Workload Forecasting for Edge Network Functions in Smart Cities

Carlo Puliafito*, Antonio Virdis*, Carlo Vallati*, Nicola Tonellotto*, Marco Galante*,
Giuseppe Anastasi*, Enzo Mingozzi*†, Marcello Cucurachi‡, Leonardo Martorana‡
*Department of Information Engineering, University of Pisa, Pisa, Italy (name.surname@unipi.it)
†Department of Information Engineering, University of Florence, Florence, Italy
‡Maticmind - Zenita Group, Italy (name.surname@maticmind.it)

I. Introduction

The cloud-edge continuum distributes compute, storage, and networking resources from centralized clouds to edge nodes close to citizens, vehicles, sensors, and urban infrastructure. In this context, *Virtual Network Functions* (VNFs) are software entities that decouple network operations and functionalities from the underlying hardware. VNFs are particularly relevant for smart cities, where digital services increasingly depend on programmable connectivity functions deployed near the urban fabric. Examples include load balancers for public-service platforms, messaging brokers for Internet of Things (IoT) deployments, and security functions for protecting critical services. Smart cities are dynamic and heterogeneous environments, where user mobility, request peaks during events, energy-aware consolidation policies, and failures can rapidly change where resources are needed. Therefore, VNFs must flexibly be instantiated, migrated, and scaled across the continuum, requiring dynamic placement and orchestration to meet evolving operational needs.

Forecasting VNF workloads is a key enabler of this adaptation, because it lets orchestrators act proactively, rather than after congestion or underutilization has already affected applications and users. Reliable forecasts can support earlier scaling decisions, anticipate service relocation, and avoid keeping unnecessary edge resources active. The practical impact is direct, including lower latency for mobile users, higher robustness during demand surges, and better energy efficiency across geographically distributed infrastructure.

Existing forecasting approaches include Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing (ETS), deep-learning models, and Long Short-Term Memory (LSTM) networks. However, they often struggle with abrupt fluctuations typical of VNFs and may require manual tuning or task-specific training. This extended abstract distills our full work [1], which proposes to use time-series foundation models for VNF-workload prediction and evaluates them on CPU and network-traffic metrics, considering both a synthetic dataset that we generated and a real dataset released by Vodafone.

II. Models and Data Sources

This section summarizes the time-series foundation models and datasets considered in our study.

A. Time-Series Foundation Models

Foundation models are large neural networks pretrained on massive and heterogeneous datasets, and designed to transfer learned representations across tasks. This enables zero-shot forecasting on unseen VNFs and fine-tuning with limited local traces. Both properties are important for dynamic and complex environments such as smart cities, where collecting and retraining task-specific models is often costly or impractical. We evaluate two complementary foundation models: (i) *Lag-Llama*, which is probabilistic and outputs a predictive distribution; and (ii) *TimesFM*, which is deterministic and returns point forecasts.

B. Datasets

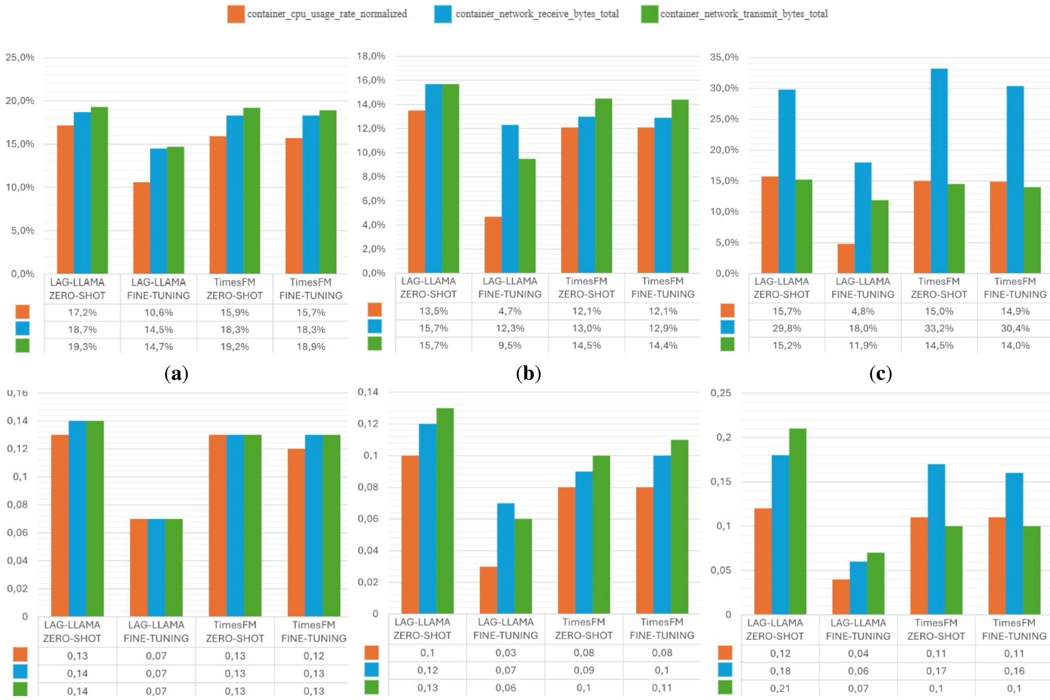
The Softwarised Network Data Zoo (*SNDZoo*) is an open initiative offering ready-to-use Software-Defined Networking (SDN) and VNF datasets, as well as a framework for running new experiments. Existing SNDZoo traces are few-minutes long and sampled every 10 seconds, which is not ideal for day-night workload studies at orchestration time scales. We therefore generated our own synthetic SNDZoo-based dataset[1], by running experiments on a testbed that executes VNFs as Docker containers and uses Prometheus for monitoring.

For each VNF, we collect 15 days of three metrics:

- `container_cpu_usage_rate_normalized`, which is the CPU consumption normalized with respect to the container allocation;
- `container_network_receive_bytes_total`, which is the cumulative bytes received by the VNF;
- `container_network_transmit_bytes_total`, which is the cumulative bytes transmitted by the VNF.

The input traffic creates a clear daytime/nighttime difference. We considered the following three VNFs: (i) *Nginx*, which is a Web server, load balancer, and Web proxy; (ii) *Mosquitto*, a Message Queuing Telemetry Transport (MQTT) broker for Internet of Things (IoT) messaging; and (iii) *Snort*, an Intrusion Detection System (IDS). They are representative of VNFs widely used in smart cities, respectively supporting service front ends, sensor/actuator messaging, and security monitoring. We pre-processed all traces through linear interpolation of short gaps and min-max scaling to the $[0, 1]$ range.

[1]https://zenodo.org/records/17780572.



(a) (b) (c) (d) (e) (f)

Fig. 1: SNDZoo – MAPE for Nginx (**a**), Mosquitto (**b**), and Snort (**c**); CRPS for Nginx (**d**), Mosquitto (**e**), and Snort (**f**).

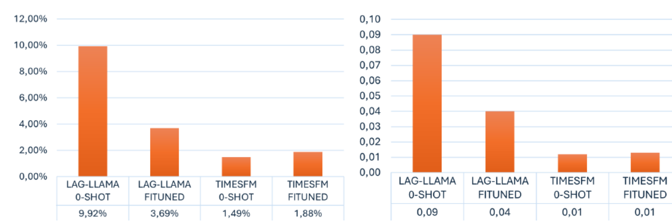
We also consider a real *Vodafone* dataset[2]. Vodafone released only CPU-usage data, with the same definition as above, for 25 VNFs randomly selected among thousands of VNFs running across 11 European countries. Each trace spans 31 days. The VNF service type is not disclosed. Moreover, unlike our SNDZoo VNFs, Vodafone VNFs run as Virtual Machines (VMs), not as containers. We randomly selected one of the 25 VNFs and applied the same pre-processing used for SNDZoo.

III. Experimental Results

We compare Lag-Llama and TimesFM in zero-shot and fine-tuned configurations. For SNDZoo, one day is used for zero-shot testing and the other 14 days for fine-tuning each VNF–metric pair. For Vodafone, zero-shot inference uses the selected 31-day trace, while models are fine-tuned only on SNDZoo and then tested on Vodafone. We use sliding-window one-step forecasting with context length 32. We measure forecasting accuracy through Mean Absolute Percentage Error (MAPE) and Continuous Ranked Probability Score (CRPS), with lower values indicating better predictions.

Fig. 1 shows that Lag-Llama consistently benefits from fine-tuning, especially on CPU. TimesFM changes little after fine-tuning. Yet, in zero-shot it is competitive because the traces contain clear day-night periodicity, a regular temporal structure that its patch-based deterministic architecture is well suited to capture. Results also show that network metrics are generally harder to forecast than CPU because they react to bursts and transient effects. Still, they are valuable complements, especially for a VNF such as Snort, whose CPU can remain low even when traffic intensity changes.

[2]https://retis.santannapisa.it/~tommaso/papers/nfvsdn22_data.tar.gz.



(a) (b)

Fig. 2: Vodafone VNF CPU usage – MAPE (**a**) and CRPS (**b**).

Fig. 2 reports the Vodafone results. This trace is easier to forecast than SNDZoo because it has a strong daily pattern, fewer bursts, and a 5-minute granularity smoothing short-lived variations. TimesFM achieves the best zero-shot performance. Lag-Llama, however, gains the most from fine-tuning. This suggests that synthetic VNF traces can specialize probabilistic models for real workloads despite differences between containers and VMs and between 1-minute and 5-minute sampling.

Overall, foundation models are promising building blocks for proactive smart-city VNF orchestration, where forecasts can anticipate load peaks, mobility-driven reallocations, failures, and energy-saving opportunities.

References

[1] C. Puliafito and et al., "Using foundation models to forecast the workload of edge network functions for green computing," in *SAYGreeN 2026*.