Enabling Delayed-Full Charging Through Transformer-Based Real-Time-to-Departure Modeling for EV Battery Longevity

Yonggeon Lee[1], Jibin Hwang[1], Alfred Malengo Kondoro[1], Juhyun Song[2*], Youngtae Noh[1*]

[1]Hanyang University, Seoul, Republic of Korea
[2]KENTECH, Naju-si, Republic of Korea
[1]{yonggeonlee, hjb7165, alfr3do, youngtaenoh}@hanyang.ac.kr
[2]jsong@kentech.ac.kr

Abstract

Electric vehicles (EVs) are key to sustainable mobility, yet their lithium-ion batteries (LIBs) degrade more rapidly under prolonged high states of charge (SOC). This can be mitigated by delaying full charging *DFC* until just before departure, which requires accurate prediction of user departure times. In this work, we propose Transformer-based real-time-to-event (TTE) model for accurate EV departure prediction. Our approach represents each day as a TTE sequence by discretizing time into grid-based tokens. Unlike previous methods primarily dependent on temporal dependency from historical patterns, our method leverages streaming contextual information to predict departures. Evaluation on a real-world study involving 93 users and passive smartphone data demonstrates that our method effectively captures irregular departure patterns within individual routines, outperforming baseline models. These results highlight the potential for practical deployment of the *DFC* algorithm and its contribution to sustainable transportation systems.

Code — https://github.com/LYGLeo/3TD-AISI-26
extended version — http://arxiv.org/abs/2512.07723

Introduction

As global efforts toward net-zero emissions intensify, electric vehicles (EVs) have emerged as a key component of sustainable transportation. However, large-scale adoption is hindered by lithium-ion battery (LIB) degradation, which reduces driving range, raises replacement costs, and leads to environmental waste. A primary cause of degradation is extended exposure to high state of charge (SOC), often resulting from early, unnecessary full charging before use. To address this, the Delayed-Full Charging (*DFC*) strategy was proposed in (Lee et al. 2024). *DFC* postpones charging until shortly before the expected EV departure time (*e.g.*, 30 minutes), then performs fast charging to reach full SOC, thereby minimizing the dwell time at 100% SOC ($t_{100\%}$). This approach promotes sustainability by extending battery life and aligns with emerging global regulations, such as the European Union's 2027 mandate on battery health.

*Corresponding authors.

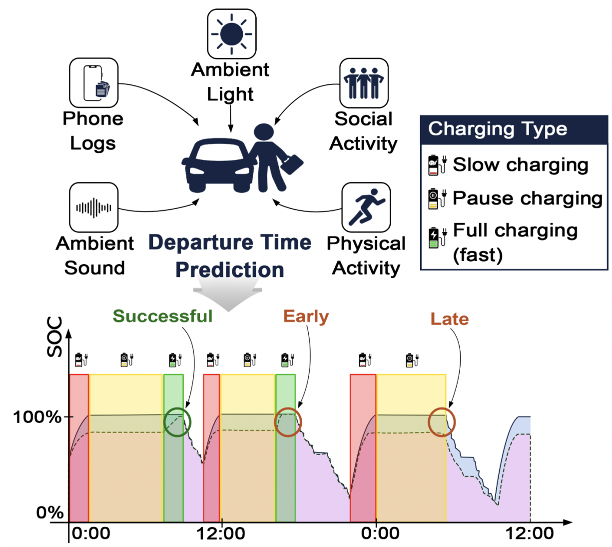

Figure 1: Virtual operation of *DFC*. Early predictions cause prolonged high SOC, reducing battery health benefits. Late predictions risk insufficient charge and range anxiety. Accurate prediction of departure at least 30 minutes in advance ensures full charging for effective *DFC* operation, extending battery life while maintaining user confidence.

Indeed, the practical efficacy of the *DFC* algorithm relies on accurate prediction of EV departure time. Inaccurate predictions introduce two critical risks: early charging reduces *DFC* 's benefits by prolonging high SOC exposure, whereas late charging can leave insufficient SOC at departure, causing user discomfort (Figure 1). Consequently, departure time emerges as a key control parameter for charging scheduling within the *DFC* framework. Existing methods primarily rely on historical statistics of usage patterns, offering limited adaptability to dynamic user contexts (Frendo, Gaertner, and Stuckenschmidt 2021; Lindroth et al. 2024). These approaches often overlook real-time causal signals of departure, leading to poor performance and interpretability. For robust deployment of *DFC*, predictive models must provide temporal precision while adapting to contextual signals that reflect pre-departure behavior.

Pre-departure behavior is shaped by diverse contextual cues, including user behavioral signals such as physical activity, smartphone engagement, and social interactions, as well as environmental factors like ambient light and sound. Smartphone passive sensing provides a scalable and unobtrusive way to capture these signals continuously, enabling a digital phenotyping approach aligned with the quantified-self paradigm (Torous et al. 2016). Prior work has shown that machine learning classifiers trained on sliding-window contextual features can improve first-daily departure prediction from home. Combined with DBSCAN, these models outperform historical-pattern baselines (Lee et al. 2024). Yet, framing departure prediction as classification leads to severe label imbalance, which becomes more pronounced at finer time resolutions for real-time inference.

Building on recent advances in deep survival analysis across domains such as healthcare, churn prediction, and mobility, we reformulate departure time prediction as a time-to-event (TTE) problem, applying survival analysis principles to estimate the probability distribution of departure. This reframing offers two key advantages: (1) it mitigates label imbalance by representing each day as a single observation instead of multiple binary windows, and (2) it supports real-time inference where the event time is unknown until departure, enabling probabilistic predictions. Unlike classification, TTE models predict survival functions, providing richer temporal modeling of departure likelihood. To this end, we propose a time-to-departure (TTD) modeling approach based on a Transformer architecture, leveraging its ability to capture temporal dependencies and achieve efficient parallel processing through multi-head attention. The model updates survival probabilities over discretized time grids using real-time contextual signals from passive sensing. This enables timely departure prediction and providing a robust foundation for successful deployment of *DFC* in real-world environments. Our contributions are threefold:

- We introduce the departure time prediction problem and formulate it as a TTE prediction task. This enables *DFC* to effectively reduce $t_{100\%}$ and extend battery lifetime.
- We design a Transformer-based TTD framework that enables token-wise streaming inference, incorporating an ordinal Gaussian-smoothed loss and regularization strategies for robust learning under dynamic contexts.
- We validate our method on a real-world dataset of 93 users, achieving significant performance gains over the baselines, and release the dataset for future research.

This work takes a first step toward establishing benchmark models and datasets for smart EV charging strategies that depend on accurate departure prediction, such as *DFC*. Our approach offers strong potential for social impact by reducing battery degradation and supporting sustainable EV usage in response to climate challenges.

Related Work

Departure Time Prediction

Departure time prediction problem is often treated as part of broader mobility modeling objectives, such as EV behavior prediction and trajectory forecasting.

In the EV domain, prior research has primarily focused on predicting charging-related behaviors, especially departure and arrival times. Most approaches leverage historical data and machine learning to forecast departure times from temporal usage trends. (Frendo, Gaertner, and Stuckenschmidt 2021) propose a regression-based framework for workplace charging, integrating predictions into a heuristic scheduling strategy. To capture sequential dependencies, (Khwaja, Venkatesh, and Anpalagan 2021) and (Boulakhbar et al. 2022) employ LSTM networks, targeting residential charging and e-taxi operations, respectively. Recently, (Lindroth et al. 2024) introduce online learning models for real-time departure and distance prediction, adapting to driving behaviors. However, existing methods largely rely on historical patterns without incorporating dynamic contextual signals influencing departure decisions.

Deep learning methods for mobility prediction largely fall into two branches: Recurrent Sequence Models and Neural Temporal Point Process (TPP) Models. **Recurrent Sequence Models.** ST-RNN (Liu et al. 2016) extends RNNs with time- and distance-specific transition matrices to model local spatiotemporal contexts for next-location prediction. DeepMove (Feng et al. 2018) employs historical attention and multi-modal embeddings to capture long-term, periodic mobility patterns. **Neural TPP Models.** RMTPP (Du et al. 2016) models event timing in continuous time via RNN-based intensity functions, and RSTPP (Yang, Cai, and Reddy 2018a) extends this with spatial context for check-in time prediction. TrajTPP (Zeng et al. 2025) introduces dual attention and spatio-temporal GRUs for joint location and travel-time prediction. Recent advances integrate graph structures and attention mechanism, as in STGNPP (Jin et al. 2023) for congestion event prediction and TANTPP (Zhang et al. 2024) for train delay forecasting.

While these approaches achieve robust performance in trajectory and event-sequence prediction, individual departure time prediction remains underexplored as a standalone task. Prior models leverage historical trajectories and inter-event relations, whereas our work focus on real-time contextual features for dynamic departure detection. As baseline, we employ state-of-the-art sequential architectures (*e.g.*, iTransformer (Liu et al. 2023)) to learn temporal or contextual dependencies from behavioral sequences.

Deep Survival Analysis

Early deep learning approaches for survival analysis extended the Cox proportional hazards (PH) framework. DeepSurv (Katzman et al. 2018) replaced the linear predictor with a neural network for non-linear risk estimation. Kvamme et al. (Kvamme, Borgan, and Scheel 2019) improved scalability and introduced non-proportional hazard extensions. DeepEH (Zhong, Mueller, and Wang 2021) unified CoxPH and AFT using a deep extended hazard formulation to account for environment-specific effects. DiffSurv (Vauvelle et al. 2023) reframed survival analysis as a differentiable ranking problem for concordance optimization.

In contrast, non-parametric models discard the proportional hazards assumption and directly learn the survival distribution as a flexible function. DeepHit (Lee et al. 2018) modeled discrete-time survival as a multi-task classification for competing risks. SurvPP (Kim 2023) used a permanental point process to estimate continuous-time hazards with time-varying covariates. Beyond architectural innovations, recent research has focused on improving robustness and fairness under distribution shifts. Hu and Chen (Hu and Chen 2024) proposed a distributionally robust framework to mitigate subgroup disparities. Stable Cox (Fan et al. 2024) reweights samples to identify stable predictors. In the mobility domain, RCR (Yang, Cai, and Reddy 2018b) applies recurrent-censored regression with RNNs to predict user check-in times by capturing spatiotemporal dependencies from past trajectories, while DILSA (Vahedian et al. 2019) proposes a two-stage deep survival framework to predict abnormal taxi demand surges within a five-hour horizon.

Despite recent advances in deep survival analysis (Lee, Yoon, and Van Der Schaar 2019; Wang and Sun 2022; Mesinovic, Watkinson, and Zhu 2024; Zhang, Zhao, and Xu 2025), existing methods either violate covariate assumptions, learning approach, or introduce auxiliary objectives that increase complexity. In this work, we focus on data-driven real-time departure detection using exogenous contextual features. Prior survival models are not directly comparable, as their theoretical foundations and architectures do not align with our setting (see extended version (Lee et al. 2025) for details). Thus, we build on the Transformer-based deep survival framework (Hu et al. 2021), which models survival probabilities over discretized time grids using ordinal regression. We extend this design with contextual feature embeddings and token-level survival updates, enabling streaming inference for timely, accurate departure detection.

Methodology

Problem Definition

We formulate real-time EV departure detection as a TTE prediction problem, specifically as a TTD task. Let a day be discretized into $T_{\max}$ intervals (*i.e.*, 5 minutes), indexed by $t \in \{0, 1, \ldots, T_{\max}\}$. For a given user-day sequence, the contextual feature at time t is denoted $\mathbf{x}_t \in R^d$, capturing smartphone-based sensing signals and temporal indicators such as time-of-day and day-of-week. The partial observation up to time t is $\mathcal{X}_{0:t} = \{\mathbf{x}_0, \ldots, \mathbf{x}_t\}$. We focus on the first significant departure of the day, which typically corresponds to the EV unplug event after long-duration charging. Let $t^* \in \{0, \ldots, T_{\max}\}$ denote this event time, defined as the interval 30 minutes prior to the actual departure to ensure sufficient time for fast charging to reach 100% SOC.

We assume a guaranteed daily departure in our target scenario, and thus all training samples are treated as uncensored. During inference, however, the model observes only partial sequences for which the event has not yet occurred, so the true event time remains unknown and each sequence is right-censored at the current time t.

Survival Analysis for Time-to-Departure

Survival analysis provides a statistical framework for modeling TTE outcomes under uncertainty and right-censoring. Let T denote the random variable representing the departure time, and let $\mathbf{X}$ denote the observed covariates. The event distribution is characterized by the survival function

$$S(t \mid \mathbf{X}) = P(T > t \mid \mathbf{X}),$$

which gives the probability that departure has not occurred by time t. In discrete time, the hazard function is

$$\lambda(t \mid \mathbf{X}) = P(T = t \mid T \geq t, \mathbf{X}),$$

representing the probability of departure at interval t given survival up to $t-1$. Defining the hazard complement $q(t \mid \mathbf{X}) = 1 - \lambda(t \mid \mathbf{X})$, the survival function becomes

$$S(t \mid \mathbf{X}) = \prod_{\tau=0}^{t} q(\tau \mid \mathbf{X}).$$

This discrete-time formulation characterizes the relationship between hazard and survival functions. In our model, we directly estimate the survival probability at each interval, denoted $\hat{S}(t \mid \mathbf{X})$, rather than predicting hazards explicitly.

Transformer-Based Time-to-Departure Modeling

Our framework predicts $\hat{S}(t \mid \mathbf{X})$ for all intervals in parallel using a Transformer-based architecture, enabling efficient survival estimation with real-time updates. Through self-attention, the model captures both regular daily routine and dynamically changing contextual patterns, supporting timely and adaptive departure predictions (Hu et al. 2021).

Objective. Given contextual observations up to time t, $\mathcal{X}_{0:t}$, the goal is to estimate the sequence of survival probabilities $\{\hat{S}(\tau \mid \mathbf{X})\}_{\tau=0}^{T_{\max}}$, which characterizes the likelihood that departure has not occurred by each interval. Formally, the model produces

$$\hat{S}_{0:T_{\max}} = f_\theta(\mathcal{X}_{0:t}),$$

where f_θ is the Transformer-based architecture parameterized by θ. During training, supervision is applied only up to the observed departure time t^*, enforcing survival for $t < t^*$ and failure at $t = t^*$. During inference, the model processes partial sequences in real time, updating survival estimates and triggering departure detection once the predicted survival probability falls below a decision threshold p. Figure 2 illustrates this real-time update process.

Input Representation. Each interval t is represented by a token integrating three components: (1) contextual features from passive smartphone sensing (*e.g.*, activity transitions, ambient environment), (2) absolute time features encoding the interval's position in the day, and (3) day-of-week features capturing weekday–weekend (and holiday) variation. Contextual and day-of-week features are concatenated and projected into a shared embedding space with layer normalization and dropout. Absolute time features are projected through a non-linear feed-forward network $g(\cdot)$ and scaled by a learnable parameter $\gamma \in R$. The time embedding is

$$\mathbf{z}_t^{\text{time}} = \gamma \cdot g(\text{abs\_time}_t).$$

Let $\alpha \in (0, 1)$ be a learnable parameter controlling the balance between contextual and temporal signals. The fused token embedding is then

$$\mathbf{h}_t = \alpha \cdot \mathbf{z}_t^{\text{context}} + (1 - \alpha) \cdot \mathbf{z}_t^{\text{time}}.$$

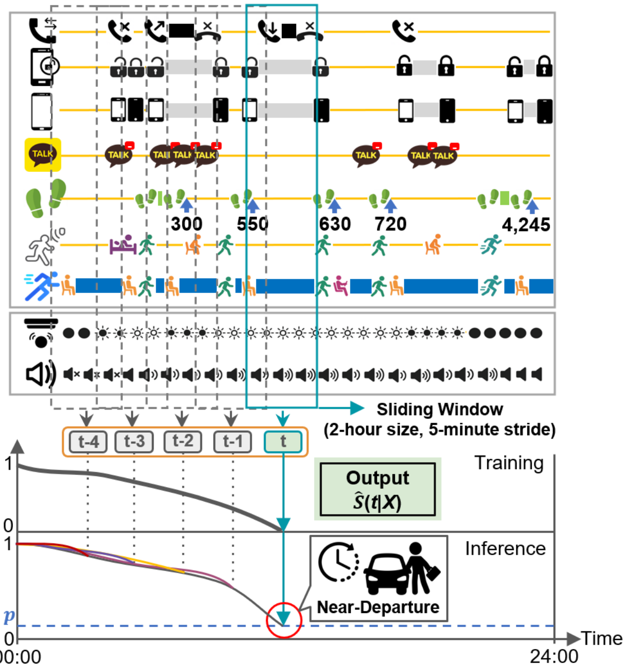

Figure 2: Real-time survival modeling with streaming contextual inputs. During training, the model estimates survival probabilities up to the observed departure event for uncensored sequences. At inference, survival estimates are updated incrementally as new tokens arrive, triggering a departure once the probability falls below a threshold p.

All token embeddings share a latent space of size d_{model} within the Transformer encoder.

Sequence Modeling. To encode temporal order, sinusoidal positional encodings are added to the fused embeddings. The resulting sequence $\{\mathbf{h}_0, \ldots, \mathbf{h}_{T_{\max}}\}$ is processed by a multi-layer Transformer encoder composed of self-attention and feed-forward sublayers. Unlike recurrent survival models, this design processes all time intervals in parallel, improving efficiency and enabling flexible partial-day inference.

Output Layer. For each interval t, the Transformer produces an output representation $\mathbf{u}_t \in R^{d_{\text{model}}}$, which is passed through a feed-forward network with a sigmoid activation:

$$\hat{S}(t \mid \mathbf{X}) = \sigma(W_o \mathbf{u}_t + b_o),$$

where $\hat{S}(t \mid \mathbf{X}) \in (0, 1)$ denotes the predicted survival probability $P(T > t \mid \mathbf{X})$. These per-interval estimates form a discrete survival curve that updates in real time as contextual observations evolve throughout the day.

Regularization Strategies. To improve robustness against over-reliance on temporal patterns, we introduce three mechanisms: (1) dropout-time, which randomly masks absolute time features during training; (2) time-scale, a learnable scalar that moderates the strength of temporal position embeddings; and (3) alpha-fusion, which balances contextual and temporal representations through a learnable weight. These strategies regularize temporal bias and promote generalization across diverse behavioral patterns.

Loss Function

We optimize a discrete-time ordinal regression objective inspired by (Hu et al. 2021). Let the training set contain N user-day sequences indexed by $i \in \{1, \ldots, N\}$. For each sequence i, let $\mathbf{X}_i$ denote the full contextual feature sequence and $t_i^* \in \{0, \ldots, T_{\max}\}$ denote the observed (uncensored) departure time. The model outputs per-interval survival probabilities $\hat{S}(t \mid \mathbf{X}_i)$ for $t \in [0, T_{\max}]$. Unlike (Hu et al. 2021), which parameterizes the hazard complement, we directly model $S(t)$ to avoid unnecessary intermediate variables, obtain cleaner uncertainty estimates, and simplify downstream inference for real-time departure detection. This survival estimate forms the basis for the ordinal regression loss used to supervise the model up to the observed event time t_i^*.

Ordinal Regression Loss. Since all training sequences contain an observed departure event, we optimize the negative log-likelihood for uncensored samples. The ordinal regression loss encourages the model to assign high survival probability before the event and a sharp probability drop at the event time:

$$\mathcal{L}_{\text{ord},i} = -\sum_{t=0}^{t_i^*-1} \log \hat{S}(t \mid \mathbf{X}_i) \; - \; \log\big(1 - \hat{S}(t_i^* \mid \mathbf{X}_i)\big).$$

The first term encourages the user to remain in the "not departed" state for all intervals before t_i^*, while the second enforces a transition to the departure state at t_i^*. Predictions after the event are excluded, as they are not meaningful in TTD modeling. Since non-departure days represent true non-events rather than right-censored sequences, training is performed solely on samples where a departure is observed.

Soft Supervision with Gaussian Smoothing. Short adjacent intervals (*i.e.*, 5 minutes) may share similar context, making strict supervision sensitive to minor timing shifts near the event. To mitigate this, we apply a normalized Gaussian weighting kernel centered at t_i^*, restricted to pre-event intervals, to smooth the loss around the departure time. This Gaussian-Smoothed Supervision (GSS) enhances robustness to label uncertainty and contextual noise immediately preceding the event:

$$w_i(t) = \frac{\exp\big[-(t - t_i^*)^2/(2\sigma^2)\big]}{\sum_{\tau \le t_i^*} \exp\big[-(\tau - t_i^*)^2/(2\sigma^2)\big]}, \qquad t \le t_i^*.$$

The smoothed ordinal regression loss becomes

$$\mathcal{L}_{\text{smooth},i} = -\sum_{t=0}^{t_i^*} w_i(t) \Big[\mathbf{1}_{t<t_i^*} \log \hat{S}(t \mid \mathbf{X}_i) + \mathbf{1}_{t=t_i^*} \log\big(1 - \hat{S}(t \mid \mathbf{X}_i)\big)\Big].$$

Event and Weekend Weights. We incorporate two weighting factors: (1) an event weight ω_e, which amplifies the failure term at t_i^* and scales the corresponding gradients; and (2) a weekend weight ω_w, which adjusts sequence-level contributions to account for irregular weekend patterns and stabilize the overall loss.

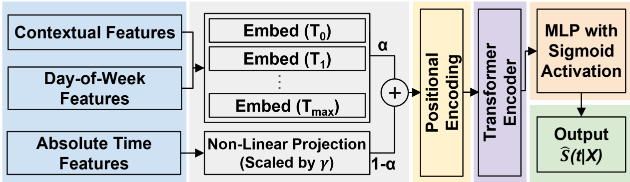

Figure 3: A diagram of the TTD model architecture. Contextual features and day-of-week embeddings are fused with absolute time features through the alpha-fusion mechanism, with absolute time scaled by a learnable parameter to control its contribution relative to contextual signal. The fused embeddings, combined with positional encoding, are processed by a multi-layer Transformer encoder to capture long-range temporal dependencies. The output layer predicts per-interval survival probabilities $\hat{S}(t \mid \mathbf{X})$ across all intervals in parallel during training and supports real-time inference by updating predictions as new tokens arrive.

Total Loss. The total loss aggregates the per-sequence losses with temporal and day-type weighting:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{N} \omega_w^{(i)} \left(\sum_{t=0}^{T_{\max}} w_i(t)\, \ell_{i,t} \right),$$

$$\text{where } \ell_{i,t} = \begin{cases} -\log \hat{S}(t \mid \mathbf{X}_i), & t < t_i^*, \\ -\omega_e \log\big(1 - \hat{S}(t_i^* \mid \mathbf{X}_i)\big), & t = t_i^*, \\ 0, & t > t_i^*, \end{cases}$$

$$\text{and } \omega_w^{(i)} = \begin{cases} \omega_w, & \text{if weekend/holiday}, \\ 1, & \text{otherwise (weekday)}. \end{cases}$$

This composite loss enforces survival consistency before t_i^*, sharp supervision at the event time, and robustness to day-type heterogeneity.

Experiments

Dataset

Participant Recruitment. We conducted an IRB-approved field study between May 2021 and July 2022, recruiting 506 Android users (ages 18–69) from the Seoul metropolitan area through social media and community postings. Given the low prevalence of EV ownership, we targeted general smartphone users under the assumption that daily departure routines are behaviorally consistent across populations.

Passive Sensing and Departure Labels. We developed *EV Analyzer* (*EVA*), an Android application for background passive sensing that collects nine behavioral and environmental streams (*i.e.*, activity transitions, step counts, significant motion, screen state, unlock state, app usage, call logs, ambient light, and ambient sound). Data were captured through a combination of periodic sampling (*e.g.*, light every 15 minutes) and event-driven triggers. Ground-truth departure times were derived from activity transition logs by identifying the first occurrence of the "in-vehicle" activity each day.

Data Preprocessing. To ensure data reliability, we applied three steps: (1) participant filtering based on data quality, (2) exclusion of temporally inconsistent records, and (3) removal of values outside realistic ranges. The final dataset comprises 93 participants with 42 days each, totaling 3,906 daily sequences.

Feature Extraction. We extracted features using statistical summaries for continuous signals and count- or duration-based metrics for event-driven logs. Additionally, we incorporated entropy measures and elapsed-time features that capture the time since the first occurrence of each key event. To determine the optimal sliding-window size, we conducted Spearman rank correlation analysis with statistical significance test. Features were computed within this window and updated every 5 minutes, resulting in 98 features. Additional correlation results are provided in the extended version (Lee et al. 2025).

Justification and Availability. This dataset is unique in providing fine-grained, real-world behavioral traces aligned with accurately labeled departure events, a combination not available in existing benchmarks. Since no comparable public dataset exists, custom data collection was required. To support reproducibility and future research, the anonymized dataset and full processing pipeline will be released upon publication (see extended version (Lee et al. 2025)).

Experimental Setups

Training Details. We train a generalized global model by splitting participants in a 4:1 ratio for training and testing. The training set is further split into training and validation subsets (4:1), yielding 60 users for training, 15 for validation, and 18 for testing. Model and training hyperparameters (*e.g.*, number of attention heads, learning rate) are tuned via grid search with 5-fold cross-validation.

The TTD model uses a 3-layer Transformer encoder with $d_{\text{model}} = 32$ and a single attention head, followed by a task-specific output layer. Training employs the Adam optimizer with a maximum of 100 epochs. All experiments are implemented in PyTorch and run on an NVIDIA RTX 4090 GPU. For robustness, each experiment is repeated with three random seeds (*i.e.*, 42, 43, 44), and we report the mean and standard deviation across runs. Additional implementation details, including hyperparameter ranges, are provided in the extended version (Lee et al. 2025).

Evaluation Metric. Performance is measured using mean absolute error (MAE) in hours, which provides an intuitive measure of temporal deviation. The predicted departure time is taken as the earliest time index at which the estimated survival probability falls below a decision threshold p. We vary this p from 0.05 to 0.20 in increments of 0.05 to assess its impact on accuracy, with 0.10 selected as the optimal value through hyperparameter tuning.

Baseline Comparison

We benchmark our TTD model against two baselines: (1) a regressor trained on historical statistics, and (2) a context-aware classifier combined with DBSCAN (Lee et al. 2024).

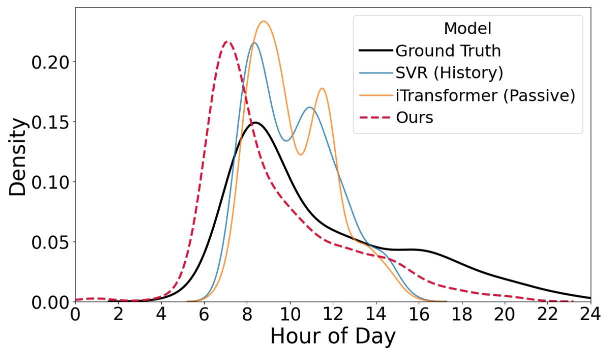

Figure 4: Kernel Density Estimation (KDE) plots of departure times across models. The proposed model shows the closest alignment with the ground-truth distribution compared to the best regressor and classifier baselines.

Historical Statistics-Based Regressors. These baselines are trained with statistics of observed departure times from the previous seven days: mean, standard deviation, minimum, maximum, kurtosis, and skewness, which are computed separately for all days, weekdays, and weekends, with an additional binary day-of-week indicator (19 features in total). We evaluate five models on this feature set: multiple linear regression (MLR), support vector regression (SVR), LightGBM Regressor, FT-Transformer (FTT (Gorishniy et al. 2021)) and iTransformer in a regression setting.

Context-Aware Classifiers. Following the strategy in (Lee et al. 2024), these baselines use the same contextual features as our method and employ the same set of model architectures in a classification setting. We further incorporate quantile regression to estimate a prediction boundary and apply DBSCAN to detect sharp probability rises, triggering departure detection only within the predicted range.

Performance Evaluation. Table 1 reports MAE (hours) across all days, weekdays, and weekends. Our model achieves the lowest error (2.20 hours), outperforming the best historical baseline (SVR, 2.57 hours) and the context-aware classifier (iTransformer, 2.59 hours). This corresponds to relative improvements of 14.4%/15.1% overall, 10.3%/11.7% on weekdays, and 21.3%/21.9% on weekends, demonstrating the effectiveness of modeling departure as a survival process rather than relying on point prediction or classification-based approaches. Notably, weekday and weekend performance is comparable across all models, despite expectations of greater weekend irregularity. We revisit this phenomenon in the Ablation Study section.

Distribution Analysis. Figure 4 presents kernel density estimates (KDE) of predicted versus ground-truth departure times. Historical baselines exhibit high concentration around global averages, particularly in morning and afternoon periods, failing to capture contextual routines relevant to departure. In contrast, our model's distribution closely tracks the ground-truth curve, demonstrating temporal adaptability despite minor deviations. A slight bias toward earlier predictions is observed, primarily due to the decision threshold

Data	Models	All days	Weekdays	Weekends
History	MLR	2.90	2.93	2.82
	SVR	2.57	2.52	2.63
	LGBM	2.58	2.54	2.59
	FTT	2.74	2.72	2.76
	iTransformer	2.61	2.59	2.65
Passive	LR	2.67	2.61	2.79
	SVC	2.66	2.64	2.70
	LGBM	2.72	2.81	2.65
	FTT	2.70	2.63	2.84
	iTransformer	2.59	2.56	2.65
Passive	**Ours**	**2.20**	**2.26**	**2.07**

Table 1: Performance results (MAE) for baselines and the proposed model across day categories.

Variants	All days	Weekdays	Weekends
w/o context	4.47 ± 0.18	4.88 ± 0.18	3.55 ± 0.16
w/o PE	4.25 ± 1.74	4.19 ± 1.54	4.40 ± 2.23
w/o time	3.01 ± 0.67	2.96 ± 0.60	3.10 ± 0.85
w/o DoW	2.64 ± 0.60	2.63 ± 0.45	2.66 ± 0.94
w/o alpha-fusion	2.55 ± 0.42	2.58 ± 0.32	2.49 ± 0.66
w/o t-scale & GSS	2.36 ± 0.04	2.42 ± 0.03	2.24 ± 0.09
Full Model	**2.20 ± 0.13**	**2.26 ± 0.13**	**2.07 ± 0.17**

Table 2: Ablation study on design choices. Variants are created by removing components from the Full Model.

$p = 0.1$. This suggests that adjusting the p improves practicality of *DFC* by providing a buffer time before departure.

Ablation Study

To evaluate the contribution of key design components, we conduct an ablation study along three dimensions: (1) input features, (2) architectural and regularization strategies, and (3) loss design. Table 2 presents MAE (hours) for all days, weekdays, and weekends across these variants.

Feature Contributions. Removing contextual features (w/o context) leads to the largest performance drop (4.47 hours), underscoring the critical role of behavioral and environmental indicators in capturing pre-departure routines. Removing absolute time features (w/o time) results in moderate degradation (3.01 hours), indicating that temporal position cues are important for capturing regular temporal patterns within each day. Excluding day-of-week features (w/o DoW) slightly worsens performance (2.64 hours), highlighting their usefulness in capturing weekly rhythms.

Architecture and Regularization. Removing positional encoding (w/o PE) substantially increases error (4.25 hours), confirming its necessity for temporal order modeling. Disabling alpha-fusion (w/o alpha-fusion) also degrades accuracy (2.55 hours), highlighting the benefit of balancing contextual and temporal representations.

Loss Design and Synergistic Regularization. Removing both time-scale and GSS (w/o t-scale & GSS) increases MAE to 2.36 hours, confirming their synergistic benefit. Time-scale improves temporal calibration across discretized time grids, while GSS provides soft supervision that mitigates over-penalization of near-boundary timing shifts.

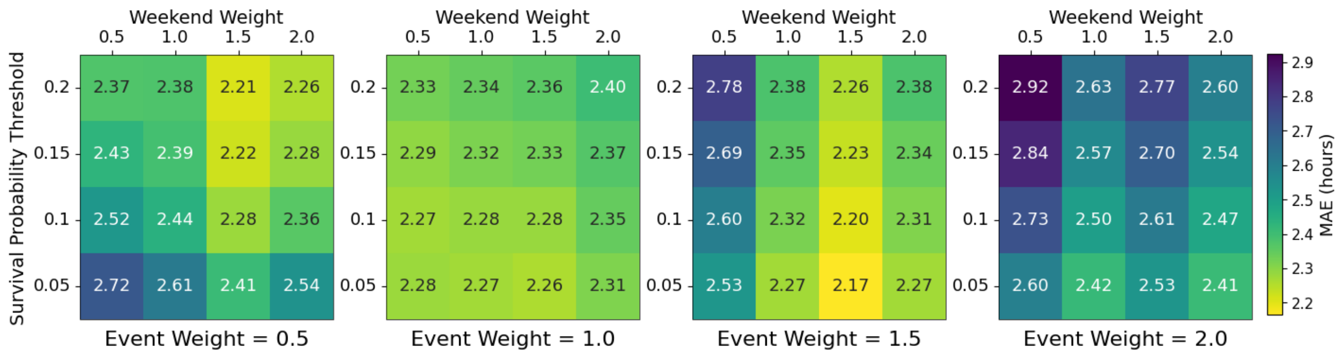

Figure 5: The impact of hyperparameters: event weight ω_e, weekend weight ω_w, and survival probability threshold p on the departure time prediction accuracy.

Overall, the ablation analysis confirms that all components jointly contribute to performance, with contextual features and positional encoding emerging as the most influential factors. Notably, several configurations exhibit lower MAE on weekends than on weekdays, contrary to the expectation of greater weekend irregularity. A plausible explanation lies in the dataset's collection period (2021–2022), which coincided with COVID-19 and post-pandemic transitions, where hybrid work schedules and flexible weekday routines introduced more variability than weekends.

Hyperparameter Sensitivity Analysis

We examine the sensitivity of our model to three key hyperparameters: event weight ω_e, weekend weight ω_w, and the survival probability threshold p. Figure 5 presents the MAE heatmaps across different combinations.

Effect of Event Weight. Each subplot corresponds to a fixed value of ω_e. Medium values (1.0–1.5) yield the best MAE overall, especially when paired with moderate ω_w. Small values weaken the event signal, while large values overemphasize the event point and harm calibration. Notably, the model's stable performance at $\omega_e = 1$ suggests that balancing survival and failure signals promotes robust training and stable outcomes. The best results are achieved when $\omega_e = 1.5$ is combined with $\omega_w = 1.5$ and a small p.

Effect of Weekend Weight. Small ω_w weakens weekend learning, causing the model to fall back on regular weekday patterns and making weekend cues less reliable. With large ω_e and p, these weak early cues combined with a softened threshold lead to unstable early predictions. With small ω_e and p, the event signal is underemphasized and the strict threshold delays detection, producing late predictions. Thus, small ω_w consistently harms weekend performance, while ω_e and p determine whether errors occur early or late.

Effect of Survival Probability Threshold. p shows a consistent MAE pattern centered around $\omega_e = 1$, where survival and failure signals are balanced. When $\omega_e = 1.0$–2.0, smaller p improves accuracy since the model learns sharper survival drops near the event. When $\omega_e = 0.5$, smaller p delays detection and worsens performance. In short, strict p works well when the event signal is emphasized.

Interactions and Practical Settings. Overall, moderate ω_e and ω_w (around 1.5) combined with a small p yield the most reliable performance. Small ω_w consistently harms weekend predictions, while extreme ω_e or p shifts the model toward unstable early or late detections. Using moderate weights and a strict threshold avoids these issues and provides accurate, stable real-time departure time predictions.

Personalization

We further evaluate user-specific adaptation through fine-tuning. All global model parameters remain frozen, and only the last Transformer layer and output layer are updated using user-specific data. Personalization yields a modest overall improvement (MAE: 2.20 $\rightarrow$ 2.13 hours), with larger gains on weekends (2.07 $\rightarrow$ 1.85 hours) than weekdays (2.26 $\rightarrow$ 2.23 hours). This trend aligns with earlier observations of lower weekend MAE, likely influenced by COVID-19 schedule shifts and the effect of ω_w on weekend variability. Additional analysis of intra-user variability, its impact, and a cold-start strategy is provided in the extended version (Lee et al. 2025). Future improvements may incorporate meta-learning or adaptive regularization to better balance global patterns with individual behaviors, and richer contextual signals from large language models may further enhance personalization.

Conclusion

This paper presents a Transformer-based TTD modeling framework as the foundation for *DFC*, a strategy aimed at reducing battery degradation and extending EV battery lifetime. Unlike prior work that relies on historical patterns, our approach uses real-time contextual signals from unobtrusive smartphone sensing and models departure as a survival process on discretized time grids. In a large-scale in-the-wild study with 93 participants and 3,906 daily sequences, our method outperforms historical and context-aware baselines, reducing MAE by 14.4%. Ablation studies highlight the role of design considerations in robust temporal modeling, while fine-tuning enables further personalization. The proposed dataset and framework establish a benchmark for real-time departure prediction in sustainable EV smart charging, supporting both battery longevity and carbon neutrality.

Ethical Statement

This study was approved by the Institutional Review Board (IRB) of our institution. All participants provided informed consent after being fully informed about the study objectives, procedures, and potential risks. Data collection was conducted through a custom mobile application under strict privacy safeguards. No personally identifiable information was collected, and all data were anonymized prior to analysis using secure hashing methods. Data transmission and storage were protected via AES-256 encryption and secured Linux-based servers. The dataset will be shared only in an anonymized and preprocessed form for research purposes. These measures ensure participant confidentiality and minimize privacy risks throughout the data lifecycle.

Acknowledgments

This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-MA2202-03.

References

Boulakhbar, A.; et al. 2022. Electric vehicles arrival and departure time prediction based on deep learning. *Energy Reports*.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 1555–1564.

Fan, S.; Xu, R.; Dong, Q.; He, Y.; Chang, C.; and Cui, P. 2024. Stable Cox regression for survival analysis under distribution shifts. *Nature Machine Intelligence*, 6: 1525–1541.

Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. DeepMove: Predicting human mobility with attentional recurrent networks. In *Proceedings of The Web Conference (WWW)*, 1459–1469.

Frendo, O.; Gaertner, N.; and Stuckenschmidt, H. 2021. Improving smart charging prioritization by predicting electric vehicle departure time. *IEEE Transactions on Intelligent Transportation Systems*, 22(10): 6646–6653.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34: 18932–18943.

Hu, S.; and Chen, J. 2024. Fairness in survival analysis with distributionally robust optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hu, S.; Fridgeirsson, E. A.; van Wingen, G.; and Welling, M. 2021. Transformer-based deep survival analysis. In *AAAI Spring Symposium on Survival Prediction–Algorithms, Challenges, and Applications (SP-ACA)*, volume 1 of *Proceedings of Machine Learning Research*, 1–17.

Jin, G.; Liu, L.; Li, F.; and Huang, J. 2023. Spatio-temporal graph neural point process for traffic congestion event prediction. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*, 14268–14276.

Katzman, J. L.; Shaham, U.; Cloninger, A.; Bates, J.; Jiang, T.; and Kluger, Y. 2018. DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1): 24.

Khwaja, A. S.; Venkatesh, B.; and Anpalagan, A. 2021. Performance analysis of LSTMs for daily individual EV charging behavior prediction. *IEEE Access*, 9: 154804–154818.

Kim, H. 2023. Survival permanental processes for survival analysis with time-varying covariates. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kvamme, H.; Borgan, Ø.; and Scheel, I. 2019. Time-to-event prediction with neural networks and Cox regression. *Journal of Machine Learning Research*, 20: 1–30.

Lee, C.; Yoon, J.; and Van Der Schaar, M. 2019. Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. *IEEE Transactions on Biomedical Engineering*, 67(1): 122–133.

Lee, C.; Zame, W. R.; Yoon, J.; and van der Schaar, M. 2018. DeepHit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2314–2321.

Lee, Y.; Hwang, J.; Kondoro, A. M.; Song, J.; and Noh, Y. 2025. Enabling Delayed-Full Charging Through Transformer-Based Real-Time-to-Departure Modeling for EV Battery Longevity. *arXiv preprint arXiv:2512.07723*.

Lee, Y.; Song, W.; Song, J.; and Noh, Y. 2024. Extending EV Battery Lifetime: Digital Phenotyping Approach for Departure Time Prediction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 8(4).

Lindroth, T.; Svensson, A.; Åkerblom, N.; Pourabdollah, M.; and Chehreghani, M. H. 2024. Online learning models for vehicle usage prediction during COVID-19. *IEEE Transactions on Intelligent Transportation Systems*, 25(8): 9387–9398.

Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, 194–200.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Mesinovic, M.; Watkinson, P.; and Zhu, T. 2024. DySurv: dynamic deep learning model for survival analysis with conditional variational inference. *Journal of the American Medical Informatics Association*, ocae271.

Torous, J.; Kiang, M. V.; Lorme, J.; and Onnela, J.-P. 2016. New tools for new research in psychiatry: A scalable and customizable platform to empower data-driven smartphone research. *JMIR Mental Health*, 3(2): e5165.

Vahedian, A.; Zhou, X.; Tong, L.; Street, W. N.; and Li, Y. 2019. Predicting urban dispersal events: A two-stage framework through deep survival analysis on mobility data. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 5199–5206.

Vauvelle, A.; Wild, B.; Eils, R.; and Denaxas, S. 2023. Differentiable sorting for censored time-to-event data. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wang, Z.; and Sun, J. 2022. Survtrace: Transformers for survival analysis with competing events. In *Proceedings of the 13th ACM international conference on bioinformatics, computational biology and health informatics*, 1–9.

Yang, G.; Cai, Y.; and Reddy, C. K. 2018a. Recurrent spatio-temporal point process for check-in time prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, 2203–2206.

Yang, G.; Cai, Y.; and Reddy, C. K. 2018b. Spatio-temporal check-in time prediction with recurrent neural network based survival analysis. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, 2976–2982.

Zeng, J.; Xiao, C.; Tang, J.; and Hu, C. 2025. Inquiring the next location and travel time: A deep-learning-based temporal point process for vehicle trajectory prediction. *IEEE Internet of Things Journal*, 12(9): 11874–11886.

Zhang, D.; Du, C.; Peng, Y.; Liu, J.; Mohammed, S.; and Calvi, A. 2024. A multi-source dynamic temporal point process model for train delay prediction. *IEEE Transactions on Intelligent Transportation Systems*, 25(11): 17865–17879.

Zhang, Z.; Zhao, Y.; and Xu, Y. 2025. TransformerLSR: Attentive joint model of longitudinal data, survival, and recurrent events with concurrent latent structure. *Artificial intelligence in medicine*, 160: 103056.

Zhong, Q.; Mueller, J.; and Wang, J.-L. 2021. Deep extended hazard models for survival analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*.