A Predictive Serverless Service Orchestration for 6G-Enabled Vehicular Edge Computing

Shahrokh Vahabi[1] and Roberto Riggio[2]

[1]CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Parma, Italy
[2]Università Politecnica delle Marche, Ancona, Italy

***Abstract*—The rapid proliferation of data-intensive vehicular applications necessitates ultra-low latency and scalable computational resources. Vehicular Edge Computing (VEC) addresses these demands, yet conventional serverless edge architectures suffer from severe cold start penalties and suboptimal resource allocation under highly dynamic mobility patterns. In this paper, we propose a predictive service orchestration framework for 6G-enabled serverless vehicular networks that integrates a three-tier computational hierarchy spanning Vehicle-to-Vehicle (V2V), terrestrial RoadSide Units (RSUs), and Low Earth Orbit (LEO) satellites. Our framework introduces a proactive mechanism to effectively suppress cold start latency by evaluating zero-shot time series foundation models, which require no retraining or fine-tuning for accurate service invocation load forecasting, combined with a lightweight kinematic extrapolation model for mobility-aware trajectory prediction. This enables proactive pre-warming of containers to optimal target RSUs before vehicle arrival. For task execution, we employ a First-In-First-Out (FIFO) function scheduling policy. Designed to optimize multiple system parameters, our approach jointly minimizes end-to-end service delay and overall network energy expenditure. Extensive simulations across realistic urban topologies demonstrate that, compared to non-predictive reactive baseline using FIFO scheduling, the proposed predictive pre-warming orchestration reduces response latency by $\sim 35\%$ and cuts total energy consumption by $\sim 50\%$.**

***Index Terms*—Vehicular edge computing, serverless computing, cold start, TN/NTN networks, time series predictors.**

I. Introduction

The evolution toward 6G wireless communication focuses on achieving truly ubiquitous connectivity across all geographical regions, including maritime and remote rural domains [1], [2]. Non-Terrestrial Networks (NTN) have emerged as a core enabler of this vision by extending network coverage into the space and aerial dimensions. By integrating satellites and high altitude platforms, NTN architectures eliminate the coverage limitations of traditional terrestrial base stations. This seamless integration ensures that global connectivity remains uninterrupted even in environments where terrestrial infrastructure is absent or damaged. Within the NTN framework, Low Earth Orbit (LEO) satellite constellations serve as high-performance edge nodes that significantly extend the computational capabilities of the planetary network [3]. These satellites function as orbital edge servers, processing data closer to the source rather than forwarding all requests to distant cloud data centers. This sky tier of computation is particularly vital for latency-critical applications where even minor propagation delays can impact system performance [4]. By deploying edge intelligence on moving satellite constellations, 6G networks can provide a stable service environment for fast-moving ground-based users.

Vehicular Edge Computing (VEC) is one of the most prominent applications of this integrated architecture, where connected vehicles offload sensing and perception tasks to the edge. By utilizing VEC, vehicles can overcome their local resource constraints and improve safety through cooperative awareness [5], [6]. However, traditional VEC relies heavily on RoadSide Units (RSUs), which are primarily restricted to urban intersections and busy highways. LEO-integrated 6G networks solve this issue by providing a continuous edge presence that supports vehicular services in RSU-denied environments. To manage the complexity of these diverse vehicular services, modern edge architectures utilize containerized technologies and pod-based orchestration. Containers provide the necessary portability and isolation to run heterogeneous microservices across different hardware tiers ranging from terrestrial RSUs to LEO satellites. These pod-based deployments enable the adoption of Function-as-a-Service (FaaS) paradigms, which allow for granular resource scaling. In a FaaS model, software components are executed as independent stateless functions that scale automatically based on incoming requests [7], [8].

Despite the benefits of FaaS and pod-based architectures, the cold start problem remains a fundamental challenge for latency-critical vehicular missions. When a function is requested on a node where it is not currently active, the system incurs a significant delay while initializing the container environment. This cold start penalty can exceed the strict real-time requirements of autonomous driving services [8]. Furthermore, the high mobility of vehicles and satellites necessitates frequent service handovers, which further complicate resource availability. Existing reactive orchestration solutions are often insufficient for the high-speed dynamics of 6G vehicular networks and usually lead to service interruptions or excessive energy waste. To address these challenges, researchers have recently introduced zero-shot and few-shot time series foundation models, which demonstrate remarkable predictive capabilities without requiring extensive fine-tuning. These foundation models are pretrained on vast amounts of diverse data, allowing them to capture complex temporal dependencies across different environments. By leveraging these advancements, it is now possible to achieve accurate forecasts of resource demands even in unseen operational scenarios [8], [9].

This paper proposes a predictive service orchestration framework based on time series foundation models to forecast service invocation loads in heterogeneous vehicular networks. The predicted demand is used to proactively pre-warm vehicular edge pods, reducing cold start latency and improving energy efficiency. In parallel, a mobility-aware placement strategy uses Euclidean distance to select the most suitable RSU or LEO satellite along vehicle paths, ensuring that function pods are instantiated only at optimal edge nodes. The combined forecasting and mobility-aware orchestration enables proactive, load and location-driven service management.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed predictive service orchestration framework. Section IV presents the simulation setup and results. Section V concludes the paper and discusses future directions.

II. Related Work

A. Prediction-Based Approaches in VEC

Recent advances in vehicular edge computing underscore the need for efficient deep neural network inference by distributing computational load across heterogeneous network components. Li et al. addressed the limitations of onboard processors by introducing an adaptive partitioning and offloading framework that utilizes an *Extreme Gradient Boosting* algorithm to predict layer-by-layer execution latency [10].

Zhao et al. introduced a vehicular edge computing offloading framework named *STRIVE* that integrates digital twin technology with generative adversarial networks to overcome the limitations of high-speed mobility and rapidly changing network topologies [11]. By leveraging a Wasserstein generative adversarial network for trajectory prediction, the system maintains a virtual representation of physical entities to proactively reduce the complexity of the task decision space.

Hasan et al. explored the convergence of federated learning with computational offloading and resource management within 6$^{\text{th}}$-generation Vehicle-to-Everything (V2X) networks [12]. Their study addresses fundamental issues, including hardware incompatibility and data privacy, by proposing a three-layer system architecture where a central cloud server trains models while edge nodes manage execution and updates.

Da Costa et al. introduced a mobility and deadline-aware task scheduling mechanism for VEC environments termed *MARINA* [13]. This approach addresses the challenges of high dynamic network topologies and varying task requirements by utilizing a hybrid architecture that incorporates both vehicles and base stations into resource pools. To ensure efficient decision-making, the system leverages an LSTM network to predict future resource availability in vehicular clouds based on spatiotemporal mobility data.

B. LEO Satellite–Assisted VEC

Wu et al. investigated energy-efficient precoding designs and update strategies for LEO satellite communications to address the challenges of frequent updates in highly mobile environments [14]. The authors formulated an optimization problem aimed at maximizing energy efficiency while adhering to quality of service requirements.

Li et al. presented a joint task offloading and resource allocation strategy tailored for hybrid multi-access edge computing-enabled LEO satellite networks [15]. The methodology accommodates two distinct service categories, consisting of delay sensitive edgy cloud processing and data-intensive cloudy edge computation. By integrating an edge cloud collaborative architecture, the system enables workloads to be executed locally at satellites or offloaded to a ground server via wireless backhaul links.

The advancement of LEO satellite networks has evolved to incorporate edge computing paradigms for enhancing service delivery to user equipment in isolated environments [16]. A specific cooperative framework allows for the collaborative processing of divisible tasks between ground terminals and orbital servers. This model leverages a specialized cost function to provide a fair assessment of execution latency and energy dissipation by normalizing metrics to comparable magnitudes.

Integration of LEO satellite communication with VEC addresses the resource limitations of terrestrial infrastructure by providing global coverage and reliable data transmission [17]. Their framework explores delay minimization through a task priority scheduling scheme that categorizes vehicular workloads based on urgency and data volume. This approach utilizes a Markov decision process to model the offloading environment and employs a preemptive earliest deadline first proximal policy to determine optimal decisions.

A satellite-terrestrial integrated vehicular edge computing system addresses the coverage gaps of terrestrial roadside units in remote areas by incorporating LEO satellites as auxiliary edge servers [18]. This framework focuses on minimizing the weighted sum energy consumption of in-vehicle user equipment through the joint optimization of terminal association, task partitioning, and multi-dimensional resource allocation.

C. Positioning of this work

Existing VEC research mainly focuses on trajectory estimation and task offloading to reduce latency and energy consumption. However, prior studies largely overlook lightweight time series forecasting for predicting future function invocation counts, which is crucial for pre-warming containers/pods and reducing cold start delays in serverless environments. To address this gap, this work evaluates three zero/few-shot time series predictors: Amazon Chronos [19], TimesFM [20], and Lag-Llama [21], which avoid costly retraining and fine-tuning.

In addition, this study supports full 6G coverage by TN/NTN networks across ground and aerial infrastructures. It also employs Euclidean-based path prediction on predefined road networks to identify optimal nearby micro servers on vehicles, RSUs, and satellites for function offloading. Therefore, the main contributions of this work are summarized as follows:

1) *Ubiquitous 6G VEC Architecture:* We design a hierarchical three-tier offloading model that ensures continuous service coverage during high mobility through seamless handovers and Inter Satellite Link (ISL) routing.

2) *Foundation Driven Load Forecasting:* We implement a demand-aware predictive mechanism that monitors and forecasts future invocation loads using state-of-the-art foundation models to prioritize resource allocation.
3) *Efficient Pod Orchestration:* We demonstrate significant improvements in latency and energy consumption by proactively warming heterogeneous vehicular services before a vehicle enters a new coverage area.
4) *Comparative Predictor Analysis:* We provide the evaluation of three lightweight zero/few-shot predictors' performance in forecasting non-stationary vehicular loads.

III. Proposed Predictive Service Orchestration

A. System Model

The proposed framework operates within a hierarchical 6G-enabled VEC architecture that spans three distinct computational tiers. At the ground level, a set of RoadSide Units $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ is deployed across the urban road network in a grid formation, each equipped with lightweight edge servers running containerized serverless functions. Above the terrestrial layer, a LEO satellite constellation $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$ orbits at $\sim$ 550km altitude [22], providing ubiquitous edge coverage that extends beyond the geographic limitations of RSU infrastructure. Connecting these tiers, a fleet of connected vehicles $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ traverses the road network, generating heterogeneous computational tasks that must be offloaded to the nearest available edge resource. A designated subset $\mathcal{V}_s \subset \mathcal{V}$ of server-capable vehicles provides an additional V2V offloading tier for ultra-low latency processing.

Each vehicle $v \in \mathcal{V}$ is characterized by its real-time position $p(v,t) = (x_v(t), y_v(t))$, its instantaneous speed $s(v,t)$, and its heading angle $\theta(v,t)$ at discrete simulation step t. These kinematic attributes [23] are obtained directly from a SUMO-based traffic microsimulation that models realistic vehicular mobility across an OpenStreetMap road topology [24]. At every time step, each non-server vehicle generates a task τ drawn from a heterogeneous workload pool $\mathcal{F} = \{f_1, f_2, \ldots f_j\}$, here comprising five distinct serverless functions: image processing for object detection, LiDAR point cloud analysis, multi-frame video analytics, path planning, and multi-sensor fusion.

B. Three-Tier Offloading Decision Model

The offloading decision for each task τ generated by vehicle v at position $p(v,t)$ follows a proximity-aware and load-aware three-tier selection strategy. The system first computes the Euclidean distance between the vehicle and all candidate edge nodes [25]. For any candidate node n with position (x_n, y_n), the Euclidean distance is defined as:

$$d(v,n,t) = \sqrt{(x_v(t) - x_n)^2 + (y_v(t) - y_n)^2} \qquad (1)$$

The offloading target is selected according to the following prioritized decision logic. The system first searches for the nearest server vehicle $v_s \in \mathcal{V}_s$ within the V2V communication range of $100m$ [26]. It simultaneously identifies the nearest RSU $\hat{r} \in \mathcal{R}$ within the RSU coverage radius of $200m$ [27]. The decision process is as follows: if a nearby server-capable vehicle exists and is less than half the distance to the closest RSU, the task is offloaded via V2V to minimize propagation delay. Otherwise, if the vehicle is within a reliable $150m$ RSU range, the task is offloaded to the RSU to avoid coverage-edge instability. In all other cases, the task is forwarded to the LEO satellite tier as a fallback.

C. LEO Satellite Constellation Model

The LEO satellite constellation is modeled as a set of orbital edge servers with time-varying visibility windows. Each satellite $l_i \in \mathcal{L}$ is defined by a visibility interval during which it maintains line-of-sight communication with the ground segment. Each satellite pass lasts a finite duration determined by the orbital altitude and inclination, and successive satellites overlap by a configurable handover window W to guarantee seamless service continuity. Each satellite maintains a per-step load counter $\lambda(l_i, t)$ to track its current utilization. The best satellite for vehicle v is selected by evaluating the current association. If the vehicle already holds an active association with the current satellite and that satellite remains visible with sufficient remaining pass time, the association persists, provided that the satellite is not overloaded. When the remaining pass time falls below the handover threshold, the system triggers a proactive handover by identifying the next candidate satellite with the longest remaining visibility among the currently visible set. If the current satellite is overloaded but still visible, the system attempts Inter-Satellite Link (ISL) forwarding by routing the task to a neighboring satellite with available capacity. This ISL forwarding adds per-hop latency d_{ISL} over the laser Free Space Optical (FSO) link connecting adjacent orbital nodes. LEO nodes utilize a serverless distributed orchestrator to manage CubeSat constraints through deterministic scheduling and predictive pre-warming.

D. Latency Factor

The total end-to-end latency $L(\tau)$ for processing a task τ offloaded to target node n of type $T \in \{V2V, RSU, LEO\}$ is decomposed into four additive components:

$$L(\tau) = L_{net}(T) + L_{proc}(\tau) + L_{cold}(\tau) + L_{\text{ISL}}(\tau) \qquad (2)$$

The first component represents the round-trip network propagation delay, which depends on the offloading tier. The one-way base latency $L_{net}(T)$ varies across V2V, RSU, and LEO tiers, reflecting the inherent differences in direct inter-vehicle communication, terrestrial wireless backhaul, and satellite uplink and downlink channels, respectively. The processing component $L_{proc}(\tau)$ represents the computational time at the edge node, determined by the specific function type and the input data volume. The cold start penalty $L_{cold}(\tau)$ is incurred when the requested function container is not already initialized at the target node, requiring time to initialize the container runtime, load the function image, and establish the execution environment. The ISL component $L_{\text{ISL}}(\tau)$ equals d_{ISL} multiplied by the number of inter-satellite hops if the task was forwarded through the constellation mesh.

E. Predictive Orchestration Framework

The proposed framework includes three mechanisms:

Service Invocation Load Forecasting: The mechanism monitors historical invocation load $\lambda(r_i, t)$ at each RSU r_i using a sliding window, e.g., the last 50 time steps. Every fixed interval, e.g., 30 time steps, this load history is processed by a zero-shot time series foundation model running in a background thread, ensuring the simulation loop is not blocked. The model uses recent load values as context to generate multi-step forecasts without any task-specific fine-tuning. If the predicted average load for an RSU exceeds its recent observed average, the system triggers proactive container pre-warming at that node by initializing function containers for all vehicular services, effectively converting potential cold starts into warm invocations with minimal startup delay.

Mobility-Aware RSU Selection: The second mechanism predicts the trajectory of each vehicle to determine which RSU it will approach next. The system maintains a position history buffer of the 20 most recent coordinates for each vehicle. As illustrated in Fig. 1 (Kinematic Trajectory Prediction [13], [28]), a kinematic extrapolation model computes the vehicle's heading $\theta(v,t)$ and average speed $s(v,t)$ at the current simulation time step t from the recent position:

$$\theta(v,t) = \text{atan2}(y_v(t) - y_v(t-n), x_v(t) - x_v(t-n)) \quad (3)$$

$$s(v,t) = \frac{\sqrt{(x_v(t) - x_v(t-n))^2 + (y_v(t) - y_v(t-n))^2}}{n \cdot \Delta t} \quad (4)$$

where the parameter n represents the history length (up to a maximum of 5 recent positions), and Δt is the simulation time step. The future position is then projected forward by a prediction horizon of $h = 5$ steps:

$$\hat{x}_v(t+h) = x_v(t) + s(v,t) \cdot \cos(\theta(v,t)) \cdot \Delta t \cdot h \quad (5)$$

$$\hat{y}_v(t+h) = y_v(t) + s(v,t) \cdot \sin(\theta(v,t)) \cdot \Delta t \cdot h \quad (6)$$

The predicted target RSU is identified as the one minimizing the Euclidean distance to the projected position:

$$\hat{r} = \arg\min\{d(\hat{p}(v, t+h), r_i) : r_i \in \mathcal{R}\} \quad (7)$$

The kinematic extrapolation approach provides a lightweight and deterministic mechanism for path prediction that does not rely on the foundation model predictor. While the foundation models are employed exclusively for service invocation load forecasting, the mobility-aware RSU selection operates independently using the geometric and kinematic properties of each vehicle's recent movement history. Once the predicted RSU $\hat{r}$ is identified, the framework proactively warms all function containers at that node before the vehicle arrives.

Task Scheduling Policies: For the third mechanism, we employ a FIFO function scheduling policy. Requests are processed in their arrival order, while a zero/few-shot time series predictor and a mobility-aware kinematic extrapolation model run in the background to forecast load and vehicle trajectories. These predictions enable proactive pre-warming of containers, so that FIFO handles only the dispatching of tasks, and the predictors supply the container initialization needed to minimize cold starts and reduce both network latency and energy consumption.

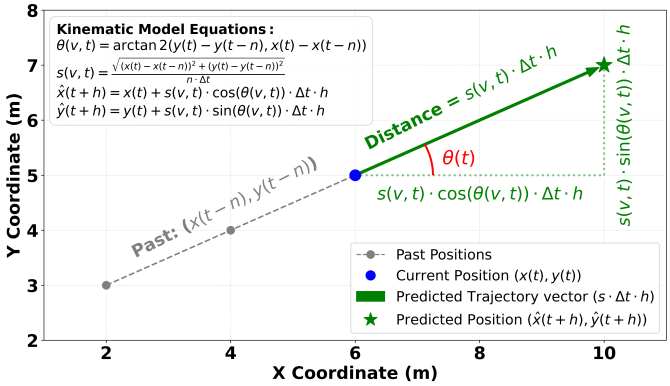

Fig. 1. Kinematic extrapolation projects a vehicle's future location from its recent trajectory.

IV. Simulation Setup and Results

A. Simulation Setup

The proposed framework is evaluated using a discrete-event simulation integrating *SUMO*, *Mininet-WiFi* [24], and *faasd*[1] as the lightweight serverless edge platform in 10 independent runs. The simulation uses a realistic urban road network from OpenStreetMap, covering approximately 2 km^2 of Ancona, Italy, with real road topology, traffic signals, and intersections. The simulation runs for 600 one-second time steps with realistic vehicle mobility generated by SUMO. As shown in 2, the environment includes 12 terrestrial RSUs (red circles) arranged in a 4×3 grid with $200m$ coverage, V2V communication up to $100m$, and three LEO satellites providing continuous NTN coverage through overlapping visibility windows. Additionally, 15% of vehicles act as server-capable nodes for V2V offloading, while all other vehicles generate one task per second. To measure power and energy consumption, the framework integrates *Scaphandre*[2]. Also, three zero-shot foundation models, Chronos [19], TimesFM [20], and Lag-Llama [21], are evaluated for service invocation load forecasting without task-specific fine-tuning. Predictions are generated every 30 simulation steps using the latest 30 load values to forecast the next 5 steps. Performance is assessed using MAPE and NRMSE metrics [9]. For trajectory prediction, a kinematic extrapolation model uses the last 5 position samples to estimate heading and average speed, predicting the vehicle position 5 steps ahead to determine the nearest RSU. This decoupling maintains edge-node responsiveness by preventing inference from blocking execution threads, while reducing cumulative network latency for a more complete measure of system-level load reduction.

[1]https://github.com/openfaas/faasd

[2]https://github.com/hubblo-org/scaphandre

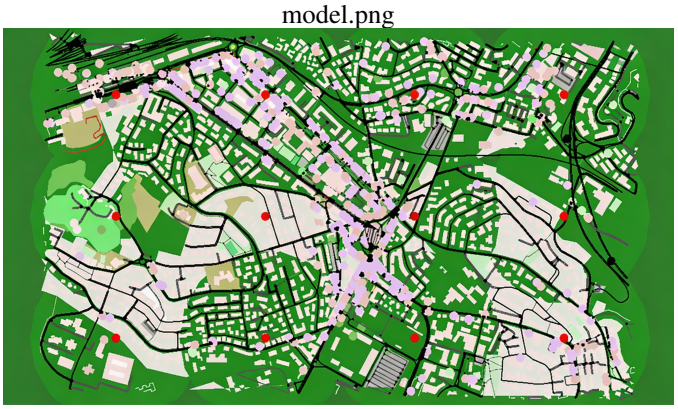

Fig. 2. Network topology over the Ancona city map, with 12 RSUs shown as red circles.

B. Simulation Results

1) Forecasting Performance: Performance is evaluated using MAPE and NRMSE. Chronos achieves the best accuracy with an MAPE of 0.09 and NRMSE of 0.26, leading to superior cold start reduction and latency performance. TimesFM follows with an MAPE of 0.21 and NRMSE of 0.30, while Lag-Llama records the highest errors (MAPE 0.26, NRMSE 0.31) but remains effective for zero-shot load prediction in dynamic vehicular environments. Overall, the low error rates without task-specific retraining demonstrate the suitability of foundation models for managing volatile edge workloads.

2) Average latency benchmark: We evaluate average execution latency across V2V, RSU, and LEO tiers to assess the efficiency of the proposed framework, highlighting the limitations of reactive serverless execution. Under a FIFO baseline without prediction, the system frequently experiences cold start stalls, resulting in average latencies of $192.0ms$ (V2V) and up to $237.4ms$ for LEO offloading, which is unsuitable for delay-sensitive vehicular applications (Fig. 3(a)).

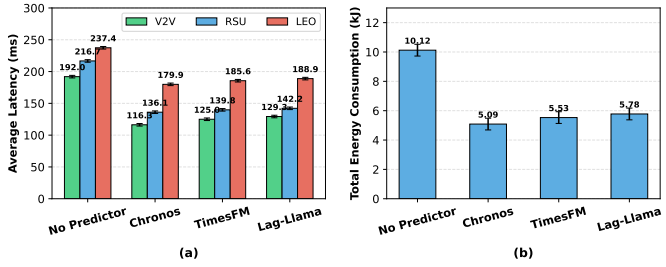

Fig. 3. Impact of predictive orchestration on (a) average latency across three tiers and (b) total energy consumption.

Integrating zero/few-shot time series foundation models for proactive pre-warming significantly reduces latency across all tiers. Driven by Chronos, V2V latency decreases to $116.3ms$, RSU to $136.1ms$, and LEO to $179.9ms$, corresponding to $\sim 39.4\%$ reduction in V2V latency over the baseline. This improvement stems from more accurate prediction of rapid load fluctuations in vehicular edge environments. Other models also improve performance: TimesFM achieves $125.0ms$ (V2V), $139.8ms$ (RSU), and $185.6ms$ (LEO), while Lag-Llama records $129.3ms$, $142.2ms$, and $188.9ms$ respectively. Although all outperform the baseline, Chronos consistently yields the lowest latency across tiers. Finally, the observed confidence intervals remain within ~ 1–2% across 10 simulation runs, indicating stable system behavior.

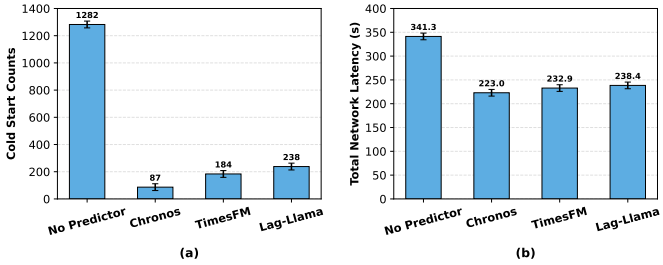

Fig. 4. Impact of predictive orchestration on (a) container cold start counts and (b) total network latency.

3) Total energy consumption: Energy efficiency is critical for battery-constrained vehicles and resource-limited edge infrastructure. The reactive baseline consumes approximately $10.12KJ$ due to repeated container bootstrapping without predictive support. As shown in Fig. 3(b), the proactive pre-warming framework improves sustainability across all predictors by avoiding these high-energy startup costs. Driven by Chronos, energy consumption is reduced to $5.09KJ$ ($\sim 49.7\%$ savings over the baseline). TimesFM and Lag-Llama reduce energy consumption to $5.53KJ$ and $5.78KJ$, respectively. These gains result primarily from eliminating repeated cold start bootstrapping and reducing idle energy overhead across tiers. In addition, mobility-aware prediction improves tier selection, ensuring tasks are executed at the most energy-efficient nodes. Overall, the predictive orchestrator reduces LEO-tier energy consumption by $\sim 21\%$ and latency by $\sim 18\%$ compared to the baseline, enhancing battery efficiency, reducing satellite load, and improving service continuity.

4) Cold start counts: The main objective of the predictive orchestration engine is to eliminate serverless cold start overhead. Experimental results show that the reactive FIFO baseline suffers from severe cold start penalties, whereas foundation model–driven forecasting substantially reduces their occurrence (Fig. 4(a)). Among the evaluated models, Chronos is the most effective, reducing cold starts from 1282 to 87 ($\sim 93.2\%$ reduction). TimesFM and Lag-Llama also achieve significant improvements, with 184 ($\sim 85.6\%$) and 238 ($\sim 81.4\%$) cold starts, respectively. These results confirm that proactive container pre-warming driven by accurate load forecasting is the key mechanism for suppressing cold start penalties under FIFO scheduling, with predictor accuracy being the main performance differentiator.

5) Total Network Latency: Total latency includes propagation, processing, and cold start delays (Section III-D). As shown in Fig. 4(b), the reactive baseline yields the highest total latency of $341.3s$ due to frequent cold starts and lack of

proactive management. Predictive orchestration under FIFO scheduling significantly reduces latency. Driven by Chronos, total latency drops to $223.0s$ ($\sim 34.7\%$ reduction). TimesFM and Lag-Llama further reduce it to $232.9s$ and $238.4s$, respectively. Overall, these results demonstrate that accurate load forecasting is key to eliminating cold start overhead and fully exploiting the three-tier offloading architecture. The NTN tier handled $\sim 22.6\%$ of offloads, ensuring resilience during terrestrial gaps. Using a break-before-make protocol [29], transitions achieved an $\sim 88.9\%$ success rate. Although LEO integration added $\sim 34ms$ latency, zero-shot predictors eliminated cold start delays during tier shifts.

V. Conclusion and Future Study

This work proposes a proactive orchestration framework for serverless vehicular edge environments, combining zero-shot load prediction and mobility-aware forecasting to reduce cold starts while improving latency and energy efficiency. Experiments show that models such as Chronos, paired with FIFO scheduling, significantly reduce delay and energy use without task-specific retraining.

Future work will develop a multi-objective optimization framework for joint function placement and resource allocation, evaluate scalability in heterogeneous and dense environments, and adopt zero-shot trajectory forecasting to better model non-linear dynamics.

Acknowledgment

This work was funded by the European Union under the Horizon Europe programme through the SNS JU (Grant Agreement No. 101192912, NexaSphere). Views expressed are those of the authors and do not necessarily reflect those of the EU or the SNS JU.

References

[1] S. Prasad Tera, R. Chinthaginjala, G. Pau *et al.*, "Toward 6g: An overview of the next generation of intelligent network connectivity," *IEEE Access*, vol. 13, pp. 925–961, 2025.

[2] N. Nomikos, A. Giannopoulos, A. Kalafatelis *et al.*, "Improving connectivity in 6g maritime communication networks with uav swarms," *IEEE Access*, vol. 12, pp. 18 739–18 751, 2024.

[3] M. Harounabadi and T. Heyn, "Toward integration of 6g-ntn to terrestrial mobile networks: Research and standardization aspects," *IEEE Wireless Communications*, vol. 30, no. 6, pp. 20–26, 2023.

[4] D. B. Tobias Pfandzelter, "Komet: A serverless platform for low-earth orbit edge services," in *SoCC '24: Proceedings of the 2024 ACM Symposium on Cloud Computing*, 2024, pp. 866–882.

[5] C. Chen, C. Wang, B. Liu *et al.*, "Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 13 023–13 034, 2023.

[6] F. Zeng, Z. Zhang, and J. Wu, "Task offloading delay minimization in vehicular edge computing based on vehicle trajectory prediction," *Digital Communications and Networks*, vol. 11, no. 2, pp. 537–546, 2025.

[7] S. Vahabi, F. Righetti, C. Vallati *et al.*, "Energy-efficient resource management for real-time applications in faas edge computing platforms," in *Proceedings of the 16th IEEE/ACM International Conference on Utility and Cloud Computing*, ser. UCC '23. Association for Computing Machinery, 2024.

[8] S. Vahabi, F. Righetti, C. Vallati *et al.*, "A predictive hybrid scheduler to reduce completion times of heterogeneous functions in faas platforms," in *2025 3rd International Conference on Federated Learning Technologies and Applications (FLTA)*, 2025, pp. 456–462.

[9] S. Vahabi, F. Righetti, C. Vallati *et al.*, "The impact of prediction models on energy-aware resource management in faas platforms," *IEEE Access*, vol. 13, pp. 85 711–85 727, 2025.

[10] C. Li, S. Liu, K. Jiang *et al.*, "Dnn inference acceleration based on adaptive task partitioning and offloading in embedded vec," *ACM Transactions on Embedded Computing Systems*, vol. 24, no. 4, pp. 1–35, 2025.

[11] L. Zhao, T. Li, E. Zhang *et al.*, "Adaptive swarm intelligent offloading based on digital twin-assisted prediction in vec," *IEEE Transactions on Mobile Computing*, vol. 23, no. 8, pp. 8158–8174, 2024.

[12] M. K. Hasan, N. Jahan, M. Z. A. Nazri *et al.*, "Federated learning for computational offloading and resource management of vehicular edge computing in 6g-v2x network," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3827–3847, 2024.

[13] J. B. D. da Costa, A. M. de Souza, R. I. Meneguette *et al.*, "Mobility and deadline-aware task scheduling mechanism for vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 11 345–11 359, 2023.

[14] S. Wu, Y. Wang, G. Sun *et al.*, "Energy and computational efficient precoding for leo satellite communications," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 1872–1877.

[15] P. Li, Y. Wang, Z. Wang *et al.*, "Joint task offloading and resource allocation strategy for hybrid mec-enabled leo satellite networks: A hierarchical game approach," *IEEE Transactions on Communications*, vol. 73, no. 5, pp. 3150–3166, 2025.

[16] A. Li, T. Zhou, T. Xu *et al.*, "Leo satellite assisted edge computing with latency and energy optimization," *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 4, pp. 2640–2653, 2025.

[17] L. Wang, J. Li, M. Dai *et al.*, "Low-earth-orbit satellite assisted edge computing for vehicular networks: A task priority-based delay minimization approach," *IEEE Internet of Things Journal*, vol. 12, no. 17, pp. 35 482–35 496, 2025.

[18] C. Li, B. Shang, J. Feng *et al.*, "Vehicular edge computing in satellite-terrestrial integrated networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 4, pp. 5290–5306, 2025.

[19] A. F. Ansari, L. Stella, C. Turkmen *et al.*, "Chronos: Learning the language of time series," *Preprint at https://doi.org/10.48550/arXiv.2403.07815*, 2024.

[20] A. Das, W. kong, R. Sen *et al.*, "A decoder-only foundation model for time-series forecasting," *Preprint at https://arxiv.org/html/2310.10688v2*, 2024.

[21] K. Rasul, A. Ashok, A. R. Williams *et al.*, "Lag-llama: Towards foundation models for probabilistic time series forecasting," *Preprint at https://arxiv.org/abs/2310.08278*, 2024.

[22] D. Bhattacherjee and A. Singla, "Network topology design at 27,000 km/hour," in *CoNEXT '19: Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, 2019, pp. 341–354.

[23] E. Lank, Y.-C. N. Cheng, and J. Ruiz, "Endpoint prediction using motion kinematics," in *CHI 2007 Proceedings. Empirical Models*, 2007, pp. 637–646.

[24] F. Alam, A. N. Toosi, M. A. Cheema *et al.*, "Serverless vehicular edge computing for the internet of vehicles," *IEEE Internet Computing*, vol. 27, no. 4, pp. 40–51, 2023.

[25] D.-g. Zhang, J. Zhang, C.-h. Ni *et al.*, "New method of edge computing-based data adaptive return in internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 2042–2052, 2024.

[26] Z. Zhao, Y. K. Mo, X. Zhang *et al.*, "An approach for testing rural los v2x minimum communication processing distance," in *2025 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2025, pp. 1–6.

[27] S. Hu, Z. Qu, B. Tang *et al.*, "Joint service placement and container retention for serverless-based vehicular edge computing," in *2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)*, 2023, pp. 2286–2294.

[28] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, pp. 1–14, 2014.

[29] S. Bhandari, T. X. Vu, N. T. Nguyen *et al.*, "Isac-enabled handover design in leo satellite networks," *IEEE Transactions on Communications*, vol. 74, pp. 5215–5231, 2026.