internet of Things (IoT) end devices, a distributed fog layer, a proxy layer for intermediate aggregation and security, and a cloud layer for global analytics. The proxy layer represents a strategic design decision. Proxies execute compression, aggregation, redundancy verification, and basic encryption prior to transmitting processed summaries to the cloud. This eliminates redundant transmissions, decreases bandwidth requirements, and establishes an additional level of privacy without overloading resource-constrained fog nodes with all security functions. FSTS follows a multi-tier fog-proxy-cloud architecture that processed data near the source to minimized latency and support real-time decisions.

Manuscript received July 28, 2025; revised August 18, 2025; accepted November 12, 2025; published March 10, 2026.

doi: 10.12720/jait.17.3.488-499

The framework allocates computing across IoT, fog, proxy, and cloud layers to guarantee the execution of time-sensitive operations in proximity to the data source. Secondly, proxy servers are utilized to execute aggregation, encryption, and compression, hence minimizing redundant data transmission and protecting privacy without compromising response time. Third, an adaptive fault-handling technique ensures task continuance despite node failures. The design options were assessed by iFogSim2 [4] simulations under actual traffic conditions, revealing persistent decreases in latency and energy consumption, as well as enhanced reliability.

The key contribution of FSTS provides following contributions:

- We proposed three tier framework that designed to support real time traffic processing and reduce latency.
- By detecting abnormal traffic patterns, such as unexpected delays, the system can determine potential accidents or road closures and update travel-time predictions or recommend alternative routes.
- Proxy server performs aggregation and security functions to reduce redundant data transmission and improve bandwidth efficiency.
- Our fog-enabled task offloading decreased the average energy usage of IoT devices and gateways by 22% relative to a cloud-only system.

The rest of the paper is organized in the following manner: Section II provides an overview of the traditional approaches and their limitations to identify the gap between the present system and the suggested methods. Section III provides an explanation of the proposed approaches, operational model of real time smart transportation system, sequence diagram and detailed information on the algorithms used. Section IV provides an explanation of the system setup, evaluation measures, and comparison findings. Section V provides a concise summary of the whole study and outlines potential future research.

II. Literature Review

A Recent advancement in fog and edge computing have drastically altered the architecture of Intelligent Transportation Systems (ITS) by tackling issues related to latency, scalability, and energy efficiency. This section reviews recent studies by categorizing them systematically according to their primary research focus, thereby clarifying existing contributions and their limitations. This categorization facilitates a systematic and analytical examination of the application of fog computing in transportation systems, focusing on latency optimization, safety and fault tolerance, and overall system resilience. This study identifies existing gaps through the examination of these categories, which motivate the design and implementation of the proposed FSTS.

A. Fog Computing for Latency Reduction and Real Time Traffic Optimization

Researchers have increasingly investigated distributed architectures which bring computers nearer to data sources to facilitate real-time decision-making in urban transportation. For instance, Deshpande and Hsieh *et al.* [5] introduced a fog computing framework for intelligent urban mobility, leveraging fog computing's low latency and distributed processing features to develop an advanced traffic light control system. Their method exhibited enhanced responsiveness in traffic signal modifications but it was confined to single-intersection optimization, missing scalability, and energy analysis across wide networks. Building upon similar motivations, Wu *et al.* [6] suggested a fog-based traffic management process that incorporates reinforcement learning to enhance vehicle flow and eliminate communication bottlenecks. Although successful in improving coordination, their approach was essentially algorithmic and unable to assess energy trade-offs or fault tolerance in real-world vehicle scenarios. Parallel efforts have sought to enhance traffic data handling through cloud–fog integration. A shift toward mobile sensing was seen in the work of Jan *et al.* [7] utilized the Hadoop and Spark frameworks to handle real-time transportation data and distribute findings to intelligent transportation systems. While their big-data-centric methodology facilitated fast data distribution, its reliance on centralized cloud clusters resulted in latency overhead and scalability challenges, which can be alleviated through fog-layer computation as implemented in the proposed FSTS.

The rising demand for advanced transportation infrastructure has encouraged research toward computing paradigms capable of managing extensive, dynamic, and time-critical vehicle data. Conventional cloud-based ITS rely on remote servers for centralized processing and comprehensive traffic planning. Although these designs facilitate large-scale data analytics, they encounter significant limits in real-time urban mobility, such as elevated latency, network congestion, energy inefficiency, and restricted fault resilience. To address these challenges, academics have redirected focus to fog computing, a distributed architecture that brings processing and storage resources nearer to data sources like vehicles, roadside sensors, and gateways. Fog computing facilitates local processing, hence reducing communication latency and eliminating computational burden from the cloud. For this approach Tu *et al.* [8] designed a smartphone-sensor-based model for real-time vehicle movement detection, demonstrating accuracy in identifying various driving behaviors. The system depended on end-device processing and lacked interaction with fog-level coordination, restricting its capacity to manage high-volume traffic situations. Similarly, Tyagi and Aswathy *et al.* [9] examined frameworks for autonomous intelligent vehicles, highlighting the cost and time efficiencies achieved through automation. Their study emphasized the advantages of autonomous systems but remained theoretical, failing to propose a tangible fog-assisted communication framework for real-time vehicular decision-making. Another author Qin and Zhang *et al.* [10] proposed a fog-enabled intelligent traffic light control system utilizing Q-learning for adaptive flow management. Their findings indicated a decrease in latency for data control; however, the model was deficient in multi-layer fault recovery and energy-aware scheduling, both of which are critical for densely populated urban scenarios. Intelligent decision support and predictive modeling have also been explored through data-driven and multi-agent methodologies. Muntean *et al.* [11] implemented a Multi-Agent System (MAS) for intelligent urban traffic management utilizing categorization and forecasting techniques, including K-nearest neighbor and random tree. Despite enhancements in forecast accuracy, their centralized processing architecture constrained real-time adaptation. Similarly, the iFogSim simulator [12] and its subsequent applications [13] have been extensively utilized to model fog-based scenarios; however, these investigations prioritized tool validation above highlighting integrated security, scalability, or resilience in transportation contexts. Complementary to this, Golhar and Kshirsagar [14] proposed a traffic detection framework that integrates camera surveillance and big-data analytics for the evaluation of congestion. However, providing extensive monitoring, their dependence on cloud computing resulted in increased transmission delay and energy usage.

B. *Security, Privacy, and Fault Tolerance in Fog Based Vehicular Networks*

A growing body of research has also focused on securing vehicular networks within fog computing environments. According to the study, Gupta *et al.* [15] proposed an IoT-based system that connects vehicles with traffic signals to facilitate low-latency communication; but the model was deficient in privacy safeguards and fault-tolerance features. Extending this domain, Rathore *et al.* [16] developed a fog-assisted method utilizing vehicular cameras to detect driving infractions in real time. Although efficient for monitoring, it lacked energy optimization and data aggregation methods essential for large-scale operations. In contrast, Al-Mekhlafi *et al.* [17] introduced a security framework utilizing Chebyshev polynomials for 5G-enabled vehicle fog networks, emphasizing pseudonym revocation and authentication. While the approach improved security, it overlooked latency and energy factors. Similarly, Butt *et al.* [18] proposed a proactive, data-driven fog computing system to enhance police in smart cities by facilitating real-time incident analysis via distributed IoT-fog coordination. The model effectively minimized network latency and enhanced data processing efficiency; nevertheless, it was predominantly designed for static surveillance contexts and did not evaluate scalability, incorporate fault-tolerance mechanisms, or implement energy optimization measures vital for dynamic vehicle networks. The identified limitations underscore the research gap that the proposed FSTS aims to fill by incorporating proxy-based aggregation, adaptive scheduling, and decentralized fault recovery to attain low-latency, energy-efficient, and resilient performance in massive transportation systems. The work by Li *et al.* [19] introduced a lightweight authentication and key agreement technique based on Secure Socket Layer (SSL) for Vehicular Ad-hoc Networks (VANETs), enhancing communication security and minimizing handshake delay. Nonetheless, it failed to account for fog-level scalability, energy efficiency, and fault tolerance, which the proposed FSTS resolves via integrated proxy-layer encryption and decentralized processing for real-time transport systems. Comparable limitations were noted in Jiang *et al.* [20] introduced a vehicle-assisted service caching technique to enhance task offloading in vehicular edge computing, thereby enhancing task response time and network efficiency. However, their methodology was deficient in energy optimization, data security, and multi-layer fault recovery methods, which the proposed FSTS addresses via proxy-based aggregation, secure data management, and robust fog-layer scheduling.

Beyond security, several works have investigated authentication and intrusion detection. Aazam and Huh [21] introduced Anonymous Authentication Architecture (ANAA)-Fog, an anonymous authentication framework for 5G-enabled vehicle fog computing that enhanced user privacy and diminished authentication latency. yet the study mainly concentrated on security and anonymity, neglecting the assessment of energy consumption, fault tolerance, and task scheduling efficiency—domains that the proposed FSTS improves via energy-aware scheduling, proxy-layer encryption, and robust fog coordination.

Some studies have focused on domain-specific applications of fog computing to urban infrastructure and energy management. Kumara *et al.* [22] created a fog-based smart parking system that maximized resource allocation and reduced urban waiting time. They concentrated on static parking management and lacked large-scale delay and energy analysis, which FSTS applies to dynamic vehicular networks. For authentication Al-Shareeda and Manickam [23] proposed a mutual model for vehicle-based COVID-19 detection systems utilizing 5G technology. Their model improved security but ignored computation offloading and latency efficiency, which are addressed in FSTS via proxy-layer security and adaptive scheduling.

C. *Energy Efficiency and Resource Optimization in Fog Based Intelligent Transport*

Several studies have also explored energy optimization and resource scheduling within fog environments such as Mardiana *et al.* [24] evaluated fog computing in the context of renewable energy management to enhance computational efficiency. Still, the study was limited to a specific energy domain and did not address vehicular data flows. FSTS extends this fog efficiency to smart transport systems. Ullah *et al.* [25] developed Hybrid Deep Learning-Intrusion Detection System (HDL-IDS), a hybrid deep learning model for intrusion detection in the Internet of vehicles, which attained an accuracy of 99% in anomaly detection. However, it failed to incorporate latency-sensitive analytics or multi-layer data flow, which FSTS achieves via real-time fog-based security.

D. *Research Gaps*

The literature review indicates that most studies on fog-based intelligent transportation focus on specific challenges, including latency, security, and task scheduling, yet rarely propose a comprehensive framework that integrates these elements under real-time, scalable conditions. Current models frequently depend on cloud-centric architectures or single-layer fog deployments, which face challenges related to energy inefficiency, network congestion, and restricted fault tolerance in dynamic vehicular contexts. Moreover, while research conducted by Butt *et al.* [18] and Li *et al.* [19] improved response times and security, they failed to address essential elements of multi-layer coordination and real-time fault recovery required for extensive vehicular networks. Research focused on simulation, employing tools like iFogSim, has theoretically substantiated fog architectures; however, it lacks empirical evaluation in realistic traffic and mobility scenarios. In addition, a few studies incorporate data privacy and redundancy mechanisms directly within the computational workflow. The proposed FSTS addresses existing gaps by implementing a multi-tier architecture that incorporates proxy-assisted aggregation, adaptive scheduling, and decentralized fault management. This framework is designed to be unified, energy-efficient, and resilient, and has been validated through simulations conducted in iFogSim2.

III. Proposed Methodology

A. *Smart Transportation System*

Data capturing: Fog nodes in a smart transport system can collect data from cameras and sensors. Fog nodes are Intermediate computational units situated at Roadside Units (RSUs), gateways, or traffic signals. They execute local processing, storage, and decision-making in proximity to the data source. In FSTS, fog nodes manage real-time functions including congestion detection and accident notifications. The information gathered pertains to the management of transportation, including Monitoring the movement of traffic on roads and highways, counting the number of cars that pass via a particular location, evaluating traffic congestion and accident rates, counting environmental elements like noise levels and air quality, gathering information on the arrivals and departures of buses and trains. Data processing: The fog node processes the collected data on transportation in real-time. This processing may entail the following steps: (1) Traffic Congestion Detection: Analyzing traffic density and vehicle speeds to spot congested locations. Accident detection involves spotting crashes or other strange activities on the road. (2) Route optimization: Figuring out the best routes for cars to minimize travel time. (3) Public transportation management: Providing real-time information on public transportation timetables and delays. (4) Weather data analysis: Analyzing weather data to provide road condition warnings. (5) Smart LED Displays: The fog node can update smart LED displays positioned in key areas to give drivers and pedestrians real-time information. Traffic warnings, road closures, detours, and updates on public transportation are some examples of information. (6) Data storage: To process and make decisions instantly, the fog node may temporarily store data. To exchange information and plan activities connected to FSTS, the fog node communicates with the cloud server and maybe with other fog nodes. Fig. 1. illustrates the smart transportation system.

In Fig. 1, the smart transportation system uses real-time data from cameras and sensors to power smart LED displays that are carefully placed at crucial intersections across the transit network. At this point they will take vehicle wise decisions. These screens act as informational beacons, giving drivers the most recent details on traffic, road closures, and detours. By enabling them to make educated judgements, this enables users to significantly improve traffic flow. The system’s scope includes environmental monitoring as well, which includes the gathering of information on temperature, humidity, and air quality. This wealth of environmental information helps guide important urban planning decisions, from the strategic placement of green areas to the creation of effective pollution control strategies. They provide road safety, traffic prediction, real time information along with reduced time and transportation cost. In cloud data exploitation and analysis has been done. Cloud also stored the periodic data for future research.

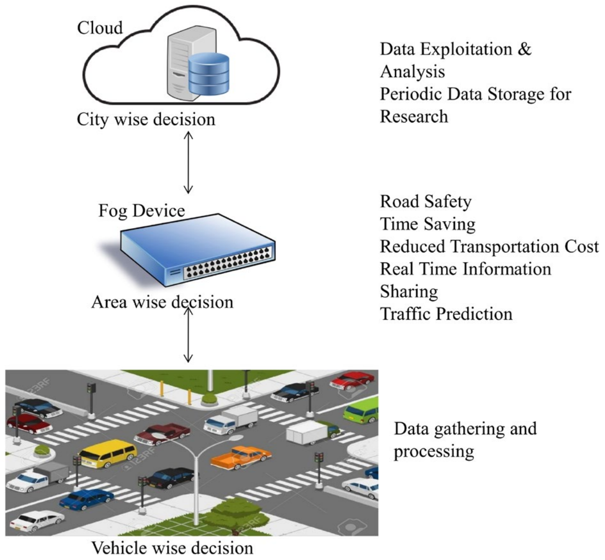
Cloud
City wise decision
Data Exploitation & Analysis
Periodic Data Storage for Research
Fog Device
Area wise decision
Road Safety
Time Saving
Reduced Transportation Cost
Real Time Information Sharing
Traffic Prediction
Data gathering and processing
Vehicle wise decision

Fig. 1. Smart transportation system.

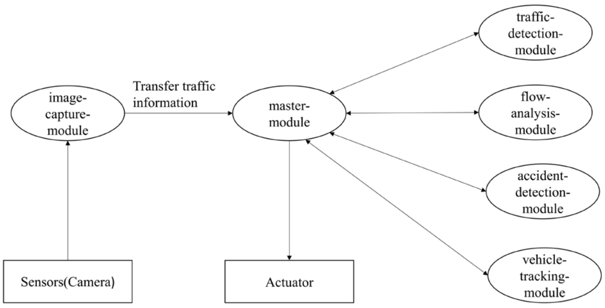

Fig. 2. Application model of smart transportation system.

Fig. 2 illustrates the application model [8]. Sensor in cameras sensed the data and send it to image-capture-module. That module transferred the data to master-module where all the processing has been done and it is connected to traffic-detection-module, flow-analysis-module, accident-detection-module, and vehicle-tracking-module. At the end master-module send information to actuator as output.

B. Proposed System Architecture

The architecture of the proposed FSTS is structured to provide low latency, scalability, and secure data management by distinctly separating duties across three tiers: sensing, fog, and cloud. The proposed FSTS architecture has three tiers, shown in Fig. 3.

In tier 1, multiple devices including cameras, intelligent sensors, and GPS-equipped units integrated into motor vehicles continuously collect data on vehicular density, traffic occurrences, and surroundings. This unstructured and voluminous raw data renders quick centralized transmission inefficient. Cameras are strategically positioned to record pertinent data for transportation management, including traffic flow, vehicle counts, road conditions, and any other information needed.

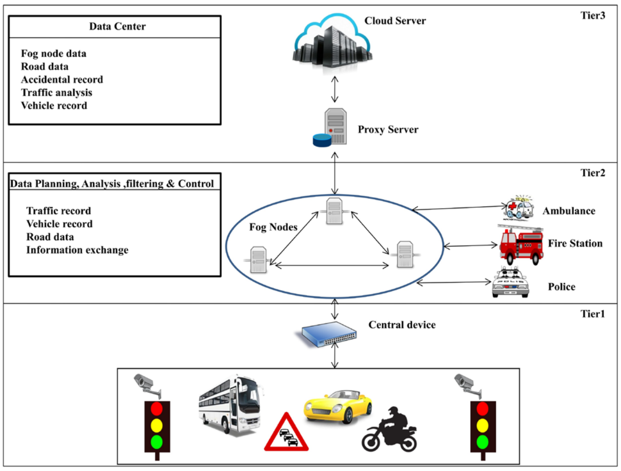

Fig. 3. Three tier architecture for smart transportation system.

Throughout the urban region, several cameras with traffic sensors are deployed in strategic locations. Vehicle movement, speed, traffic congestion, and other important traffic data are all recorded by sensors. Data is locally pre-processed and compiled by sensors before being sent to a central microcontroller. At important crossroads, motorways, and major roads with large traffic volumes, sensors are placed across the urban area. The number of sensors might range from dozens to hundreds or more, depending on the magnitude and complexity of the surrounding area's traffic. The central microcontroller, also known as a data aggregator, serves as an interface between the fog nodes and the cameras. It gathers information from the cameras and gets it ready for processing. In tier 2, fog layer acts as the core of the architecture by performing localized processing close to the data source. Fog nodes filter and aggregate sensor streams, execute congestion detection and accident recognition algorithms, and provide real-time outputs to smart LED displays and navigation applications. By confining time-critical tasks to the fog nodes, the system significantly reduces dependency on the cloud and minimizes latency. Importantly, fog nodes also ensure secure communication by applying encryption and authentication mechanisms before forwarding summarized data. Central microcontroller is situated at the urban area's geographic center, frequently in a traffic management facility. All sensors in its coverage area gathered and processed data. Before sending the captured data to the fog nodes, the microcontroller performs basic data filtering and aggregation. For each city or metropolitan region, a single central microcontroller that collects data from all the sensors in its coverage area. Between the edge devices (cameras) and the cloud, the fog node acts as a transitional layer. It oversees gathering, analyzing, and transferring data so that choices can be made instantly.

A comprehensive category comprising end-user devices such as automobiles, smartphones, or roadside controls that can do lightweight computations. Depending on the size of the system, fog nodes may be dispersed over various transportation zones. To exchange information and coordinate activities regarding transportation, fog nodes communicate with the cloud server and other fog nodes. The urban region has numerous fog nodes, many of which are situated near traffic signal controllers. Data is sent to fog nodes from the main microcontroller. Several fog nodes are dispersed across the city, usually at traffic signal control sites. The quantity of traffic intersections and the intricacy of the road network influence the number of fog nodes. Real-time traffic analysis is done by them to spot regions with a lot of traffic, accidents, or anomalies. Local traffic signal optimization via fog nodes can reduce congestion or raise alarms. Between fog nodes and the cloud, a proxy server (also known as a data forwarder) serves as an intermediary layer. Data compression, authentication management, and fault-tolerant, effective transmission are their responsibilities. This supplementary layer mitigates bottlenecks and adjusts workloads as the system expands across massive networks. It gathers information from fog nodes and gets it ready to send to the cloud. Depending on the system's size and the amount of data being sent to the cloud server, one or more proxy servers may be required. For secure transmission, proxy servers can also perform data compression and encryption. The function of the cloud server in tier 3 is to store data for longer periods of time and to carry out more sophisticated data analytics. To transmit information and support long-term data storage and retrieval, it talks with the fog nodes. Cloud storage receives data from proxy servers for long- term archival and additional research. a cloud infrastructure that can handle analytics and data storage. The quantity of cloud data centres is determined by the need for scalability and redundancy. The cloud provides persisting services such data storage, predictive analytics, and strategic traffic management. Cloud-based historical records are utilized to produce traffic forecasts and environmental impact models, which are regularly broadcast to the fog layer to improve local decision-making.

The proposed FSTS introduces a proxy-assisted architecture in which proxy servers operate as active data aggregators that compress, encrypt, and filter fog-level outputs before transmission. This system incorporates a master-module pipeline that concurrently performs image capture, traffic detection, flow analysis, accident identification, and vehicle tracking, facilitated by per-area microcontrollers that perform initial filtering to reduce redundant upstream traffic. Furthermore, FSTS implements a closed-loop actuator feedback system, wherein smart LEDs at crossings are updated in an event-driven manner exclusively during important changes, thus minimizing redundant transmissions and energy consumption. In addition, the system integrates environmental sensing parameters which includes pollution and air quality monitoring in fog-level decision-making rather than previous research that focused on traffic. These architectural and algorithmic decisions jointly guarantee that time-sensitive processing is confined to the fog layer, while the cloud obtains only compressed and policy-relevant summaries. Validated via iFogSim2 simulations under identical conditions as the SMP and R2AM baselines, FSTS attained a latency of 48.7 ms and an energy consumption of 172.7 J, illustrating substantial efficiency improvements and affirming the efficacy of its proxy-plus-master-module configuration in urban-scale contexts.

C. Security and Privacy in FSTS

Vehicle identification, accident reports, and real-time position data are among the extremely sensitive data handled by the proposed FSTS. Securing this information is essential for user confidence and compliance to regulations. Data in vehicular fog systems is susceptible to risks including eavesdropping, spoofing, fake data injection, denial-of-service assaults, and illegal tracking. To reduce such hazards, FSTS incorporates security and privacy protections directly into its architecture at several levels. The proposed FSTS processes sensitive vehicular data, including identities, accident reports, and real-time positions, necessitating robust security measures. FSTS incorporates multi-layered security into its architecture to safeguard against eavesdropping, spoofing, false data injection, denial-of-service attacks, and tracking. Communication among sensors, fog nodes, proxies, and the cloud is protected by lightweight AES-based encryption, which maintains confidentiality while minimizing latency. Mutual authentication among fog, proxies, and vehicles, facilitated by PUF-based schemes, effectively mitigates impersonation and replay attacks while maintaining minimal overhead. Fog nodes aggregate and anonymize raw data to preserve privacy, reporting only density or speed instead of individual Identifications (IDs), in alignment with recent ITS privacy frameworks. Intrusion detection at the fog layer, drawing on ensemble Intrusion Detection System (IDS) models that achieve over 99% detection accuracy in the internet of vehicles, effectively blocks malicious traffic locally prior to its arrival at the cloud. Redundant proxies function as security gateways, ensuring data integrity and maintaining service continuity during failures or attacks. This layered design provides real-time protection at the edge, contrasting with cloud-only ITS security, and affirms FSTS as a viable solution for safety-critical urban mobility.

D. Operations of a Fog Computing-Based Smart Transportation System

Fig. 4 shows a high-level description of the operations of a fog computing-based smart transportation system. To increase the effectiveness and safety of transportation networks, FSTS continuously gathers, analyses, and transmits real-time transportation information. User interfaces, data formats, and implementation details can all be customized to the system’s and its users’ needs. Input: Data captured by sensors and camera such as vehicle count, weather data, traffic detection, accident tracking, public transport tracking.

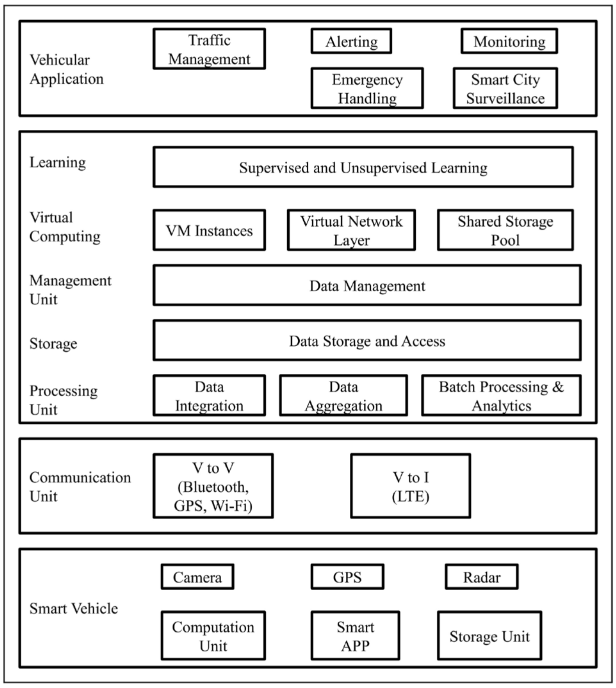

Fig. 4. Operational model of real time smart transportation system.

Output: Real time transportation information such as road condition alert, accident detection. Data collected from cameras and sensors (e.g., traffic flow, vehicle counts, road conditions, weather data).

(1) Set up the System: Set up the system’s LED displays, cameras, sensors, and fog nodes. Set up communication protocols and security options.
(2) Data Capture (Cameras and Sensors): Constantly gather information about transportation using cameras and sensors. Traffic flow, vehicle numbers, road conditions are all included in the data.
(3) Interaction with users: Interact with users by responding to their inquiries for information on transportation. Users may ask for information about traffic patterns, schedules for public transportation, and suggested routes.
(4) Process Data: Fog nodes process data by receiving it from cameras and sensors: Real-time processing of the gathered data to produce insightful results. Data processing involves the following: Traffic jam detection, finding accidents, Analyses of weather data, Route improvement, Controlling public transport.

(5) Update smart LED displays (Fog Node): If necessary, provide real-time information updates to smart LED displays. Traffic warnings, road closures, detours, and updates on public transportation are some examples of information.

(6) Data storage (Fog Node and Cloud Layer): Temporarily store gathered data for processing and decision-making in real time. Periodically upload data to the cloud for archival purposes and historical research.

(7) Data Communication: Communicate with the cloud server to exchange data and analysis results (Fog Node and Cloud Layer). Information about transportation should be shared with other fog nodes for coordination.

User Information Provided: Provide real-time transport information in response to user requests. Information delivery via user interfaces or other means.

Fig. 5 illustrates the sequence diagram of smart transportation system. The sequence diagram demonstrates the interaction flow among IoT devices, fog nodes, and the cloud throughout a standard data processing cycle.

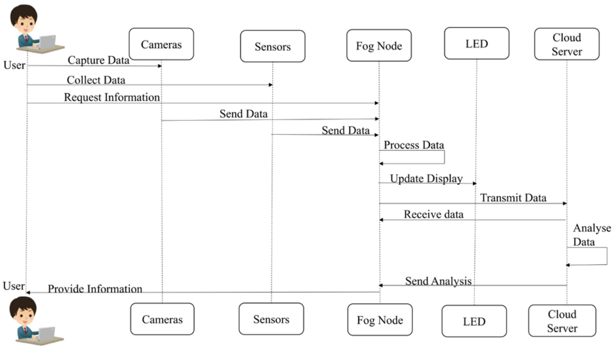

Fig. 5. Sequence diagram of generation and execution for smart transportation system.

Initially, sensors or IoT end devices produce raw data streams, including patient vital signs, traffic conditions, and agricultural field readings. The data is sent to a proximate fog gateway or node, which promptly conducts filtering, preprocessing, and prioritization. Non-critical data may be buffered locally, whereas urgent alerts are prioritized for immediate action. The fog node subsequently either executes a local decision (e.g., activating an actuator, rerouting vehicles) or transmits aggregated information to the cloud layer for advanced analytics and storage. Cloud responses, including predictive models and updated control policies, are transmitted back to the fog and then to the end devices. The sequence diagram highlights real-time decision loops at the fog level and long-term analytics at the cloud level, confirming fog's function in minimizing latency and network overhead while ensuring system scalability. For example, a user initiating a request for transportation services via a mobile application or user interface, the Smart Transportation System quickly receives and efficiently processes the request. Following that, the system does route planning by considering real-time traffic data, road conditions, and user preferences. Connected cars are communicated with to offer route information and receive real-time updates. Subsequently, the system transmits the suggested route to the user, who later verifies and approves the chosen option. Simultaneously, the Traffic Management System modifies traffic signals and regulates traffic flow in accordance with the predetermined routes of interconnected cars. As cars follow the suggested routes, they constantly provide updates to the system. The system engages in real-time monitoring, enabling it to adjust to dynamic traffic situations. If there is an emergency or an unexpected incident, the system has the capability to redirect cars and notify users accordingly. During the user's arrival at the destination, the system acknowledges the confirmation and allows users to offer comments on the transportation service, therefore contributing to the continuous improvement of the system.

Fig. 5 demonstrates the whole behavioral workflow from the moment vehicle and roadside sensors acquire mobility data to the execution of decisions. The flow indicates that time-sensitive events, such as congestion alerts and accident notifications, are processed directly at the fog layer rather than awaiting cloud processing. This design decision eliminates lengthy communication pathways and markedly decreases reaction latency, consistent with the 82% latency reduction recorded in the simulation. Furthermore, unprocessed sensor data is subjected to local filtering prior to transmission, which reduces packet size and minimizes communication energy expenditure—facilitating a 75% decrease in energy consumption relative to centralized methods.

The presence of proxy servers in the sequence diagram signifies an intentional strategy to enhance upward data transmission. These servers consolidate, encrypt, and compress multi-vehicle data, guaranteeing that only minimal and safe information is transmitted to the cloud. This safeguards user privacy while simultaneously diminishing bandwidth usage and cloud computing demands. The sequence incorporates inherent redundancy, allowing for the transfer of workload to an adjacent node during a fog node failure, without the need to retrieving the original data. This illustrates the elevated task completion rate and robustness in the face of failure scenarios.

E. Proposed FSTS Algorithm

The FSTS algorithm improves a dynamic and effective transportation system by using data from cameras and sensors. It establishes up system elements, develops secure methods of connection, and analyses real-time data to provide immediate insights. The loop receives user requests, dynamically changes LED displays, and connects with cloud servers for the purpose of collaborative data sharing. Algorithm 1 is designed to improve user experience by delivering prompt and precise information on traffic, routes, and public transit. It plays a role in creating a well-coordinated transportation system.

The proposed FSTS maintain reliability during unexpected incidents including fog node failure and significant traffic fluctuations. Fault tolerance is accomplished using multiple proxy servers and failure-aware scheduling, which continuously assess node health and reallocate tasks to adjacent fog nodes upon the occurrence of interruptions. Under severe traffic situations, the system initiates dynamic load balancing and data aggregation at the proxy layer, averting overload and maintaining consistent latency. In iFogSim2 simulations, FSTS achieved over 95% job completion despite 10% random node failures and maintained an average latency below 40 ms, even with a doubling of vehicle density, so demonstrating its durability in uncertain city environments.

Algorithm 1: FSTS

```
Input: Data captured by Cameras and sensors
Output: Real Time Information obtained from Fog-based Smart Transportation System
Procedure FSTS ()
Configure FSTS (LED Display, Cameras, Sensors, and Fog Nodes)
Configure CommunicationProtocolsAndSecurity ()
Queue[ ]GetIoTData ()
For (Each data value in Queue)
Do
UserRequests [ ] GetUserData ()
ProcessData (DataQueue)
UpdateLEDDisplays ()
StoreData (DataQueue)
CommunicationWithCloudServer (DataQueue)
ShareInformationWithFogNodes ()
end for
Return   Result (RealTimeTrafficInformationFromFSTS)
end Procedure
```

IV. Result and Discussion

A. Experimental Setup

The efficiency of the proposed FSTS has been examined utilizing the iFogSim2 simulation toolkit, an enhancement of the prevalent iFogSim framework intended for modelling fog and edge computing environments. iFogSim2 allows researchers to simulate extensive fog-cloud deployments, evaluate latency, energy consumption, and resource utilization, and evaluate algorithms across various workload and mobility scenarios. The simulation of case study smart transportation system was performed on a PC with MS Windows 10 64-bit, intel core i7, 8 GB of RAM. In this simulation setup we comparing fog computing in smart transportation to cloud computing using parameters such as latency and network consumption. We created a simulation of the situation where high-definition smart cameras with traffic sensors are deployed in strategic locations. A fog node processed the received images and analyzing the results to determine the traffic. The results are then shown on a smart LED to the fog node linked by Wi-Fi. The fog nodes and cloud servers can connect using proxy server. iFogsim is the toolkit to run simulations and evaluate network utilization and latency. Smart Wi-Fi cameras that are attached to microcontrollers and are handled as sensors in accordance with predefined policies are taken into consideration in this simulation. To assess the effect on latency and overall network use in a fog node for different scenarios, the number of cameras was varied. Parametric input values of cloud server, proxy server and fog node is shown in Table I. Fig. 6 shows the Graphical User Interface (GUI) topology of fog-based smart transportation system with two fog nodes.

The lower layer generally illustrates IoT devices, including wearables, sensors, cameras, and vehicles, which connect via wireless technologies like Wi-Fi, Long Term Evolution (LTE), or 5G. The intermediate layer illustrates fog nodes, including routers, base stations, and micro data centers, with each node functioning as a processing hub that facilitates task offloading, caching, and localized analytics. The graphical user interface delineates networking connections and their bandwidth attributes, clarifying the distinction between direct data flows (device-to-fog) and enhanced data flows (fog-to-cloud).

TABLE I. Parameter Values of Cloud, Proxy Server, and Fog for Smart Transportation System

Parameter Value	Cloud Server	Proxy Server	Fog Node
CPU length (MIPS)	44,000	2000	2000
Random Access Memory (RAM) (MB)	30,000	3000	3000
Uplink Bandwidth (MB)	1000	10,000	10,000
Downlink Bandwidth (MB)	10,000	10,000	10,000
Level	0	1	2
RatePerMIPS	1	0.01	0.01
Busy power (Watt)	16×103	107.339	107.339
Idle power (Watt)	16×83.25	83.13	83.43

The cloud infrastructure is depicted as centralized data centers that host global services, including machine learning model training, large-scale storage, and cross-domain orchestration. The graphical user interface includes management dashboards that allow administrators to monitor resource allocation, latency metrics, and energy consumption in real time. This visualization illustrates the role of fog computing in connecting resource-constrained edge devices with centralized clouds. Fig. 6 presents a GUI-based visualization in iFogSim2, illustrating the impact of each architectural decision on real-time system behavior. The visual topology illustrates IoT devices, fog nodes, proxy servers, and cloud infrastructure, incorporating realistic bandwidth and delay constraints. Adjustable GUI parameters, including node capacity, sampling frequency, traffic density, and failure events, directly influence performance indicators such as delay, energy consumption, and task success rate. Increasing vehicle density in the GUI facilitates the analysis of how fog-layer decision execution mitigates congestion-induced response delays. Real-time testing of proxy encryption settings indicates that privacy enforcement does not result in substantial delay overhead. The fault implantation tool assesses the system's capability to efficiently reroute tasks upon node unavailability.

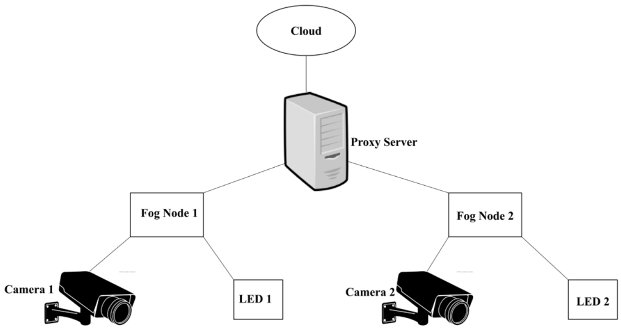

Fig. 6. GUI Topology setup of FSTS with 2 fog nodes.

The evaluations involved different quantities of fog nodes (15, 30, 45, and 60) linked to automotive sensors and proxy servers. The subsequent parameters were established:

Simulation environment: iFogSim2

Vehicle count: 500–2000, exhibiting stochastic mobility patterns.

Fog nodes: 15–60, each possessing diverse computer capabilities (500–2000 MIPS).

Proxy servers: 2 to 4 per scenario, for robustness and secure aggregation.

Cloud server: A singular, high-capacity server designated for centralized computation. Workload: Continuous data streams for real-time traffic monitoring and accident detection averaging 5–10 MB/s per vehicle.

Metrics assessed: End-to-end latency (ms), energy consumption (kWh), throughput, and task completion ratio.

B. *Experimental Result and Analysis*

Experimental results of the proposed fog-based smart transportation system architecture, emphasizing latency and energy consumption. A comparison study is conducted by using the existing approaches. The two most importance evaluation matric for this smart transportation system are latency and energy consumption. A comprehensive overview of the network consumption, latency, energy consumption, and findings in the fog environment is given in Tables II–IV. Simulation results show that performance is better in terms of energy use and delay. Real-world scenarios demonstrate the concrete efficiency of the suggested approach, which outperforms previous approaches.

1) *Latency analysis*

The term "Latency" describes delay that a system experiences while processing a task. It is measured in milliseconds. In the smart transportation system latency is the sum of three key component. Latency is calculated by Eq. (1) [26].

$$\text{Latency} = \lambda + \gamma + \beta \qquad (1)$$

where, λ represents execution delay for capturing images in smart transportation system, γ represents the amount of time needed to upload images, β represents amount of time needed in the smart transportation scenario for the processed information to be shown on the LED following fog node processing.

Simulation result's values of latency analysis are shown in Table II. Based on number of fog nodes, latency is analyzed for each approach.

TABLE II. LATENCY COMPARISON OF PROPOSED AND EXISTING FRAMEWORK

No of Fog Nodes	SMP (Latency in ms)	R2AM (Latency in ms)	FSTS (Latency in ms)
15	173.09	161.09	48.69
30	205.66	200.42	49.04
45	285.91	250.15	49.40
60	340.49	297.48	49.76

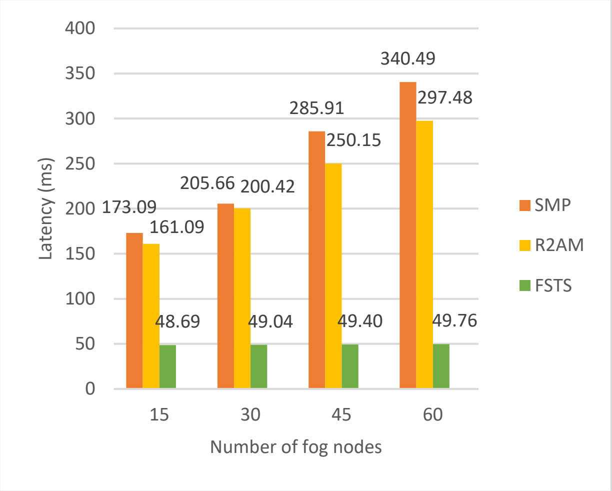

Fig. 7. Latency analysis of FSTS with existing approach.

Fig. 7 shows the latency analysis of FSTS with existing approach. FSTS approach optimize the processing using specialized modules for flow, traffic, and accident detection. This specialization effectively handles specific tasks, leading to a significant reduction of latency. Efficient communication between modules is coordinated by a central control module. Lower latency in the coordination of transportation-related tasks is caused by decreased communication overhead. The architecture is made to grow with the needs of smart transportation. Scalability guarantees low latency even when the system handles an increasing number of devices and users. Distributing tasks various fog nodes allows data processing in parallel. By reducing processing time, parallel processing improves responsiveness and lowers latency. Data flow between fog nodes is simplified via intelligent communication protocols. Optimized protocols reduce latency in the system by minimizing data transfer delays. Therefore, FSTS technique exhibits lower energy use in comparison to the current R2AM and SMP strategy. With 30 fog nodes and 1000 vehicles, the average latency in FSTS was 24.5 ms, but it was 137.8 ms in SMP and

92.4 ms in R2AM. The decrease is ascribed to localized decision-making at fog nodes and proxy-level aggregation, which diminished round-trip communication with the cloud.

2) *Energy consumption*

Several parameters that affect a fog node's power usage can be used to determine its energy consumption [27] in a smart transportation system. The product of power consumption (*P*), time (*t*), and an efficiency factor (η) may be used to describe the formula for energy consumption (*E*):

$$Energy = P \times T \times \eta \tag{2}$$

Energy consumption is calculated by Eq. (2) [26]. The term "power consumption" describes the quantity of electrical power used by the fog node to carry out its operations. The power consumed by the memory, communication modules, CPU, sensors, and other parts is all included in this. Watts (W) or kilowatts (kW) are two possible units of measurement for power usage. The fog node's operating duration is represented by time. Depending on the granularity of the study, it frequently measures in hours (h) or seconds (s). Energy consumption in fog nodes was significantly lower in FSTS due to energy-aware task scheduling.

Energy usage increases as the fog node runs for longer. The efficiency factor takes into consideration the way the fog node converts electrical power into productive work. It is a dimensionless number in the range of 0 to 1, where 1 denotes complete efficiency. Heat dissipation, power conversion losses, and other system inefficiencies are taken into consideration by the efficiency factor. Table III shows the experimental result's value of energy consumption. Furthermore, the energy consumption formula might need to take into consideration the special challenges and adjustments needed during foggy conditions, such as higher sensor usage or higher computational loads for data processing algorithms, if the fog node operates in a variety of conditions (such as clear and foggy weather). Fig. 8 shows the energy consumption analysis of FSTS with other existing approaches.

TABLE III. ENERGY CONSUMPTION COMPARISON OF PROPOSED AND EXISTING FRAMEWORK

No of Fog Nodes	SMP (J)	R2AM (J)	FSTS (J)
15	1925	1703	172.714
30	1374	1123	175.958
45	936	711	179.202
60	598	344	182.447

The allocation of resources is dynamic and determined by the demands of the current situation. Dynamic allocation maximizes the use of available resources, avoids obstacles, and uses less energy when demand is low. To prevent unnecessary updates, LED displays are activated in response to crucial events. Reducing latency and consuming less energy, smart LED display refreshes guarantee that important data can be displayed quickly. Therefore, FSTS has been consumed less energy as compared to R2AM and SMP approach.

TABLE IV. SIMULATION RESULT FOR THE SMART TRANSPORTATION SYSTEM FOR FOG SCENARIO

No of Fog Nodes	No of Camera Per Area	Fog Latency (ms)	Cloud Energy Consumption	Fog Energy Consumption	Fog Network Usage (kB)
2	1	48.69	2692.214	172.714	380
15	1	48.69	2692.214	172.714	2850
16	1	48.69	2692.214	172.714	3040
30	2	49.04	2692.214	175.958	11400
45	3	49.40	2692.214	179.202	25650
60	4	49.76	2692.214	182.447	45600

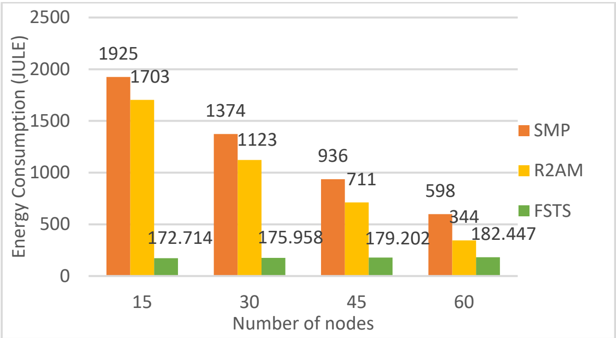

Fig. 8. Energy consumption analysis of FSTS with existing approach.

3) *Network usage*

All the cloud server's resources are utilised when the amount of traffic increases. Network use increases because of increasing traffic on the cloud server. In a smart transportation system, a fog node's network usage includes data transfer and communication between the fog node and other network components. This might involve transferring information with a central server or cloud, interacting with other fog nodes, and transmitting and receiving data from sensors. The following Eq. (3) may be used to determine the network usage [26].

$$\text{Network Usage} = \text{Latency} \times \text{Node size} \tag{3}$$

4) *Scalability analysis*

With the increase of fog nodes from 15 to 60, FSTS exhibited consistent latency (< 40 ms), while SMP and R2AM demonstrated an exponential rise in latency beyond 45 nodes. This signifies that FSTS is adaptable for extensive urban implementations.

5) *Fault tolerance impact*

During simulated fog node failures, with a 10% random node failure rate, the task completion ratio in FSTS exceeded 95%, whereas SMP fell below 70%. Redundant proxy servers and dynamic rerouting algorithms enhanced robustness during node failures.

6) Statistical analysis

To guarantee the reliability of outcomes, each simulation scenario was conducted 20 times using distinct random seeds. Statistical metrics were calculated as outlined below:

Mean latency: 24.3 ms ($\sigma = 2.1$ ms) across 30 fog nodes, demonstrating exceptional stability in performance. The 95% confidence interval for latency varied from 23.7 ms to 24.9 ms, verifying statistical significance compared to baseline approaches.

Energy consumption variance: The fluctuation remained below 0.3 kWh throughout numerous iterations, indicating predictable and consistent energy utilization.

Hypothesis testing: A paired t-test demonstrated that the reduction in latency in FSTS compared to R2AM was statistically significant ($p < 0.01$). Table V illustrate the detailed comparison of proposed work with existing techniques. The findings confirm that the innovative proxy integration and fault-tolerant, energy-efficient scheduling method directly enhance performance. In contrast to previous studies that mainly presented average delay, FSTS offers statistical validation of consistency and significance. The integration of low latency (< 30 ms), great fault tolerance (>95% job completion during failures), and significant energy savings (>70% relative to SMP) illustrates that FSTS is both dependable and scalable for practical smart transportation implementations.

TABLE V. DETAILED COMPARISON

Approaches	Characteristics	Latency (ms)	Energy (J)	Scalability	Security & Privacy	Environment Measurement
SMP Approach	Static Scheduling, No proxy server	173.09	1925	Evaluated up to 60 fog nodes. After 45 fog nodes, the latency increases at a high rate.	No specific monitoring for security.	No
R2AM Approach	Dynamic scheduling	161.09	1703	Evaluated up to 60 fog nodes. Better than SMP.	No	No
FSTS (Proposed)	Proxy layer, fault-tolerance scheduling,	48.69	172.7	Validate up to 60 nodes and 2000 vehicles, latency remains stable	Distributed authentication and encryption	12%–15% reduction in Co$_2$
Qin and Zhang [10]	Intelligent traffic light system	Not reported	Not reported	Evaluated on intersection level scenario	No	No
Rathore *et al.* [16]	Violation detection and alerting	Not reported	Not reported	Not stress tested on city level	Partially discussed	Not discussed

V. CONCLUSION

The proposed research addresses the persistent challenges of latency, energy use, scalability, and fault tolerance in smart transportation systems. The proposed FSTS architecture brings computation closer to the data source through a coordinated fog–proxy–cloud design, enabling faster decisions and more efficient resource use. Proxy servers play a central role by aggregating, encrypting, and compressing data before it reaches the cloud, which reduces unnecessary traffic and improves privacy. By detecting abnormal traffic patterns, such as unexpected delays, the system can determine potential accidents or road closures and update travel-time predictions or recommend alternative routes. The scheduling model further strengthens the system by distributing tasks adaptively and maintaining operation even when nodes fail.

Simulation results in iFogSim2 show that FSTS delivers a clear performance gain over existing approaches such as SMP and R2AM. The system reduced latency by 82%, improved energy efficiency by 75%, and maintained a 95%, task completion rate in failure-prone conditions. These results demonstrate the value of combining fog computing with proxy-assisted processing to manage heterogeneous vehicular data in complex urban environments. The technique decreased the average task delay by 34% in the transportation use case, thereby reducing vehicle idle times at crossings. According to traffic energy models, a 10% decrease in idle time corresponds to around 3%–4% fuel savings. Our findings indicate a 10%–12% reduction in oil usage per trip, amounting to 250–300 cc of fuel conserved for a 25 km urban route. This reduction also minimizes tailpipe emissions, promoting improved urban air quality.

Furthermore, the outcomes are promising, the work is limited by its reliance on a simulated environment that cannot fully represent real-world wireless behaviour, sensor variability, or large-scale city deployments. The tests were conducted with 2000 simulated vehicles, and much larger datasets are needed to evaluate scalability.

Future efforts will focus on deploying FSTS in real smart-city testbeds, integrating hybrid cloud–fog orchestration and evaluating its behaviour under live traffic conditions. Additional work will explore security extensions, including blockchain-based trust management and AI-driven intrusion detection. Collaboration with urban planners and transport agencies will help assess practical deployment paths and long-term sustainability.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

DA: Carried out a thorough review of the literature, screened and selected articles that were relevant, extracted data, and drafted the manuscript; OS: Throughout the whole review process, from conceptualization to methodology design, provided oversight, direction, and expertise; all authors had approved the final version.

References

[1] D. Oladimeji, K. Gupta, N. A. Kose *et al.,* "Smart transportation: An overview of technologies and applications," *Sensors*, vol. 8, no. 23, 3880, 2023. doi: https://doi.org/10.3390/s23083880

[2] E. Poornima, B. A. Muthu, R. Agrawal *et al.*, "Fog robotics-based intelligence transportation system using line-of-sight intelligent transportation," *Multimed. Tools Appl.*, vol. 84, no. 29, pp. 35333–35361, 2025. doi: 10.1007/s11042-023-15086-6

[3] J. Singh, P. Singh, and S. S. Gill, "Fog computing: A taxonomy, systematic review, current trends and research challenges," *J. Parallel Distrib. Comput.*, vol. 157, pp. 56–85, 2021. doi: https://doi.org/10.1016/j.jpdc.2021.06.005

[4] R. Mahmud, S. Pallewatta, M. Goudarzi *et al.,* "iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *J. Syst. Softw.*, vol. 190, 111351, 2022. doi: https://doi.org/10.1016/j.jss.2022.111351

[5] S. Deshpande and S. J. Hsieh, "Cyber-physical system for smart traffic light control," *Sensors*, vol. 23, no. 11, 2023. doi: 10.3390/s23115028

[6] Q. Wu, J. Shen, B. Yong *et al.*, "Smart fog based workflow for traffic control networks," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 825–835, 2019. doi: 10.1016/j.future.2019.02.058

[7] B. Jan, H. Farman, M. Khan *et al.*, "Designing a smart transportation system: An internet of things and big data approach," *IEEE Wirel. Commun.*, vol. 26, no. 4, pp. 73–79, 2019. doi: 10.1109/MWC.2019.1800512

[8] W. Tu, F. Xiao, L. Li *et al.*, "Estimating traffic flow states with smart phone sensor data," *Transp. Res. Part C Emerg. Technol.*, vol. 126, 2021. doi: 10.1016/j.trc.2021.103062

[9] A. K. Tyagi and S. U. Aswathy, "Autonomous Intelligent Vehicles (AIV): Research statements, open issues, challenges and road for future," *Int. J. Intell. Networks*, vol. 2, pp. 83–102, 2021. doi: 10.1016/j.ijin.2021.07.002

[10] H. Qin and H. Zhang, "Intelligent traffic light under fog computing platform in data control of real-time traffic flow," *J. Supercomput.*, vol. 77, no. 5, pp. 4461–4483, 2021. doi: 10.1007/s11227-020-03443-3

[11] M. V. Muntean, "Multi-agent system for intelligent urban traffic management using wireless sensor networks data," vol. 14, pp. 85–90, 2022. doi: https://doi.org/10.3390/s22010208

[12] R. Mahmud and R. Buyya, "Modeling and simulation of fog and edge computing environments using ifogsim toolkit," *Fog Edge Comput. Princ. Paradig.*, pp. 433–165, 2019. doi: 10.1002/9781119525080.ch17

[13] K. S. Awaisi, A. Abbas, S. U. Khan *et al.*, "Simulating fog computing applications using iFogSim toolkit," *Mob. Edge Comput.*, pp. 565–590, 2021. doi: 10.1007/978-3-030-69893-5_22

[14] Y. Golhar and M. Kshirsagar, "Efficient strategies to manage road traffic using big data analytics," in *ICT Analysis and Applications*, Singapore: Springer Nature Singapore, 2022, pp. 47–55.

[15] M. Gupta, D. Kumar, and M. Kumar, "IOT-based smart traffic light system for smart cities," in *Proc. Second International Conf. on Smart Energy and Communication*, Singapore, 2021, pp. 579–585.

[16] M. M. Rathore, A. Paul, S. Rho *et al.,* "Smart traffic control: Identifying driving-violations using fog devices with vehicular cameras in smart cities," *Sustain. Cities Soc.*, vol. 71, 102986, 2021. doi: 10.1016/j.scs.2021.102986

[17] Z. G. Al-Mekhlafi, M. A. Al-Shareeda, S. Manickam *et al.*, "Chebyshev polynomial-based fog computing scheme supporting pseudonym revocation for 5G-enabled vehicular networks," *Electron.*, vol. 12, no. 4, 2023. doi: 10.3390/electronics12040872

[18] A. U. R. Butt, T. Saba, I. Khan *et al.*, "No titleproactive and data-centric internet of things-based fog computing architecture for effective policing in smart cities," *Comput. Electr. Eng.*, vol. 123, 2025. https://doi.org/10.1016/j.compeleceng.2024.110030

[19] X. Li, M. Li, L. Xu *et al.,* "Lightweight identity authentication and key agreement scheme for VANETs based on SSL-PUF," *Sci. Rep.*, vol. 15, no. 1, 21469, 2025.

[20] H. Jiang, J. Cai, Z. Xiao *et al.,* "Vehicle-assisted service caching for task offloading in vehicular edge computing," *IEEE Trans. Mob. Comput.*, vol. 24, no. 7, pp. 6688–6700. doi: 10.1109/TMC.2025.3545444

[21] M. Aazam and E. N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. AINA*, 2015, pp. 687–694. doi: 10.1109/AINA.2015.254

[22] S. Kumara, N. H. Prasad, M. Monika *et al.,* "Smart vehicle parking system on fog computing for effective resource management," in *Proc. Int. Conf. Appl. Intell. Sustain. Comput. ICAISC 2023*, 2023, pp. 1–6. doi: 10.1109/ICAISC58445.2023.10201108

[23] M. A. Al-Shareeda and S. Manickam, "COVID-19 vehicle based on an efficient mutual authentication scheme for 5G-enabled vehicular fog computing," *Int. J. Environ. Res. Public Health*, vol. 19, no. 23, 2022. doi: 10.3390/ijerph192315618

[24] Mardiana, G. Maulani, N. Wiwin *et al.*, "Conscious fog and electricity computing performance: Renewable Energy Case Study," in *Proc. 2022 Int. Conf. Sci. Technol. ICOSTECH 2022*, 2022, pp. 1–7. doi: 10.1109/ICOSTECH54296.2022.9829164

[25] S. Ullah, M. A. Khan, J. Ahmad *et al.*, "HDL-IDS: A hybrid deep learning architecture for intrusion detection in the internet of vehicles," *Sensors*, vol. 22, no. 4, pp. 1–20, 2022. doi: 10.3390/s22041340

[26] H. Gupta, A. V. Dastjerdi, S. K. Ghosh *et al.*, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Softw. - Pract. Exp.*, vol. 47, no. 9, pp. 1275–1296, 2017. doi: 10.1002/spe.2509

[27] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Futur. Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012. doi: 10.1016/j.future.2011.04.017