Beyond Car Sharing: Uncertainty-Aware Pooling of Vehicular Compute at the Network Edge

Wellington Lobato
Inria Saclay
France
viana-lobato-junior.wellington@inria.fr

Nadjib Achir
Inria Saclay
France
nadjib.achir@inria.fr

Aline Carneiro Viana
Inria Saclay
France
aline.viana@inria.fr

***Abstract*—Connected vehicles increasingly embed AI accelerators, offering a substantial yet volatile source of supplemental compute near the network edge. Unlike provisioned MEC hosts, vehicular resources are highly dynamic: vehicles may leave the cell, become locally occupied, or offer heterogeneous compute capacities. Therefore, exploiting vehicular resources requires making admission decisions without knowing the compute capacity that will be available during task execution. We present SMART, an uncertainty-aware admission mechanism that enables an ETSI Multi-access Edge Computing (MEC) orchestrator *to opportunistically exploit vehicular compute under predictive uncertainty.* SMART predicts future vehicular capacity using a Bayesian Neural Network (BNN) – whose uncertainty estimates are the best calibrated among the evaluated forecasters at the nominal 95% level, and incorporates its calibrated predictive uncertainty into a chance-constrained admission program reformulated through Sample Average Approximation (SAA) and Conditional Value-at-Risk (CVaR) approximations. Under a compute-only admission model, SMART admits 95% of tasks while maintaining a median capacity-violation rate of about 4.2%. It achieves a favorable admission–violation tradeoff compared with seven reactive, mean-only, and uncertainty-aware baselines, by approaching the performance of a compute-capacity oracle under the modeled assumptions. Finally, the sensitivity analyses show that variability in base-station compute availability is a key determinant of admission performance, highlighting the need for a calibrated admission and resource allocation mechanism tailored to opportunistic base-station compute pooling.**

***Index Terms*—Multi-access Edge Computing, Vehicular Edge Computing, Bayesian Neural Networks, Uncertainty Quantification, Chance-Constrained Optimization, Resource Orchestration.**

I. Introduction

European Telecommunications Standards Institute (ETSI) MEC offers a standardized execution environment for deploying latency-sensitive services at the network edge [1]. A MEC host provides a fixed compute capacity, whereas computational demand fluctuates with vehicle density throughout the day, leading to saturation during peak traffic and underutilization during off-peak periods [2]. At the same time, modern vehicles increasingly embed AI accelerators rated at 20–275 Tera Operations Per Second (TOPS), whose capacity often goes underutilized or idle during routine driving, parking, or charging [3]. Collectively, these underutilized resources can rival a co-located MEC server in aggregate compute capacity [4].

Recent work leverages vehicles as edge compute hosts and migrates workloads when they leave the coverage area, positioning vehicle compute as a practical extension of MEC capacity [3], [5]. However, these approaches inherit two assumptions from conventional ETSI MEC orchestration that do not hold in vehicular environments. First, the orchestrator admits tasks only after observing the available resource pool, preventing admission from exploiting resources that are not yet observable and forcing operators either to overprovision the static MEC tier or to rely on costly workload migration [5]. Second, forecasts of the future resource pool are typically reduced to point estimates, causing admission decisions to treat these predictions as exact [6], [7] without distinguishing epistemic uncertainty, arising from incomplete knowledge, from aleatoric uncertainty, arising from inherent system variability [8]. Addressing these limitations requires incorporating predictive uncertainty, i.e., both its magnitude and type, into admission decisions under partial system observability. This raises the following question: *How should volatile vehicular capacity be committed under predictive uncertainty?*

This paper introduces *Strategic Mobility and Resource Uncertainty Tracker (SMART), an ETSI-compliant mechanism for uncertainty-aware admission of opportunistic vehicular compute.* Rather than proposing a new orchestration architecture, SMART augments an existing ETSI MEC orchestrator with a probabilistic admission layer that translates predictive uncertainty into risk-aware decisions [1]. By doing so, it enables admission decisions that explicitly account for uncertainty in future resource availability. SMART comprises two functions. The *Knowledge Manager (KM)* learns the predictive distribution of future vehicular capacity using a BNN, separating epistemic from aleatoric uncertainty. The *Decision Manager (DM)* propagates this distribution into a chance-constrained admission program, where a configurable risk level bounds the probability of capacity violations.

Unlike prior uncertainty-aware approaches, which typically target stable cloud environments or model uncertainty only at a coarse level [9], SMART directly incorporates calibrated predictive uncertainty into admission decisions for highly dynamic vehicular resource pools. *To preserve deployability*, it reuses existing ETSI MEC reference points and the Application Mobility Service (AMS) for departure-driven migration, introducing no additional interfaces [1], [10]. Workload arrivals are treated as exogenous, allowing the mechanism to isolate uncertainty in resource availability. Overall, we make

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

three classes of contributions:

Methodological: After a detailed literature review (cf. **§II**), we present SMART, a mechanism for uncertainty-aware pooling of vehicle compute at the network edge (cf. **§III**), along with the associated ETSI MEC context. The section **§III** further details the two manager functions of SMART, the BNN-based forecasting pipeline and the chance-constrained admission program. Together, these components enable SMART to explicitly propagate predictive uncertainty into admission decisions (cf. Fig. 1c). To the best of our knowledge, SMART is the first mechanism that jointly integrates ETSI-compliant vehicular resource pooling, probabilistic resource forecasting with separated epistemic and aleatoric uncertainty, and risk-aware admission control under explicit uncertainty constraints.

Empirical: For empirical evaluation, we use a pseudonymized real-world taxi mobility dataset (T-Drive [11]), containing $10\,223$ taxis and $2\,161$ Base Stations (BSs) covering $1\,254$ km^2 of Beijing, China. We implement a SMART-embedded ETSI-compliant MEC mechanism and evaluate it against a compute-capacity oracle and seven admission baselines, including reactive, mean-based, and uncertainty-aware approaches.

Code release: The above contributions compose SMART code – *an uncertainty-aware admission mechanism for opportunistic vehicular compute* – a Python implementation to be released upon acceptance of the paper. We conclude this paper in section **§V**.

II. The MEC-vehicle Landscape

Vehicular edge capacity varies with mobility and thus cannot be modeled as a stable resource pool, as is often assumed in conventional MEC orchestration. Vehicles may leave the cell, decline to participate, throttle under thermal constraints, reclaim accelerators for local workloads, or offer heterogeneous hardware budgets. Exploiting vehicular pooling is therefore not only a matter of resource discovery but of allocation under uncertainty. This section positions SMART with respect to vehicular far-edge resources, ETSI compliance, and uncertainty-aware allocation.

A. Urban Vehicular Compute Pooling at the Edge

Manufacturers equip vehicles with sufficient computational capacity to satisfy peak processing demands (cf. Fig. 1a). However, autonomous navigation, sensor fusion, and other time-critical functions execute only intermittently, leaving substantial processing resources idle during normal operation [12] (cf. Fig. 1b). Fig. 1a illustrates the CAPEX required to provision fixed infrastructure per BS to sustain peak vehicular demand, with C672 incurring the highest cost and other BSs exhibiting lower investment. Fig. 1b further shows that peak demand is transient and sustained only for a small fraction of the time across most BSs, confirming a persistent mismatch between provisioned capacity and typical demand.

When aggregated across vehicles within a single BS coverage area, idle capacity can match or exceed that of a conventional MEC server [12]. This aggregated capacity does not constitute a stable infrastructure but instead forms a transient resource pool driven by daily mobility patterns (cf. Fig. 1b). Vehicle density varies across diurnal and weekly cycles, with pronounced morning peaks and increased weekday activity [11]. Spatially, vehicles concentrate in commercial districts during working hours, in residential areas in the evening, and at event venues during large gatherings. These patterns induce continuous variation in the available compute envelope within each BS [12]. Such temporal and spatial volatility complicates orchestration: decisions based on instantaneous observations risk acting on outdated system states, while single-point forecasts lack robustness under distributional shifts.

Insight: *The vehicular resource pool enables reuse of distributed computational capacity in user-dense environments, reducing the need for dedicated MEC infrastructure [12]. The main trade-off is runtime orchestration under highly time-varying resource availability.*

B. Literature Limitations

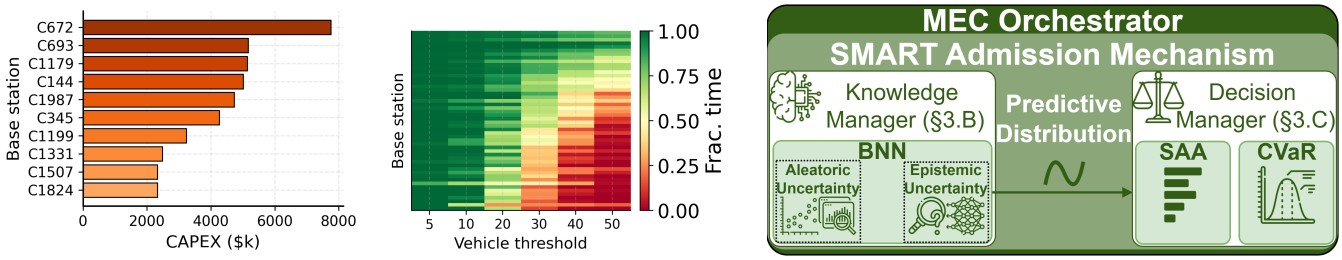
Table I positions SMART against related work. Existing approaches address only subsets of the design space. None jointly integrates ETSI-compliant vehicular resource usage (cf. *Vehicle usage* and *ETSI compliance*), probabilistic forecasting with epistemic–aleatoric uncertainty decomposition (cf. *Resource prediction* and *Uncert. (e/a)*), and uncertainty-aware admission control (cf. *Risk-aware*).

Among the most closely related efforts, Feraudo et al. [5] extend ETSI MEC by incorporating vehicular nodes, including parked far-edge hosts, into the resource pool through participation incentives and transparent migration. Their orchestrator operates on the currently observed pool using a reactive policy, without forecasting future capacity or modeling mobility-induced uncertainty in admission decisions.

The remaining approaches address only parts of this design space. Giannone et al. [13] assume fixed and known resource capacities; Fan et al. [4] optimize vehicular offloading under known resources without ETSI compliance; and AQUATOPE [9] introduces uncertainty awareness but targets stable cloud environments without separating epistemic and aleatoric uncertainty. Likewise, Deep Reinforcement Learning (DRL)-based offloading frameworks generally assume that future resource states are known at decision time, relying on point estimates of system dynamics [6]. Mobility-aware schedulers behave analogously, forecasting individual trajectories or channel values with no uncertainty representation [7]. Both classes of approach treat predictions as exact, failing to separate epistemic uncertainty (from incomplete knowledge) from aleatoric uncertainty (from inherent system variability) [8].

Taken together, these limitations reveal a fundamental missing capability: uncertainty-aware admission control over volatile vehicular resources while preserving ETSI compliance. From a deployment perspective, ETSI compliance is essential, as operators are unlikely to adopt mechanisms that require non-standard MEC reference points.

Positioning: SMART addresses the literature gaps by providing the uncertainty-aware admission logic missing from



(a) CAPEX of BS's peak demand. (b) Time fraction per BS. (c) SMART overview.

Fig. 1: (a) Capital cost of fixed accelerator deployments matching the peak vehicular presence in BSs. Provisioning fixed hardware to meet peak demand is costly and may lead to resource underutilization, as (b) high vehicle counts are sustained for only a small fraction of time. (c) SMART overview with its functional models.

TABLE I: Comparative overview of SMART and related work. ✓ = addressed; — = not addressed; ∼ = partial.

Work	Vehicle usage	ETSI compliance	Resource prediction	Uncert. (e/a)	Risk-aware
Giannone [13]	—	✓	—	—	—
Feraudo [5]	✓	✓	—	—	—
Fan [4]	✓	—	—	—	✓
Nguyen [6]	—	—	—	—	✓
Zhan [7]	—	—	✓	—	✓
AQUATOPE [9]	—	—	✓	∼	✓
SMART (ours)	✓	✓	✓	✓	✓

existing pool-based designs while remaining fully compliance with the ETSI MEC architecture.

The SMART admission mechanism also differs from prior uncertainty-aware decision-making approaches. Chance-constrained and robust formulations (e.g., Gaussian-process MPC, distributionally robust optimization, and conformal prediction) typically target stable cloud workloads, worst-case ambiguity sets, or distribution-free guarantees rather than highly dynamic vehicular resource pools. In contrast, SMART integrates Bayesian capacity forecasting, epistemic–aleatoric uncertainty decomposition, and chance-constrained admission within an ETSI-compatible MEC orchestration path.

Insights: *Deployable vehicular MEC systems must preserve strict ETSI conformance while explicitly handling uncertainty in highly dynamic vehicular resource environments. This, in turn, requires propagating calibrated predictive distributions into admission decisions rather than relying on deterministic resource forecasts.*

III. The SMART Admission Mechanism

SMART extends the ETSI MEC orchestrator with uncertainty-aware admission mechanism for dynamic vehicular computing resources. Specifically, it addresses two challenges in opportunistic vehicular computing: (i) admission decisions should account for future resource availability rather than only the current state; and (ii) predicted resource availability should be treated as uncertain rather than as deterministic point estimates because of mobility-driven fluctuations.

- **Knowledge Manager (KM):** uses a BNN to learn a calibrated predictive distribution of per-cell schedulable capacity, separating epistemic and aleatoric uncertainty (§III-B);
- **Decision Manager (DM):** transforms predicted distributions into risk-aware admission decisions by solving a chance-constrained optimization problem using *SAA* or CVaR approaches (§III-C).

SMART does not introduce any additional ETSI reference points [14]. The KM module utilizes resource-monitoring information already available within the MEC management framework, while the DM module converts admission decisions into standard operations for the MEC application lifecycle. Thus, SMART enhances orchestration while preserving interoperability with existing ETSI MEC deployments.

Next, we present the formalization of the vehicle compute pooling problem, followed by the KM and the BNN used to estimate predictive capacity distributions. Finally, we present the DM and the corresponding risk-aware admission policies based on SAA and CVaR optimization.

A. Problem Formalization

We consider a ETSI MEC system composed of a *set of cells* $\mathcal{K} = \{1, \ldots, K\}$, where each cell corresponds to the coverage area of a BS and is managed by a local orchestrator running SMART. Time is discretized into one-second epochs. Let $\mathcal{V}_k(t)$ denote the *set of vehicles* present in *cell* k at *time* t and $N_k(t) = |\mathcal{V}_k(t)|$ the corresponding *vehicle count*.

We assume that each *vehicle* i allocates a portion of its computing resources to support the execution of external tasks. The available capacity depends on the embedded hardware characteristics, which we denote by the *nominal budget* c_i in TOPS. A fraction, $u_i(t) \in [0, 1]$, of this capacity is consumed onboard for applications such as perception, navigation, and infotainment. The *residual capacity* available for external tasks is therefore given by $c_i\,(1 - u_i(t))$. In a practical deployment, vehicles periodically report their currently available compute capacity to the local ETSI MEC orchestrator.

In addition to the vehicles' capacities, each cell is provided with a static ETSI MEC host with a fixed *compute budget*, denoted c_0. In this case, the orchestrator aggregates the reported capacities of all vehicles currently present in the cell and combines them with the capacity of the static ETSI MEC

host to obtain the *total schedulable capacity* available in cell k at time t, which is equal to:

$$C_k(t) \;=\; c_0 + \sum_{i \in \mathcal{V}_k(t)} c_i \left(1 - u_i(t)\right). \tag{1}$$

This *real capacity* $C_k(t)$ is time-varying due to fluctuations in vehicle workloads and changes in the vehicle population $\mathcal{V}_k(t)$ caused by mobility. As a result, *the amount of computing resources available to the orchestrator cannot be considered constant and must be predicted in advance.*

Thus, the ETSI MEC orchestrator continuously observes $C_k(t)$ and builds a *historical capacity trace*. This trace serves as input to the ***KM***, which predicts the schedulable capacity over a prediction horizon h, noted $C_k(t+h)$. The output of the KM predictor is *a distribution that captures both the expected future schedulable capacity and the uncertainty* resulting from mobility and workload variation.

The KM predictive distribution is then provided to the ***DM***, whose role is to determine which incoming tasks should be admitted for execution within the cell. At each *decision epoch* t, the DM receives a *set of candidate tasks* $\mathcal{J}_W(t)$ to be executed during the *future window* $[t, t+h]$, where $W = h$. Each task $j \in \mathcal{J}_W(t)$ is characterized by a *computing demand* w_j (in TOPS). In this case, the admission decision can be represented by the binary variable $a_j \in 0, 1$ for $j \in \mathcal{J}_W(t)$, where $a_j = 1$ indicates that *task j is admitted* for local execution and $a_j = 0$ indicates that *it is redirected to the cloud*. The total admitted load is equal to:

$$L_k(\mathbf{a}) = \sum_{j \in \mathcal{J}_W(t)} w_j a_j. \tag{2}$$

If the *future real schedulable capacity* $C_k(t+h)$ were known when admission decisions are made, admission control would reduce to the following deterministic optimization problem:

$$\max_{\mathbf{a} \in \{0,1\}^{|\mathcal{J}_W(t)|}} \sum_{j \in \mathcal{J}_W(t)} a_j \quad \text{s.t.} \quad L_k(\mathbf{a}) \le C_k(t+h), \tag{3}$$

In practice, however, $C_k(t+h)$ is unknown because it depends on future vehicle availability. Therefore, since the KM predicts $C_k(t+h)$ as a *conditional probability distribution*, the DM replaces the deterministic constraint in (3) with the following probabilistic chance constraint:

$$\begin{aligned} \max_{\mathbf{a} \in \{0,1\}^{|\mathcal{J}_W(t)|}} \quad & \sum_{j \in \mathcal{J}_W(t)} a_j \\ \text{s.t.} \quad & \Pr\left(L_k(\mathbf{a}) \le C_k(t+h)\right) \ge 1 - \varepsilon. \end{aligned} \tag{4}$$

where $\varepsilon \in [0, 1]$ is the *admissible violation probability*. DM therefore allows the admitted load to exceed the future capacity only with probability at most ε. *The parameter ε therefore controls the trade-off between aggressive local admission and protection against capacity shortfalls.*

We consider two risk-aware admission approaches in the DM: (i) a *Sample Average Approximation (SAA)* of the chance constraint, and (ii) a *Conditional Value-at-Risk (CVaR)* criterion to account for the severity of potential capacity deficits.

B. Knowledge Manager

In order to predict the schedulable capacity of a cell and quantify the associated uncertainty, KM relies on a BNN, which produces probabilistic predictions rather than a single capacity estimate. Since $C_k(t+h)$ denotes the *schedulable capacity* to be predicted h seconds ahead, and given past capacity observation as input x_t, we can derive the *predictive distribution* using the BNN as follow:

$$p(C_k(t+h) \mid x_t, \mathcal{D}) \;=\; \int p(C_k(t+h) \mid x_t, \theta)\; p(\theta \mid \mathcal{D})\, \mathrm{d}\theta. \tag{5}$$

where θ denotes the *network parameters*, $\mathcal{D}$ is the *dataset*, and $p(\theta|\mathcal{D})$ is the *posterior distribution* of the network parameters.

A separate BNN model is trained for each cell using its own capacity history. The objective is to preserve cell-specific mobility patterns and uncertainty characteristics, enabling the predictive distribution to be calibrated to local capacity dynamics rather than averaging across heterogeneous traffic regimes. The *input* $x_t \in \mathbb{R}^{307}$ comprises the most recent capacity observations sampled at 1 Hz (i.e., 300 last seconds) with seven calendar features (time of day and day of week as sin/cos pairs, plus a weekend indicator).

The input data are not used directly in their raw form. They are first normalized using the mean and standard deviation computed on the training set. This step facilitates BNN training by bringing all variables to a comparable scale. The model, therefore, learns and produces its predictions in this normalized space. The predicted future capacities are then transformed back to the cell's original capacity scale, so that the admission controller can make decisions using values expressed in real capacity units.

We consider a BNN model, $f_\theta : \mathbb{R}^{307} \to \mathbb{R}$, composed of two hidden layers of widths 64 and 32, followed by a scalar output. The model is implemented using Bayesian residual blocks to improve learning stability while enabling uncertainty quantification. Batch normalization is used within these blocks, and dropout with a rate of 0.1 provides regularization to mitigate overfitting. Both model uncertainty and observation noise are captured by the BNN using a probabilistic formulation in which the network parameters are assigned prior distributions, and the output is modeled through a Gaussian likelihood. The *posterior distribution* of the model parameters is approximated using Stochastic Variational Inference (SVI). At inference time, the KM draws S *samples* from the approximated posterior distribution and generates S *corresponding predictions* of the future capacity. Finally, the predictive mean capacity is then estimated as

$$\hat{\mu}(x_t) = \frac{1}{S} \sum_{s=1}^{S} f_{\theta^{(s)}}(x_t). \tag{6}$$

The uncertainty associated with the prediction is decomposed into: (i) *epistemic uncertainty* ($\hat{\sigma}^2_{\text{epi}}$), reflecting the model's lack of knowledge, and (ii) *aleatoric uncertainty*

($\hat{\sigma}^2_{\text{ale}}$), capturing the intrinsic variability of the capacity process. Together, they compose the total variance ($\hat{\sigma}^2_{\text{tot}}(x_t)$):

$$\hat{\sigma}^2_{\text{epi}}(x_t) = \frac{1}{S}\sum_{s=1}^{S}\Big(\mu^{(s)}(x_t) - \hat{\mu}(x_t)\Big)^2, \tag{7}$$

$$\hat{\sigma}^2_{\text{ale}} = \frac{1}{S}\sum_{s=1}^{S}\Big(\sigma^{(s)}\Big)^2, \tag{8}$$

$$\hat{\sigma}^2_{\text{tot}}(x_t) = \hat{\sigma}^2_{\text{epi}}(x_t) + \hat{\sigma}^2_{\text{ale}}. \tag{9}$$

where $\sigma^{(s)}$ denotes the *observation-noise standard deviation* associated with the s-th posterior sample.

The *epistemic uncertainty* (7) is reducible with additional data, whereas the *aleatoric uncertainty* (8) represents an irreducible homoscedastic floor. Their relative contributions are summarized by the *epistemic-to-total variance ratio*:

$$\rho(x_t) \;=\; \frac{\hat{\sigma}^2_{\text{epi}}(x_t)}{\hat{\sigma}^2_{\text{tot}}(x_t)} \in [0,1], \tag{10}$$

Finally, once de-standardized, the S predictive samples forming the *forecast schedulable capacity distribution*, $\hat{C}_k^{(s)}(t+h)|_S^1$, and the respective uncertainties are sent to the DM for the risk-aware admission (cf. §III-C).

C. Decision Manager

The objective of the DM is to determine the set of tasks that can be admitted while controlling the risk of future capacity shortages. Starting from the chance-constrained formulation introduced in (4), we investigate two scenario-based approaches. The first approach uses SAA to control the frequency of capacity violations, while the second incorporates CVaR to account for the severity of such violations.

*1) **SAA-Based Admission**:* Solving the chance-constrained problem (4) requires access to the forecast schedulable capacity distribution $\hat{C}_k(t+h)$. Since the KM provides only a finite set of predictive scenarios rather than a closed-form probability density, the DM approximates the chance-constraint using the SAA framework. The probability constraint in (4) is thus replaced by the empirical counterpart as follow:

$$\frac{1}{S}\sum_{s=1}^{S}\mathbf{1}\{L_k(\mathbf{a}) \le \hat{C}_k^{(s)}(t+h)\} \ge 1-\varepsilon, \tag{11}$$

where, $\varepsilon \in (0,1]$ denotes the *admissible violation probability*.

Since the admitted load $L_k(\mathbf{a}) = \sum_j w_j a_j$ does not depend on the scenario index s, the empirical violation probability remains below ε if and only if the admitted load does not exceed the empirical lower-tail ε-quantile of the scenarios.

Let consider $\hat{C}_k^{(1)} \le \ldots \le \hat{C}_k^{(S)}$ denote the ordered capacity scenarios. In this case, the corresponding *SAA capacity budget* is defined by

$$\widehat{\mathcal{C}}_k^{\text{SAA}}(\varepsilon) \;=\; \hat{C}_k^{(\lceil \varepsilon S \rceil)}(t+h), \tag{12}$$

This value corresponds to the empirical lower-tail quantile, ensuring that at most an ε-fraction of scenarios violate the capacity constraint. Therefore, the *chance-constrained admission problem* reduces to

$$\text{(P1-SAA-KP)} \qquad \max_{\mathbf{a}\in\{0,1\}^{|\mathcal{J}_W(t)|}} \sum_j a_j \quad \text{s.t.} \quad L_k(\mathbf{a}) \le \widehat{\mathcal{C}}_k^{\text{SAA}}(\varepsilon). \tag{13}$$

Insight: *A smaller value of ε selects a lower capacity budget, resulting in a more conservative admission policy.*

*2) **CVaR-Based Admission**:* While the SAA formulation controls the frequency of capacity shortages, it does not account for their severity. In practice, the operational cost of an over-admitted event typically increases with the magnitude of the resulting capacity deficit. To address this limitation, a CVaR-based admission approach is also considered.

If we define the over-admitted loss as $\Lambda(\mathbf{a}, C) = L_k(\mathbf{a}) - C$, the corresponding admission problem is formulated as

$$\text{(P1-CVaR)} \qquad \max_{\mathbf{a}\in\{0,1\}^{|\mathcal{J}_W(t)|}} \sum_j a_j \quad \text{s.t. } \text{CVaR}_\alpha\big(\Lambda(\mathbf{a}, C_k(t+h))\big) \le 0, \tag{14}$$

where CVaR_α is the *expected loss* over the worst α-fraction of outcomes.

Because the admitted load $L_k(\mathbf{a})$ is deterministic with respect to the capacity scenarios, the largest losses occur in the lowest-capacity realizations. Thus, the CVaR of the loss can be computed directly from the worst α-tail of Λ, which is the lowest α-tail of the capacity scenarios. Let $S_\alpha^{\text{low}}(k)$ denote the set containing the $\lceil \alpha S \rceil$ smallest capacity samples for cell k. The CVaR constraint is equivalent to

$$L_k(\mathbf{a}) \le \widehat{\mathcal{C}}_k^{\text{CVaR}}(\alpha), \tag{15}$$

where the corresponding *CVaR capacity budget* is

$$\widehat{\mathcal{C}}_k^{\text{CVaR}}(\alpha) \;=\; \frac{1}{\lceil \alpha S \rceil}\sum_{s\in S_\alpha^{\text{low}}(k)} \hat{C}_k^{(s)}(t+h), \tag{16}$$

Hence, the *chance-constrained admission problem* becomes

$$\text{(P1-CVaR-KP)} \qquad \max_{\mathbf{a}\in\{0,1\}^{|\mathcal{J}_W(t)|}} \sum_{j\in\mathcal{J}_W(t)} a_j \quad \text{s.t.} \quad L_k(\mathbf{a}) \le \widehat{\mathcal{C}}_k^{\text{CVaR}}(\alpha). \tag{17}$$

Insight: *A smaller value of α places greater emphasis on the lowest capacities realizations, resulting in a conservative admission policy.*

*3) **Problem resolution**:* The two formulations address different aspects of predictive uncertainty. While the SAA budget in (12) depends on a single lower-tail order statistic, the CVaR budget in (16) averages the entire lower-capacity tail. The two budgets diverge when the predictive distribution is highly dispersed or asymmetric. This is particularly *the case when epistemic uncertainty contributes significantly to the total uncertainty measured by* (10).

TABLE II: Notation used throughout the paper.

Symbol	Meaning
System model: Base stations, vehicles, tasks, time	
$\mathcal{D}$; $\mathcal{K}$; k	T-Drive dataset; set of BS; its index
x_t	input $\mathcal{D}$: 300 capacity lags and 7 calendar features
W ; h	decision window; prediction horizon: $W{=}h{=}1$s
$N_k(t)$	vehicle count: $N_k(t) = \lvert\mathcal{V}_k(t)\rvert$
$\mathcal{J}_W(t)$; j	candidate tasks for window $[t, t{+}h]$; task index
w_j	compute demand of task j (TOPS)
a_j ; $\mathbf{a}$	admission variable of task j; admission vector
$L_k(\mathbf{a})$	admitted load: $L_k(\mathbf{a}) = \sum_j w_j a_j$
Schedulable capacity – Eq. (1)	
c_i ; $u_i(t)$	nominal budget; onboard utilization of vehicle i
c_0	static operator-owned host budget (TOPS)
$C_k(t)$	real schedulable capacity of BS_k at time t (TOPS)
$C_k(t{+}h)$	future real schedulable capacity of BS_k at time $t{+}h$
Knowledge Manager (BNN)	
$\mu^{(s)}(x_t)$; $\hat{\mu}(x_t)$	per-sample (s) mean; predictive mean (6)
$\hat{C}_k^{(s)}(t{+}h)$	s-th future estimated or predicted schedulable capacity
Uncertainty decomposition	
$\hat{\sigma}^2_{\mathrm{epi}}$; $\hat{\sigma}^2_{\mathrm{ale}}$	epistemic (7) and aleatoric (8) uncertainties
$\hat{\sigma}^2_{\mathrm{tot}}$	total variance: $\hat{\sigma}^2_{\mathrm{tot}} = \hat{\sigma}^2_{\mathrm{epi}} + \hat{\sigma}^2_{\mathrm{ale}}$
$\rho(x_t)$	epistemic-to-total variance ratio (10): $\rho(x_t) = \hat{\sigma}^2_{\mathrm{epi}}/\hat{\sigma}^2_{\mathrm{tot}}$
Decision Manager (risk-aware admission)	
ε	SAA admissible violation probability
$\widehat{\mathcal{C}}_k^{\mathrm{SAA}}(\varepsilon)$	SAA capacity budget of the lower-tail ε-quantile (12)
α	CVaR tail probability level
$\widehat{\mathcal{C}}_k^{\mathrm{CVaR}}(\alpha)$	CVaR capacity budget of the worst α-tail (16)
$\varepsilon^\star$, $\alpha^\star$	per-BS operating risk levels (§IV-C)

Once the risk-adjusted capacity budget is determined, both (13) and (17) admission formulations can be expressed as *0/1 Knapsack Problem* over the set of candidate tasks in $|\mathcal{J}_W(t)|$. The *capacity budget* is computed from the S *predictive samples*, and the resulting knapsack problem can be solved using any standard knapsack solver; in our implementation, we use a *branch-and-bound mixed-integer solver (CBC)*.

This work does not propose an online mechanism that adapts the *risk-aware admission parameters*, ε and α, during operation. Instead, their operating regimes are characterized empirically. For each *cell* k, ε and α are swept over a predefined grid to solve (13) and (17) on the test windows. The resulting admission rate and capacity overshoot are then measured to evaluate the impact of the risk parameters.

Insight: *The risk-aware admission parameters ε and α govern the trade-off between task admission and the risk of capacity over-commitment. They therefore require empirical calibration (cf. §IV) to identify operating points that balance resource utilization and over-commitment risk.*

IV. Evaluation

We evaluate SMART on vehicular mobility traces. §IV-A describes the dataset, the pipeline that turns raw GPS trajectories into the capacity trace, and the experimental configuration. §IV-B then assesses the KM's forecasting accuracy and calibration. Finally, §IV-C–§IV-D evaluate the DM's admission against a compute-capacity oracle and seven baselines.

A. Evaluation Setup

1) Dataset and Capacity Generation: We use the T-Drive dataset [11], which contains one week of GPS trajectories collected from a Beijing taxi fleet, as a realistic proxy for connected vehicles contributing as computing resources at the edge. The dataset comprises 10 223 vehicles and 2 161 BSs covering approximately 1 254 km^2 in Beijing. Because GPS observations are sparse and irregular, with position reports typically available every few minutes, the dataset provides only intermittent observations of vehicle locations.

We transform the raw trajectories into per-base-station vehicle counts. We start by restricting the GPS records only to the Beijing bounding box (latitude $[39.76, 40.09]$, longitude $[116.17, 116.57]$). Thereafter, we map each GPS sample to its nearest BS using a Voronoi partition of the BS locations. Finally, since the original traces are sparsely sampled, we apply a zero-order hold (ZOH) interpolation to reconstruct a one-second vehicle association process. Under this assumption, a vehicle remains attached to its last observed BS until it receives a new GPS update.

At each second, the reconstructed associations define the set of vehicles present in cell k, denoted by $\mathcal{V}_k(t)$, and the corresponding vehicle count $N_k(t) = |\mathcal{V}_k(t)|$. These quantities are then used to compute the schedulable capacity $C_k(t)$ introduced in Section III-A. In addition, each vehicle contributes to a computing capacity of 275 TOPS, comparable to a modern automotive accelerator. To isolate the impact of vehicular resource dynamics, we assume the MEC host contributes no compute resources ($c_0 = 0$), such that the total available capacity is determined solely by the vehicles.

Finally, in order to evaluate SMART under different mobility conditions, we select 45 BSs that exhibit the greatest temporal variability in vehicle availability. Variability is quantified using the standard deviation of the vehicle count process $N_k(t)$. The selected BSs are sorted according to this metric and partitioned into three groups of 15 BSs: *Stable*, *Moderate*, and *Volatile*. The corresponding thresholds are defined by the 33rd and 67th percentiles of the vehicle-count standard deviation (4.01 and 10.66 vehicles, respectively) and remain fixed throughout the evaluation: 2.68 to 3.81 vehicles in *Stable* (median 3.40), from 4.11 to 4.91 in *Moderate* (median 4.57), and from 22.16 to 156.48 in *Volatile* (median 44.63).

2) Experimental Configuration: We train a separate KM for each BS using its own capacity history. The capacity traces are split chronologically: the first four days for training, the fifth for validation, and the sixth for testing. The validation day is used for temperature calibration and early stopping, while all reported results are obtained from the held-out test day. At each decision epoch, the KM draws $S = 100$ posterior samples of $C_k(t{+}h)$; §IV-B shows that admission performance is insensitive to this sample count. Both the decision window and the prediction horizon are set to 1 s ($W = h = 1$ s).

The DM is evaluated under workloads that keep each BS close to saturation. Tasks arrive at a constant rate of 100 tasks/s throughout the test day. The compute demands are scaled so that the average offered load matches the average schedulable

capacity of each BS. This normalizes the offered-load-to-capacity ratio across BSs while preserving differences in capacity dynamics. Each task represents a generic Vehicular Edge Computing (VEC) inference job characterized by its input size, compute demand w_j, and deadline.

At each decision epoch, the DM solves one admission problem. As shown in §III-C, both the SAA (13) and CVaR (17) formulations reduce to a 0/1 knapsack problem, and are solved using the CBC branch-and-bound solver. We also considered an oracle policy that uses the true future capacity $C_k(t+h)$, thereby providing the maximum achievable admission rate under perfect knowledge. The SAA policy uses the ε-quantile of the predictive scenarios, whereas the CVaR policy uses the average of the worst α fraction of capacity realizations.

In this sense, the parameters ε and α are independently swept over the set $\{0.1, 0.2, \ldots, 0.9, 0.99\}$ to characterize the impact of risk sensitivity. Across the 45 BSs, this produces $45 \times (10 + 10 + 1) = 945$ experimental runs. For each run, we measure the admission rate, admission loss with respect to the oracle, capacity overshoot, cloud offloading, and realized capacity utilization. The parameter sweep allows us to identify operating points that balance admission performance and overcommitment risk and to relate them to the uncertainty characteristics of each BS (§IV-C).

B. KM: Predictive Performance

We evaluate the KM along three complementary dimensions. First, accuracy measures how closely the predicted capacity matches the realized capacity. Second, uncertainty calibration to evaluate whether predictive intervals achieve the true capacity as often as they claim. Finally, the sensitivity to the number of posterior samples used during inference.

We compare the proposed BNN with three forecasting baselines: a deterministic Long Short-Term Memory (LSTM) (i.e., producing a single value with no uncertainty), Monte Carlo (MC) Dropout (i.e., approximating Bayesian inference by sampling dropout masks) [15], and a deep ensemble (i.e., reading uncertainty from the spread of independently trained networks) [16]. All models predict the one-second-ahead schedulable capacity $C_k(t+h)$ on the held-out test day.

1) Accuracy: Fig. 2(a)-(b) report point accuracy. The deep ensemble achieves the lowest median Mean Absolute Error (MAE) (20 TOPS), followed by the deterministic LSTM (21 TOPS), whereas MC Dropout and the proposed BNN obtain comparable errors (35 and 36 TOPS, respectively). Point accuracy alone, therefore, does not distinguish the BNN; its advantage lies in predictive calibration, which is discussed next. Forecasting errors are concentrated in the *Volatile* BS group, where median MAE increases from tens to hundreds of TOPS, confirming that the proposed grouping captures forecasting difficulty. The LSTM is particularly sensitive to abrupt changes, with the worst case (BS C672) reaching $19\,725$ TOPS because it cannot express predictive uncertainty.

Figure 2b reports the corresponding relative error. Although *Volatile* BSs exhibit much larger absolute errors, they also provide substantially larger compute capacity, keeping the BNN's mean absolute percentage error nearly constant across all groups (1.1%, 1.2%, and 1.0% for *Stable*, *Moderate*, and *Volatile*, respectively). Since admission decisions depend on relative rather than absolute capacity error, this explains why the *Volatile* group ultimately achieves admission performance closest to the oracle (§IV-C). *As a conclusion, point accuracy alone does not distinguish the probabilistic models; however, the consistently low relative error across regimes ensures that forecasting errors have limited impact on admission decisions, which depend on capacity ratios rather than absolute values.*

2) Uncertainty Calibration: While accuracy measures how close the predicted capacity is to the true capacity, the calibration asks a different question: *do predictive intervals contain the real capacity as often as they claim?*

Reliability. Figure 2c reports empirical versus nominal (real) coverage, where perfect calibration lies on the diagonal. The proposed BNN closely follows this perfect case across the full range, achieving 95.7% empirical coverage at the nominal 95% level (IQR 95.3–96.3%). In contrast, MC Dropout (99.0%) and the deep ensemble (98.1%) consistently over-cover, indicating overly conservative predictive intervals.

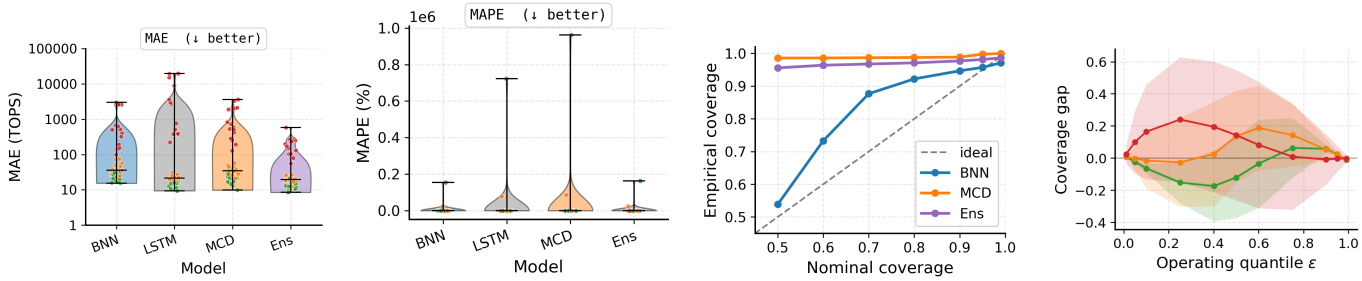
This behavior is reflected in the calibration error at the 95% level, defined as the absolute difference between empirical and nominal coverage. The BNN achieves the lowest median error (0.79 percentage points), compared with 3.11 for the ensemble and 4.03 for MC Dropout. *The BNN provides the best balance between predictive sharpness and calibration.*

Tail calibration. Nominal 95% coverage alone is insufficient because SMART bases admission decisions on lower predictive quantiles (12). Fig. 2d reports the coverage gap (empirical minus nominal) across the full quantile range. A gap near zero indicates that the realized admission risk matches the target risk. *Stable* and *Volatile* BSs exhibit opposite behaviors. *Stable* BSs are over-confident, with the largest negative gap (-0.17 at $\varepsilon = 0.40$), whereas *Volatile* BSs are under-confident, with the largest positive gap ($+0.24$ at $\varepsilon = 0.25$). *These results indicates that a single calibration temperature cannot calibrate all volatility regimes simultaneously.*

TABLE III: Expected Calibration Error (ECE) between empirical and nominal coverage, with sharpness (MPIW_{95} in TOPS).

	ECE (pp)				Sharpness
Model	Stable	Moderate	Volatile	All	MPIW_{95}
BNN	11.8	14.1	12.4	13.2	287
MC Dropout	19.6	19.0	21.2	19.8	1006
Ensemble	19.0	17.9	13.3	18.0	313

Overall calibration. Table III presents the ECE, calculated as the mean absolute difference between empirical and nominal coverage for the 50%, 80%, 90%, and 95% prediction intervals, along with the median width of the 95% prediction interval (MPIW_{95}). The BNN achieves the lowest ECE overall (13.2 pp, versus 18.0 for the ensemble and 19.8 for MC Dropout) while maintaining the narrowest prediction intervals (287 TOPS, versus 313 and 1006 TOPS). It consistently outperforms the competing probabilistic models across all volatility groups. Residual calibration error is concentrated



(a) Mean absolute error.

(b) Mean absolute percentage error.

(c) Reliability vs. model.

(d) Coverage gap vs. group.

Fig. 2: Predictive accuracy (a,b) and calibration (c,d) by volatility group (● Stable ● Moderate ● Volatile).

at intermediate coverage levels because temperature scaling is optimized for the 95% interval. Nevertheless, the BNN's advantage is statistically significant across the 45 BSs: a Friedman test rejects equal calibration error among the probabilistic models ($\chi^2 = 42.2$, $p < 10^{-9}$), while Holm-corrected Wilcoxon signed-rank tests confirm lower calibration error than both MC Dropout ($p < 2 \times 10^{-4}$) and the deep ensemble ($p < 5 \times 10^{-3}$). *The BNN provides the most reliable predictive quantiles while avoiding the excessive conservatism exhibited by alternative probabilistic predictors.*

3) Sample-count sensitivity: Finally, we evaluate the sensitivity of the KM to the number of Monte Carlo samples. We vary the sample size S in (5) over $S \in \{25, \ldots, 1000\}$ across all 45 BSs. Point accuracy is insensitive to S, while calibration improves with larger sample sizes. In particular, on the *Volatile* group, the spread of the 95% predictive coverage (Prediction Interval Coverage Probability $(\text{PICP})_{95}$) across BSs and MC trials decreases from 14.3 pp at ($S = 25$) to 1.9 pp at ($S = 1000$). However, admission decisions stabilize at much smaller sample sizes. At $S = 100$, re-running the SAA and CVaR sweeps changes the median deviation from the oracle by at most 0.02 percentage points compared to $S = 1000$. *As a conclusion, stable admission decisions can be obtained with only $S = 100$ posterior samples, providing a favorable trade-off between decision robustness and inference latency.*

Findings: *The BNN combines competitive point accuracy with substantially better calibration than alternative probabilistic predictors, while requiring only a modest number of posterior samples for stable inference. These properties make it well suited to drive SMART's uncertainty-aware admission policy, whose performance depends on reliable predictive quantiles rather than point forecasts alone.*

C. DM: Admission Performance

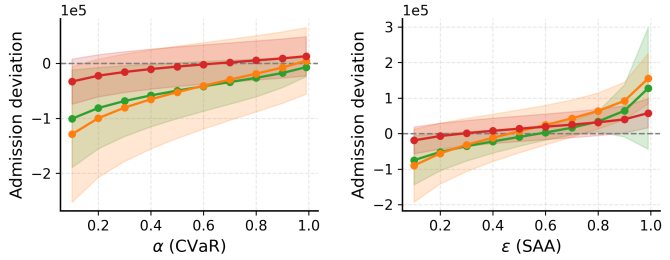
Unless otherwise stated, all reported quantities are expressed as deviations from the oracle admission. A value of zero, therefore, indicates that the admission policy matches the oracle, which has perfect knowledge of the future capacity. The risk parameters ε (SAA) and α (CVaR) control the conservativeness of the admission decision: increasing either parameter relaxes the admission constraint and allows more tasks to be admitted.

1) Risk–Admission Tradeoff: Figure 3 reports the admission deviation only with respect to the oracle for the three volatility groups. As we can observe, the behavior of SAA is more aggressive. Small values of ε lead to conservative admission, while larger values eventually cause over-admission. At $\varepsilon = 0.99$, the *Stable* and *Moderate* groups exceed the oracle admission, indicating that optimistic capacity estimates can accept more load than the realized capacity can support. On the other hand, the *Volatile* group stays nearest the oracle across the whole ε range, since, against its large capacity, the same prediction error affects a smaller fraction of its admission.

Under CVaR, all groups under-admit at small values of α and gradually approach the oracle as the risk level increases. The *Volatile* group remains closest to the oracle over the entire range and slightly exceeds it at large α values. *Stable* and *Moderate* cells approach the oracle more conservatively.

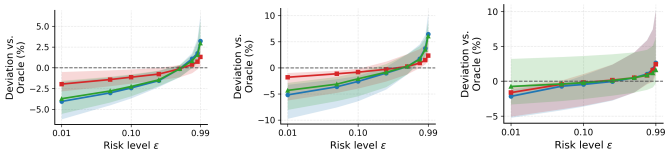
Finally, we observe that, for both approaches, the optimal risk parameter configuration ($\varepsilon^\star$ for SAA and $\alpha^\star$ for CVaR) depends on the volatility regime of each cell. We can see that the *Volatile* group favors lower risk parameters and a more conservative approach. On the other hand, *Stable* and *Moderate* cells require larger risk parameters to recover the oracle admission. This observation suggests that the admission risk needs to be adjusted to each cell's volatility. *As a conclusion, both SAA and CVaR achieve admission levels close to the oracle while offering different risk profiles. SAA provides more aggressive admission, whereas CVaR offers a safer and more conservative behavior. The optimal risk level depends on the uncertainty regime of the cell.*

2) Impact of Uncertainty Decomposition: Figure 4 evaluates the contribution of epistemic and aleatoric uncertainty to the admission decision. We considered three variants. The first variant, noted **TOTAL**, uses the full predictive draws. The seconde variant, noted $\textbf{EPI}_{\textbf{only}}$, keeps the posterior-mean samples $\mu^{(s)}(x_t)$ alone. Finally, the last variant, noted $\textbf{ALE}_{\textbf{only}}$, uses the posterior mean plus fresh draws from $\mathcal{N}(0, \hat{\sigma}_{\text{ale}})$. The results show that the two uncertainty components do not contribute equally to the admission decision. In *Stable* and *Moderate* cells, the aleatoric component dominates the



(a) Admission vs. α (CVaR). (b) Admission vs. ε (SAA).

Fig. 3: Admission deviation against the oracle, by group. ● Stable ● Moderate ● Volatile.



(a) Stable. (b) Moderate. (c) Volatile.

Fig. 4: Variance decomposition ablation. ● Total ($\sigma^2_{\text{epi}}+\sigma^2_{\text{ale}}$) ■ Epistemic only ($\sigma^2_{\text{epi}}$) ▲ Aleatoric only ($\sigma^2_{\text{ale}}$).

lower predictive tail and largely determines the effective capacity budget. In contrast, the epistemic component becomes increasingly important in the *Volatile* group. *In conclusion, removing either component changes admission behavior, and the direction of that change depends on the regime rather than simply on shrinking or expanding a uniform safety margin.*

Findings. *Epistemic and aleatoric uncertainty influence admission decisions differently across mobility regimes. While aleatoric uncertainty primarily determines the effective admission budget, the contribution of epistemic uncertainty increases in highly volatile cells, making their explicit separation beneficial for risk-aware admission.*

D. Comparison Baselines

Beyond the oracle comparison, we evaluate SMART against seven baselines spanning three classes, all evaluated on the same 45 BSs and workload (Table IV). These baselines allow us to isolate the benefits of forecasting (Class A vs. Class B), of distributional information (Class B vs. Class C), and of the proposed probabilistic forecasters (B1 vs. B2):

- **Class A -** Reactive policies that do not forecast capacity: A1 admits a task if its demand fits within the last observed capacity (adapted from [5]), while A2a and A2b greedily prioritize tasks by decreasing (First-Fit Decreasing) and increasing demand, respectively.
- **Class B -** Policies using point forecasts of mean future capacity: B1 relies on the predictive mean of the proposed BNN, whereas B2 uses a deterministic LSTM.
- **Class C -** Uncertainty-aware policies: C1 applies split-conformal prediction using the per-BS residual quantile q_ε, while C2 adopts robust optimization with a box uncertainty set $(\mu - \Gamma\sigma)$.

Table IV compares *Admission* and *Capacity-Violation* rates across the three BS volatility groups and baselines. A *Capacity-Violation* occurs when the admitted load exceeds the real capacity during a one-second decision window.

For SMART, each BS operates at its calibrated risk level $\varepsilon^\star$ for SAA or $\alpha^\star$ for CVaR), selected such that its admission rate does not exceed the oracle. At these operating points, SMART-SAA and SMART-CVaR admit 95.1% and 95.0% of tasks, respectively (cf. *All* column), effectively matching the oracle (95.1%). In contrast, the remaining baselines admit only 75% to 77% of tasks, except the greedy policies A2a (3.7%) and A2b (26.3%).

Across all policies, *Volatile* BSs achieve the highest admission rates, including under the oracle (97.9% versus 93.5% for *Stable* BSs), indicating that admission performance is primarily driven by available capacity dynamics rather than forecasting difficulty. Neither reactive policies nor methods that rely solely on point forecasts or generic uncertainty estimates match SMART's performance, indicating that the gain comes from translating predictive uncertainty into a risk-aware admission budget.

The SMART's higher admission rate comes at the cost of more frequent *Capacity Violations*. Overall (cf. *All* column), SMART-SAA and SMART-CVaR violate capacity in 6.1% and 4.2% of admitted decisions, respectively, whereas all baselines remain below 0.5% by admitting substantially fewer tasks. CVaR consistently provides the safer operating point, particularly for *Volatile* BSs (i.e., 2.5% of violated decisions versus 8.2% for SAA), because averaging the lower α-tail produces a more conservative capacity estimate than the single SAA quantile.

V. Conclusions and Future Work

We presented SMART, an ETSI-compliant mechanism for uncertainty-aware admission of opportunistic vehicular compute. SMART is composed of two modules: a Knowledge Manager that learns a calibrated predictive distribution of future per-BS compute capacity using a BNN, and a Decision Manager that performs risk-aware admission via chance-constrained optimization using SAA and CVaR approximations.

Using a real-world taxi mobility dataset, we showed that the proposed BNN provides the best-calibrated forecasts at the nominal 95% coverage level. Under the compute-only admission model, SMART admits approximately 95% of tasks (i.e., 95.1% with SAA and 95.0% with CVaR), matching the Oracle and outperforming seven reactive, point-forecast, and uncertainty-aware baselines by approximately 23% in relative admission rate

This improvement is achieved at capacity-violation rates of 6.1% (SAA) and 4.2% (CVaR). All decisions are executed within the one-second control interval with lightweight per-BS control-plane overhead. We analyze the operating conditions under which SMART remains effective. Sensitivity

TABLE IV: Mean $\pm$ 95% confidence interval of admission and capacity-violation rates across the 45 cells.

	Admission (%)				Cap. violation (%)			
Policy	Stable	Moderate	Volatile	All	Stable	Moderate	Volatile	All
A1 Reactive (Feraudo)	75.5 ± 2.3	75.7 ± 1.3	78.6 ± 0.2	76.6 ± 0.9	0.1 ± 0.1	0.0 ± 0.1	0.0 ± 0.0	0.0 ± 0.0
A2a Greedy FFD	3.7 ± 0.0	3.7 ± 0.1	3.7 ± 0.0	3.7 ± 0.0	0.0 ± 0.1	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
A2b Greedy SFD	26.1 ± 0.3	26.1 ± 0.2	26.5 ± 0.0	26.3 ± 0.1	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
B1 Mean only (BNN)	75.4 ± 2.4	75.9 ± 1.7	79.1 ± 0.8	76.8 ± 1.0	0.2 ± 0.3	0.1 ± 0.1	0.0 ± 0.0	0.1 ± 0.1
B2 Mean only (LSTM)	75.6 ± 2.2	75.7 ± 1.3	74.1 ± 3.6	75.1 ± 1.4	0.9 ± 1.8	0.4 ± 0.6	0.0 ± 0.0	0.4 ± 0.6
C1 Conformal	75.4 ± 2.4	75.9 ± 1.7	79.0 ± 0.8	76.8 ± 1.0	0.2 ± 0.2	0.1 ± 0.1	0.0 ± 0.0	0.1 ± 0.1
C2 Robust	74.9 ± 2.5	75.2 ± 1.8	78.7 ± 0.7	76.3 ± 1.1	0.1 ± 0.1	0.1 ± 0.1	0.0 ± 0.0	0.1 ± 0.0
SMART-SAA ($\varepsilon^\star$)	93.5 ± 3.4	93.8 ± 1.7	97.9 ± 0.4	95.1 ± 1.3	4.9 ± 1.1	5.1 ± 2.5	8.2 ± 4.7	6.1 ± 1.7
SMART-CVaR ($\alpha^\star$)	93.4 ± 3.4	93.7 ± 1.9	97.7 ± 0.4	95.0 ± 1.3	5.1 ± 3.0	4.9 ± 1.5	2.5 ± 1.2	4.2 ± 1.1
oracle (capacity)	93.5 ± 3.4	94.0 ± 1.7	97.9 ± 0.4	95.1 ± 1.3	—	—	—	—

to variability in base-station compute availability is significant. Consequently, under abrupt capacity drops, aggressive admission reduces available headroom, suggesting the need for complementary mechanisms such as reserved capacity, proactive migration, or fallback to static MEC resources. Overall, the homogeneous-capacity assumption used in this work isolates mobility-driven uncertainty and should be interpreted as an evaluation setting rather than a deployment guarantee. Future work will focus on online risk adaptation, multi-step forecasting, and extending the model to include communication and queueing effects beyond compute-only resources.

ACKNOWLEDGMENT

This work was funded by the French government under the France 2030 ANR program "PEPR Networks of the Future" (ref. ANR-22-PEFT-0003)

REFERENCES

[1] ETSI, "Multi-access edge computing (MEC); framework and reference architecture," ETSI GS MEC 003 V4.1.1, 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/mec/001_099/003/04.01.01_60/gs_mec003v040101p.pdf

[2] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[3] S. Lu and W. Shi, "Vehicle computing: Vision and challenges," *Journal of Information and Intelligence*, vol. 1, no. 1, pp. 23–35, 2023.

[4] W. Fan, J. Liu, M. Hua, F. Wu, and Y. Liu, "Joint task offloading and resource allocation for multi-access edge computing assisted by parked and moving vehicles," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5314–5330, 2022.

[5] A. Feraudo, A. Calvio, A. Bujari, and P. Bellavista, "A novel design for advanced 5G deployment environments with virtualized resources at vehicular and MEC nodes," in *2023 IEEE Vehicular Networking Conference (VNC)*, 2023, pp. 97–103.

[6] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Cooperative task offloading and block mining in blockchain-based edge computing with multi-agent deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 22, no. 4, pp. 2021–2037, 2023.

[7] W. Zhan, C. Luo, G. Min, C. Wang, Q. Zhu, and H. Duan, "Mobility-aware multi-user offloading optimization for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3341–3356, 2020.

[8] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, vol. 30, 2017, pp. 5574–5584, https://proceedings.neurips.cc/paper_files/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html.

[9] Z. Zhou, Y. Zhang, and C. Delimitrou, "AQUATOPE: QoS-and-uncertainty-aware resource management for multi-stage serverless workflows," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*. Association for Computing Machinery, 2023, pp. 1–14.

[10] ETSI, "Multi-access edge computing (MEC); application mobility service API," ETSI GS MEC 021 V2.2.1, 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/021/

[11] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-Drive: Driving directions based on taxi trajectories," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010, pp. 99–108.

[12] R. Patanè, N. Achir, A. Araldo, and L. Boukhatem, "Vehicular cloud computing: A cost-effective alternative to edge computing in 5G networks," *Computer Networks*, vol. 268, p. 111365, 2025.

[13] F. Giannone, P. A. Frangoudis, A. Ksentini, and L. Valcarenghi, "Orchestrating heterogeneous MEC-based applications for connected vehicles," *Computer Networks*, vol. 180, p. 107402, 2020.

[14] ETSI, "Multi-access edge computing (mec); mec management; part 2: Application lifecycle, rules and requirements management," ETSI, Tech. Rep. ETSI GS MEC 010-2 V4.1.1, 2025.

[15] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, vol. 48, 2016, pp. 1050–1059, https://arxiv.org/abs/1506.02142.

[16] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, https://arxiv.org/abs/1612.01474.