Personalized and Environment-Aware Battery Prediction for Electric Vehicles

Dongyue Li*
Shanghai Jiao Tong University
lidongyue12138@gmail.com

Guangyu Li*
University of Southern California
guangyul@usc.edu

Bo Jiang*
Didi Chuxing
scottjiangbo@didiglobal.com

Zhengping Che
Didi Chuxing
chezhengping@didiglobal.com

Yan Liu
University of Southern California
yanliu.cs@usc.edu

ABSTRACT

With the proliferation of Electric Vehicles (EVs), the range anxiety issue, the fear of losing power on the road, has become a concerning problem. To address this issue, we aim to provide an accurate prediction of the State of Charge (SoC) consumption on the road and interpretable tips for drivers. However, existing methods have not considered personal driving styles and environmental factors in modeling EV dynamics. They are based only on conventional state estimation or simple regression methods, suffering from low prediction accuracy and limited interpretability. Furthermore, no work has tested its models on noisy real-world, large-scale datasets. In this paper, we present a data-driven study of the battery power prediction problem with both synthetic data generated from a real physical model and real-world data collected from a transportation company. First, we design a novel state-space model, Extended Kalman Filter Network (EKFN), integrating driving behaviors and environmental factors. EKFN enjoys inference efficiency and timestep-wise interpretability from extended Kalman filter as well as the representation power from neural networks. Second, we establish an Estimation Maximization algorithm for learning the EKFN. Experimental results on the two datasets demonstrate that our model reduces the mean square error by at least 5% compared to baseline models for predicting the remaining battery power. Moreover, the local linear approximation of EKFN provides a way to interpret the significant factors that affect SoC consumption on the road.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**.

ACM Reference Format:
Dongyue Li, Guangyu Li, Bo Jiang, Zhengping Che, and Yan Liu. 2021. Personalized and Environment-Aware Battery Prediction for Electric Vehicles. In *MileTS '21: 7th KDD Workshop on Mining and Learning from Time Series, August 14th, 2021, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

*Asterisk indicates equal contribution to this research.

MileTS '21, August 14th, 2021, Singapore

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The growth of Electric Vehicles (EVs) starts to change the way that people transit [6, 19]. The EVs enjoy many advantages over vehicles powered by internal combustion engines, such as better energy efficiency, zero-emission, and lower operating cost. However, due to its limited EV driving range, long charging time, and limited charging stations, the *range anxiety* appears, which refers to the fear of losing power and seeing EVs shut down in the middle of a long-distance drive. It remains one of the major issues preventing the widespread adoption of EVs.

A straightforward way to alleviate *range anxiety* is by accurate and interpretable prediction of State of Charge (SoC) consumption on the road, which requires modeling the dynamics of EV battery status under real-world working conditions. The current emergence of connected vehicle networks enables the collecting of vast volumes of EV operating data, including battery status, driver profiles, and environment information. It offers a chance for in-depth analysis of EV dynamics to assist the driving. For example, given the current status, a driver could know the total energy consumption to the desired destination, the energy usage distribution along the trip, and the factors affecting the battery status.

However, mining real-world, large-scale EV operating data for accurate and interpretable EV battery dynamics modeling presents several challenges. **First**, the real operating status of EVs is governed by a complicated dynamic system. Most of the existing works only consider current battery status [4, 5, 7, 21]. However, besides the EV battery design, the driving style (e.g., drivers' aggressiveness and experience) and environmental attributes (e.g., wind, precipitation, and road angle) also matter. Moreover, how these factors contribute to energy consumption is hard to be elucidated by primitive intuitions or idealized physical models. **Second**, there is an accuracy v.s. interpretability trade-off. Classical state-space models, e.g., Kalman Filter (KF) and its extensions [24], are based on a two-stage prediction-correction approach. Although enjoying fine efficiency and interpretability, these models cannot handle highly non-linear dynamics between EV battery status and heterogeneous attributes. On the other hand, neural networks have excellent representation power to handle the non-linear mappings of heterogeneous data. However, the black-box nature limits their interpretability and hinders their application to battery status prediction. **Third**, the ground truth is unavailable for learning a prediction model since battery status data collection is a noisy measurement process. This raises the problem of simultaneously identifying the actual system state and learning the prediction model from noisy observations.

To overcome these challenges, in this work, we 1) establish a data-driven model to capture the complicated dynamics of EV batteries under real working conditions and 2) provide an accurate and interpretable prediction of the SoC consumption considering both the behavioral and environmental factors. We summarize our contributions as follows.

- We present a Neural Network (NN)-based Extended Kalman Filter (EKF) system, **Extended Kalman Filter Network (EKFN)**, for modeling. EKFN leverages the design principle of EKF and integrates a NN prediction model into the dynamic system. This model enjoys both the inference efficiency and interpretability from EKF and non-linear representation power from NN.
- We design an Expectation Maximization (EM) learning algorithm that iteratively updates the model parameters and infers actual states from noisy observations. We identify the E-step as extended Kalman smoothing and the M-step as maximizing the observation likelihood through stochastic backpropagation. Besides, we interpret the predictions by taking the linear approximation of the prediction process to identify affecting factors.
- We evaluate EKFN on battery energy prediction tasks and compare its performance with several competitive baselines. The experimental results show that EKFN outperforms the baselines by at least 5% in terms of SoC prediction accuracy on a synthetic and a real-world dataset. Besides, we provide representative case studies of the significant factors that affect the SoC consumption based on the linear approximation interpretation.

2 MODEL DESCRIPTION

In this section, we describe our proposed model and the corresponding tasks. We treat an EV as a discrete-time non-linear dynamical system, specified by the tuple $\{\mathbf{X}, \mathbf{U}, \mathbf{S}, \mathbf{Z}, \mathbf{f}, \mathbf{h}\}$. For a journey with time horizon T, we describe the notations as follow:

- Vehicle state variable $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_\mathrm{T}\}$ represents the hidden full information of the EV battery, indicating the true EV status at each time step, which can hardly be measured exactly in practice.
- Environmental variable $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{\mathrm{T}-1}\}$ represents traffic and trip information during the journeys, e.g., wind speed, weather, and temperature.
- Driver-specific variable $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{\mathrm{T}-1}\}$ describes driver's age, gender, year of driving, and other information related to his/her driving style.
- Observation variable $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_\mathrm{T}\}$ stands for the noisy observation of the EV battery status at each time step, e.g., noisy corrupted motor voltage, current, and revolutions per minute.

Model. We model the dynamical system by the proposed *Extended Kalman Filter Network* (EKFN), built upon the extended Kalman filter model and empowered by the neural networks. Figure 1 illustrates our model. EKFN aims to approximate and recover the actual dynamical process of an EV motor battery. In EKFN, at every time step t, there is a hidden state $\mathbf{x}_t$ representing all the information of the physical battery system itself. In the dynamical process of this model, a new hidden state $\mathbf{x}_{t+1}$ is generated from the last hidden state $\mathbf{x}_t$ and modulated by the driving behavior $\mathbf{s_t}$ and environmental factor $\mathbf{u}_t$. The observation vector $\mathbf{z}_{t+1}$ could be generated from the new hidden state $\mathbf{x}_{t+1}$ and represent what we could observe from the battery. These generation processes are

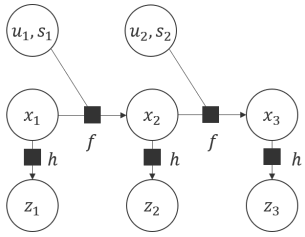

Figure 1: The dynamical process of our proposed model.

denoted as the **prediction** and the **observation** process, respectively. These processes could be disturbed by random noises, such as noises in sensor circuits while observing the batteries' indicators.

The dynamical process of prediction and observation is governed by a prediction function $\mathbf{f}(\cdot)$ and an observation function $\mathbf{h}(\cdot)$ as follows:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{s}_t) + \boldsymbol{\omega}_t \quad (1)$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{v}_t \quad (2)$$

where $\mathbf{x}_t$, $\mathbf{u}_t$, $\mathbf{s}_t$, $\mathbf{z}_t$ are the hidden state vector, the environmental variables, the driver-specific variables, and the observation at time t, repectively. $\boldsymbol{\omega}_t, \mathbf{v}_t$ are independent zero-mean Gaussian noises with covariance matrix $\mathbf{Q}$ and $\mathbf{R}$, respectively. The driver-specific variables $\mathbf{s}_t$ and environmental variables $\mathbf{u}_t$ are addressed as action data in the following sections. $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are two non-linear functions, modeling the prediction and observation processes. In EKFN, both functions are parameterized by neural networks, e.g., multilayer perceptron (MLP).

Tasks. The central task in modeling the EV system is the *system identification* task, i.e., understanding how vehicle states evolve in a journey. Formally, the goal of this task is to learn prediction function $\mathbf{f}(\cdot)$ and observation function $\mathbf{h}(\cdot)$ from n journeys, in which each journey contains a set of noisy observation $\mathbf{Z}$, environmental information $\mathbf{U}$ and driver-specific information $\mathbf{S}$.

On the application side, we are interested in predicting the battery states of an EV. In this paper, we focus on providing an accurate prediction of SoC consumption on the road and interpretable results for drivers. With that said, once the system is identified, the system will be applied to the *vehicle status prediction* task, i.e, given the initial noisy observation of the vehicle $\mathbf{z}_1$ with a planned journey environments $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{\mathrm{T}-1}\}$ and potential behaviors $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{\mathrm{T}-1}\}$, the model predicts vehicle status $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_T\}$ at each time step along the journey.

3 PARAMETER LEARNING

Before the vehicle status prediction, we need to determine parameters of the functions $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ that control the system dynamics, i.e., the *system identification* task. In the task, no ground truth of vehicle states $\mathbf{X}$ is available to learn the functions $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ parameterized by θ. To determine the parameters, we aim to learn model parameters from noisy observation data $\mathbf{z}_{1:\mathrm{T}}$ and action data $\mathbf{u}_{1:\mathrm{T}-1}$ and $\mathbf{s}_{1:\mathrm{T}-1}$. In order to learn the framework, the ultimate goal of *system identification* is to maximize the log-likelihood of all

observations, which is summed along all journeys:

$$\theta^* = \arg\max_{\theta} \sum_{\mathrm{J}} L^j(\theta) = \arg\max_{\theta} \sum_{\mathrm{J}} \log p_\theta(\mathbf{z}^j_{1:\mathrm{T}}) \tag{3}$$

where $p_\theta(\mathbf{z}^j_{1:\mathrm{T}}) = p(\mathbf{z}^j_{1:\mathrm{T}}|\mathbf{x}_0, \mathbf{U}^j, \mathbf{S}^j, \theta)$ is the marginal log likelihood of the observations along one journey j, $\theta = \{\mathbf{f}, \mathbf{h}, \mathbf{Q}, \mathbf{R}\}$ and $\mathbf{x}_0$ indicates prior distribution of model initial status. In the following derivation, we leave out journey indexes j for brevity. The marginal log-likelihood distribution $p_\theta(\mathbf{z}_{1:\mathrm{T}})$ can be obtained by integrating along all the hidden state $\mathbf{x}_{1:\mathrm{T}}$. However, the integration cannot be analytically solved. Therefore, we aim to maximize the evidence lower bound (ELBO) $F(q, \theta) \leq L(\theta)$ with respect to both q and θ. Note that here q is a hidden distribution of $\mathbf{x}_{1:\mathrm{T}}$, and θ is the original model parameter set.

We establish an Expectation Maximization (EM) algorithm to optimize the above object, which is capable of both parameter learning and state inference. The EM algorithm alternates between maximizing F with respect to distribution q (state inference and denoising) and the parameter θ (parameter learning).

3.1 E-Step Optimization

Starting from some initial parameters, θ_0 and q_0, we alternatively apply the E-step and the M-step. For the E-step, the optimization problem aims to maximize the ELBO F with respect to the distribution q over hidden states $\mathbf{x}_{1:\mathrm{T}}$:

$$q_{k+1} \leftarrow \arg\max_{q} F(q, \theta_k) \tag{4}$$

where the maximum is achieved when q is exactly the conditional distribution by Jensen's Inequality:

$$q^*_{k+1}(\mathbf{x}_{1:\mathrm{T}}) = p(\mathbf{x}_{1:\mathrm{T}}|\mathbf{x}_0, \mathbf{z}_{1:\mathrm{T}}, \mathbf{U}, \mathbf{S}, \theta_k) \tag{5}$$

when the lower bound becomes an equality $F(q^*_{k+1}, \theta_k) = L(\theta_k)$.

In our proposed model, the E-step corresponds exactly to solving the smoothing problem, which estimates the hidden state trajectory given both the observations inputs and the parameter values. The method to solve the smoothing problem is identified as **Extended Kalman Smoothing** (EKS) [8]. We apply a square root filtering algorithm [11] to avoid getting the non-positive-definite covariance matrix in the filtering process. The derivation of EKS and square root filtering is described in the supplementary material.

3.2 M-Step Optimization

For the M-step, the optimization problem aims to maximize F w.r.t. the parameter set θ under the fixed hidden distribution q_{k+1}:

$$\theta_{k+1} \leftarrow \arg\max_{\theta} F(q_{k+1}, \theta) \tag{6}$$

We show that the maximum of the M-step is obtained by maximizing the expectation of the observation likelihood under the hidden distribution of the states:

$$\theta_{k+1} = \arg\max_{\theta} E_{q^*_{k+1}}[\log p(\mathbf{z}_{1:\mathrm{T}}, \mathbf{x}_{1:\mathrm{T}}|\mathbf{x}_0, \mathbf{U}, \mathbf{S}, \theta)] \tag{7}$$

where $q^*_{k+1}(\mathbf{x}_{1:\mathrm{T}}) = p(\mathbf{x}_{1:\mathrm{T}}|\mathbf{x}_0, \mathbf{z}_{1:\mathrm{T}}, \mathbf{U}, \mathbf{S}, \theta_k)$.

We apply the stochastic backpropagation [22] and ancestral sampling for estimating all these gradients and train the model by Stochastic Gradient Descent (SGD) algorithm. We describe the overall procedures of the algorithm in Algorithm 1.

Algorithm 1 Learning EKFN with the EM Algorithm

Input: the observation variable $\mathbf{z}_{1:\mathrm{T}}$, environmental variable $\mathbf{U}$, and driver-specific variable $\mathbf{S}$ during all the journeys.

1: Initialize model parameters θ as θ_0 and hidden distribution q as q_0.
2: **while** not converged **do**
3: Obtain $q_{k+1}(\mathbf{x}_{1:\mathrm{T}})$ through the EKS by Equation (5).
4: Estimate the gradients of the joint likelihood $\log p(\mathbf{z}_{1:\mathrm{T}}, \mathbf{x}_{1:\mathrm{T}})$ under the hidden distribution q_{k+1} in Equation (7).
5: Update θ_{k+1} by SGD.
6: **end while**

4 PREDICTION INTERPRETATION

Following the idea of the extended Kalman filtering process, we interpret the prediction results by linearly approximating the prediction function and emission function in the dynamic process. We first calculate the Jacobian matrix of prediction and emission function for the input state $\mathbf{x}_k$ at step k and multiply them together to form a linear function of observation $\mathbf{z}_{k+1}$ about action data $\mathbf{u}_k, \mathbf{s}_k$. We then use the weight matrix in the linear function as the local changing rate for the dynamic process:

$$F_{k+1,k} = \frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{s})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_k} \tag{8}$$

$$H_{k+1} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_{k+1}} \tag{9}$$

where $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{s}_k)$. Thus, we can represent the observation vector as a linear function of the input action data:

$$\mathbf{z}_k = \frac{\partial h(\mathbf{x}_t)}{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{s})} \frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{s})}{\partial \mathbf{x}} [\mathbf{x}, \mathbf{u}, \mathbf{s}] \tag{10}$$

$$= H_{k+1} F_{k+1,k} [\mathbf{x}, \mathbf{u}, \mathbf{s}] = W_k [\mathbf{x}, \mathbf{u}, \mathbf{s}] \tag{11}$$

where $W_k = H_{k+1} F_{k+1,k}$ and $[\cdot]$ denotes concatenation. If the SoC variable is the $i-$th variable in the observation vector, then the changing rate of SoC w.r.t. the environmental variable $\mathbf{u}$ and driver-specific variable $\mathbf{s}$ is the i-th row of weight matrix W_k. The factor corresponding to the largest weight in the i-th row is the most significant factor affecting the SoC consumption at time step k.

5 EXPERIMENTS

5.1 Experiment Design

Datasets. We evaluate model performance on two datasets, a Real-World Human Driving Records dataset (*RWHDR*) and a Vehicle Energy Synthetic Dataset (*VESD*). The RWHDR dataset is extensive EV operating data collected from over 30,000 drivers and 100,000 trips in 50 different cities from a transportation company.[1] This dataset records more than 100 features regarding EV design, driver profile, and environment information. To the best of our knowledge, it is the largest dataset in the electric vehicle domain. To test in various time lengths, we generate three different sets of driving track data of three different sampling rates: 50s, 100s, and 200s. Corresponding time lengths of datapoints in each track data are $T = 20, 15$, and 10 (lasting 1000s, 1500s, and 2000s), respectively. We

[1] All data are anonymized, desensitized, and aggregated.

Table 1: Test RMSE (lower is better) of prediction results on the *VESD* dataset. Full battery is at 100 SoC level.

Models	RF	MLP	LSTM	PLSTM
RMSE	1.334	1.201	1.206	1.203
Models	VAR	KF	DMM	EKFN
RMSE	1.245	1.679	1.155	**1.107**

Table 2: Test RMSE (lower is better) of prediction results on the three generated datasets of the *RWHDR* dataset.

	T-20-Rate-5	*T-15-Rate-10*	*T-10-Rate-20*
RF	0.703	1.343	1.907
MLP	0.672	1.255	1.743
LSTM	0.689	1.611	1.784
PLSTM	0.684	1.274	1.750
VAR	0.667	1.244	1.723
KF	1.005	1.349	2.320
DMM	0.684	1.315	1.855
EKFN-I	0.648	**1.182**	1.810
EKFN-M	**0.628**	1.231	**1.636**

denote these three datasets as *T-20-Rate-5, T-15-Rate-10*, and *T-10-Rate-20* where the first number indicates the number of time steps in one track datapoint and the second number indicates the sampling rates. The VESD dataset is a synthetic EV operating dataset based on a real physical model [27]. We generate 100 battery profiles with different battery initialization states and collect 3000 data points of length $T = 10$. A more detailed data description is in the supplementary material.

Settings. We apply the proposed model to a prediction task. This task is to predict the delta SoC of each time step. It requires using initial vehicle state observation data and action data of the whole track. For example, if using the *T-20-Rate-5* data, which means tracks are of track length 1000s and of sampling rate 50s, the task is to use state data of first 50s and action data of the whole 1000s to predict the delta SoC for the rest 19 samples that are records of 950s. We implement two kinds of EKFN in our experiments: one with emission function parameterized by an MLP model (EKFN-M) and one with identical emission (EKFN-I). More detailed experimental settings are described in the supplement material.

Baselines. We compare our model with Random Forest (RF), Multilayer Perceptron (MLP), Long-Short Term Memory (LSTM) [9], Phased LSTM (PLSTM) [20], Vector AutoRegression (VAR), Kalman Filter (KF), and Deep Markov Models (DMM) [15]. To ensure a fair comparison, we use roughly the same amount of parameters for all models. Meanwhile, train/valid/test sets were split as 60/10/30, we chose the best models based on the performance on the validation set, and we reported the average Root Mean Squared Error (RMSE) on the held-out test dataset of three random seeds.

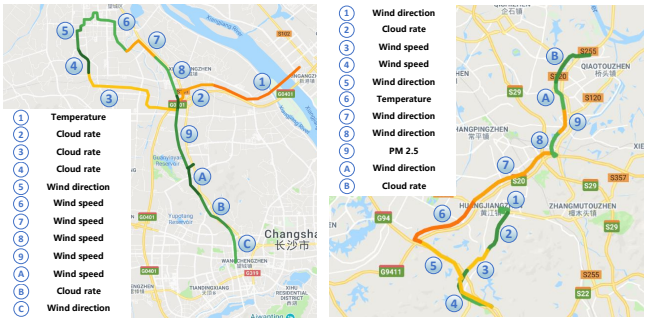

Figure 2: Two sampled driving trajectories. Driving trajectories in various colors indicate the expected SoC drop in a fixed time length. Dark red indicates a severe SoC drop and light green indicates a moderate SoC drop. Environmental factors affecting SoC change most significantly at every time step are labeled to the corresponding trajectory.

5.2 Quantitative Results

Table 1 shows the RMSE results on the *VESD* dataset. Table 2 shows prediction results on generated *T-20-Rate-5, T-15-Rate-10*, and *T-10-Rate-20* datasets. Our model beats baseline models by at least 5% on all four datasets with all features. Comparing two different implementations, EKFN-I performs better than EKFN-M on the dataset of sampling rate 10s. EFKN-M outperforms EKFN-I on the dataset of sampling rates 5s and 20s. In all our experiments, both implementations of our proposed model outperform other models by a significant margin. The denoising ability of the generative model on the EV driving dataset, especially our proposed EKFN, helps improves upon regression models. Moreover, the implementation of neural network emission of our proposed model, EKFN-M, captures data distribution of longer time better than EKFN-I which is with identical emission.

5.3 Interpretation Cases

To illustrate the interpretability of our proposed model, we conduct case studies on samples from the *RWHDR* dataset and identify the primary factor that affects SoC to drop with the interpretation method in Section 4. By taking the local linear approximation of the prediction function, we can obtain a linear model for delta SoC at every time step. Since all the data are normalized to the same scale, the linear model shows which factors influence current SoC consumption to the most significant extend. First, as expected, speed is the most crucial factor influencing the SoC drop most of the time. This finding aligns with the common knowledge that driving fast can lead to more energy costs. Besides, we also demonstrate the environmental factor affecting the SoC change in real driving trajectories in Figure 2. We show the delta SoC at each time step on the track by colors. Darker red indicates the SoC drop is severe, and green indicates the driving is energy-efficient. Besides, we label each track with its most significant environmental factor causing SoC consumption. Given this information, drivers can know what affects their SoC consumption and make the corresponding route or driving action changes to control their SoC drop. More interpretation cases are shown in the supplementary material.

6 CONCLUSIONS AND FUTURE WORK

We propose EKFN, a novel state-space model that parametrizes the extended Kalman filter model with neural networks for EV SoC prediction. Extensive experiments show that this model beat other baselines by at least 5%. Local linear representation of the proposed model provides an interpretable way to explain predictions at each time step. We visualize the significant factors affecting SoC consumption based on this interpretation on maps. Future work will focus on predicting attainable destinations and illustrating the predicted range with real-time traffic data.

REFERENCES

[1] Pieter Abbeel and Andrew Y Ng. 2005. Learning first-order Markov models for control. In *Advances in Neural Information Processing Systems*.

[2] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*.

[3] D Duckworth. 2013. pykalman, an implementation of the kalman filter, kalman smoother, and em algorithm in python. https://pykalman.github.io/index.html

[4] João C Ferreira, Vítor Monteiro, and João L Afonso. 2013. Dynamic range prediction for an electric vehicle. In *World Electric Vehicle Symposium and Exhibition*.

[5] Joao C Ferreira, Vítor Duarte Fernandes Monteiro, and João L Afonso. 2012. Data mining approach for range prediction of electric vehicle. In *Conference on Future Automotive Technology-Focus Electromobility*.

[6] João C Ferreira, Vítor Duarte Fernandes Monteiro, and João L Afonso. 2014. Electric vehicle assistant based in driver profile. *International Journal of Electric and Hybrid Vehicles* (2014).

[7] Stefan Grubwinkler, Tobias Brunner, and Markus Lienkamp. 2014. Range prediction for EVs via crowd-sourcing. In *IEEE Vehicle Power and Propulsion Conference*.

[8] Simon Haykin. 2004. *Kalman filtering and neural networks*. John Wiley & Sons.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* (1997).

[10] Shyh-Chin Huang, Kuo-Hsin Tseng, Jin-Wei Liang, Chung-Liang Chang, and Michael Pecht. 2017. An online SOC and SOH estimation model for lithium-ion batteries. *Energies* (2017).

[11] Paul Kaminski, A Bryson, and Stanley Schmidt. 1971. Discrete square root filtering: A survey of current techniques. *IEEE Trans. Automat. Control* (1971).

[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Jonathan Ko and Dieter Fox. 2009. GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Autonomous Robots* (2009).

[14] Jonathan Ko and Dieter Fox. 2011. Learning GP-BayesFilters via Gaussian process latent variable models. *Autonomous Robots* (2011).

[15] Rahul G Krishnan, Uri Shalit, and David Sontag. 2017. Structured inference networks for nonlinear state space models. In *AAAI Conference on Artificial Intelligence*.

[16] Benson Limketkai, Dieter Fox, and Lin Liao. 2007. CRF-filters: Discriminative particle filters for sequential state estimation. In *IEEE International Conference on Robotics and Automation*.

[17] Zifan Liu. 2017. Battery Aging Studies Based on Real-World Driving. (2017).

[18] Marjan Momtazpour, Ratnesh Sharma, and Naren Ramakrishnan. 2014. An integrated data mining framework for analysis and prediction of battery characteristics. In *Innovative Smart Grid Technologies-Asia*.

[19] Vítor Duarte Fernandes Monteiro, João C Ferreira, JG Pinto, JC Fernandes, and João L Afonso. 2017. New opportunities and perspectives for the electric vehicle operation in smart grids and smart homes scenarios. In *International Conference on Vehicle Technology and Intelligent Transport Systems*.

[20] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. 2016. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*.

[21] Peter Ondruska and Ingmar Posner. 2014. The route not taken: Driver-centric estimation of electric vehicle range. In *International Conference on Automated Planning and Scheduling*.

[22] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* (2014).

[23] Skipper Seabold and Josef Perktold. 2010. Statsmodels: Econometric and statistical modeling with python. In *Python in Science Conference*. Scipy.

[24] Greg Welch, Gary Bishop, et al. 1995. An introduction to the Kalman filter. (1995).

[25] Thomas A Wenzel, KJ Burnham, MV Blundell, and RA Williams. 2006. Dual extended Kalman filter for vehicle state and parameter estimation. *Vehicle System Dynamics* (2006).

[26] Yinjiao Xing, Eden WM Ma, Kwok L Tsui, and Michael Pecht. 2011. Battery management systems in electric and hybrid vehicles. *Energies* (2011).

[27] Caiping Zhang, Jiuchun Jiang, Weige Zhang, and Suleiman M Sharkh. 2012. Estimation of state of charge of lithium-ion batteries used in HEV using robust extended Kalman filtering. *Energies* (2012).

A RELATED WORK

The battery prediction problem arises from the growth of EVs. on one side of existing works, previous works [1, 14, 25] aim to solve the prediction problem through vehicle state and parameter estimation. Conventional methods start from analyzing and designing physical models of battery [10, 17, 26]. Most works in state estimation build dynamics systems trough Kalman Filter (KF) and Extended Kalman Filter (EKF) [25], Gaussian Process (GP) [14], First-order Markov model [1], GP-Bayes Filters [13], and Particle Filters [16]. Among them, KF is the most frequently favored by researchers, and non-linear modeling of the state prediction and the observation emission is crucial for estimating vehicle state. However, the applications of these methods to battery prediction problem mostly only consider battery or vehicle model. They do not take driver behavior or road environmental conditions into consideration. Furthermore, their non-linear functions always depend on hand-designed functions and cannot handle complex relationships among real data features. Remedies for both issues have been elaborated by our method.

Neural network (NN)-based models emerge as a powerful tool to learn the non-linear dynamics among features and has shown their representation ability in various tasks, such as image classification, function approximation, and data processing. Another line of existing works treats the battery prediction problem through data mining approaches [18]. They mostly directly apply existing prediction models and address the prediction task as a simple regression problem [4, 5, 7, 21]. Nevertheless, they have not provided unique designs for EV battery systems. Their results are hard to be interpreted due to the black-box nature of these models. Moreover, none of the existing works has been tested in a large dataset to our best knowledge. Therefore, in this work, our method contributes by taking advantage of neural networks and extended Kalman filter and building a personalized and environmental-aware system for the EV battery system.

B ALGORITHM DERIVATION

B.1 The Evidence Lower Bound

The derivation of the evidence lower bound (ELBO) is shown as follows.

$$
\begin{aligned}
L(\theta) &= \log p(\mathbf{z}_{1:T}|\mathbf{x}_0, \mathbf{u}_{1:T-1}, \theta) \\
&= \log \int_{\mathbf{x}_{1:T}} p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{u}_{1:T-1}, \theta) d\mathbf{x}_{1:T} \\
&= \log \int_{\mathbf{x}_{1:T}} q(\mathbf{x}_{1:T}) \frac{p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{u}_{1:T-1}, \theta)}{q(\mathbf{x}_{1:T})} d\mathbf{x}_{1:T} \\
&\geq \int_{\mathbf{x}_{1:T}} q(\mathbf{x}_{1:T}) \log \frac{p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{u}_{1:T-1}, \theta)}{q(\mathbf{x}_{1:T})} d\mathbf{x}_{1:T} \\
&= \int_{\mathbf{x}_{1:T}} q(\mathbf{x}_{1:T}) \log p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{u}_{1:T-1}, \theta) d\mathbf{x}_{1:T} \\
&\quad - \int_{\mathbf{x}_{1:T}} q(\mathbf{x}_{1:T}) \log q(\mathbf{x}_{1:T}) d\mathbf{x}_{1:T} \\
&= F(q, \theta)
\end{aligned}
$$

B.2 Implementation of Extended Kalman Smoothing

We apply a square root filtering algorithm [11] to overcome the issue of non-positive definite of the covariance matrix. In the square root filter, the key idea is to represent every positive semi-definite symmetric matrix by the product of a lower triangular matrix and its transpose. Here we define:

$$\mathbf{P}_k = \mathbf{S}_k \mathbf{S}_k^T \tag{12}$$

$$\mathbf{P}_k^- = \mathbf{S}_k^- \mathbf{S}_K^{-T} \tag{13}$$

$$\mathbf{Q} = \mathbf{U}\mathbf{U}^T \tag{14}$$

$$\mathbf{R} = \mathbf{V}\mathbf{V}^T \tag{15}$$

where $\mathbf{Q}$ and $\mathbf{R}$ indicate covariance matrices of Gaussian noise in transition and emission process. $\mathbf{P}_k$ and $\mathbf{P}_k^-$ denote covariance matrices of posterior state estimation and prior state estimation at time k. For the Extended Kalman Filter model, we take the local linear for filtering:

$$\mathbf{F}_{k+1,k} = \frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{s})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_k} \tag{16}$$

$$\mathbf{H}_k = \frac{\partial h(\mathbf{x}_t)}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_k^-} \tag{17}$$

where $\mathbf{x}_k$ denotes the posterior estimation of $\mathbf{x}$ at time k, and $\mathbf{x}_k^-$ indicate prior estimation. Based on the local linear approximation, we can also interpret time-step change through the linear surrogate model. The square root filtering algorithm is to update the following steps from time $t = 1$ to $t = \mathrm{T}$:

(1) Time Update

$$\mathbf{x}_k^- = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{s}) \tag{18}$$

$$\mathbf{P}_k^- = \mathbf{F}_{k,k-1} \mathbf{P}_{k-1} \mathbf{F}_{k,K-1}^T + \mathbf{Q} \tag{19}$$

$$\mathbf{S}_k^- = \mathbf{P}_k^{\frac{1}{2}} \tag{20}$$

(2) Measurement Update

$$\mathbf{B}_k = \mathbf{S}_k^{-T} \mathbf{H}_k^T \tag{21}$$

$$\mathbf{G}_k = [\mathbf{R} + \mathbf{B}_k^T \mathbf{B}_k]^{\frac{1}{2}} \tag{22}$$

$$\mathbf{S}_k = \mathbf{S}_k^- - \mathbf{S}_k^- \mathbf{B}_K \mathbf{G}_k^{-T} (\mathbf{G}_k + \mathbf{V})^{-1} \mathbf{B}_k^T \tag{23}$$

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{S}_k^- \mathbf{B}_k \mathbf{G}_k^{-T} \mathbf{G}_k^{-1} (\mathbf{z}_k - h(\mathbf{x}_k^-)) \tag{24}$$

where $A^{\frac{1}{2}}$ indicates Cholesky decomposition where A is a positive semi-definite matrix.

B.3 M-Step Expectation Estimation

In practice, we optimize the single sample Monte Carlo estimate of this expectation:

$$\log p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{u}_{1:T-1}, \theta) \tag{25}$$

$$= \log p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}, \theta) p(\mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{u}_{1:T-1}, \theta) \tag{26}$$

$$= \log \prod_{t=1}^{T} p(\mathbf{z}_t|\mathbf{x}_t, \theta) \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \theta) \tag{27}$$

$$= \sum_{t=1}^{T} \log p(\mathbf{z}_t|\mathbf{x}_t, \theta) + \sum_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \theta) \tag{28}$$

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 Ablation Studies

Table 3: Test RMSE (lower is better) of prediction results with four ablation settings on *T-20-Rate-5*, *T-15-Rate-10*, and *T-10-Rate-20*.

T-20-Rate-5	MLP	EKFN-M	EKFN-I
w.-env-w.-bev	0.672	0.629	0.648
w.-env-w.o.-bev	0.682	0.637	0.671
w.o.-env-w.-bev	0.688	0.633	0.637
w.o.-env-w.o.-bev	0.711	0.636	0.680
T-15-Rate-10	MLP	EKFN-M	EKFN-I
w.-env-w.-bev	1.255	1.231	1.182
w.-env-w.o.-bev	1.268	1.247	1.218
w.o.-env-w.-bev	1.258	1.235	1.194
w.o.-env-w.o.-bev	1.274	1.255	1.231
T-10-Rate-20	MLP	EKFN-M	EKFN-I
w.-env-w.-bev	1.743	1.636	1.810
w.-env-w.o.-bev	1.800	1.711	1.882
w.o.-env-w.-bev	1.754	1.667	1.830
w.o.-env-w.o.-bev	1.802	1.709	1.879

We further conduct ablation studies to discover the effects of environmental factors and driver behavior factors in the prediction tasks. We compare the performance of our model given four types of data: with or without environmental information and with or without driver behavior information. We denote these four settings as *w.-env-w.-bev*, *w.-env-w.o.-bev*, *w.o.-env-w.-bev*, and *w.o.-env-w.o.-bev* where *env*, *bev*, *w.* *w.o.* are short for environment, driver behavior, with, and without, respectively. Note that the observation data from the battery itself remains the same in these settings.

Table 3 shows prediction results with the four settings on the generated *T-20-Rate-5*, *T-15-Rate-10*, and *T-10-Rate-20* dataset from the *RWHDR* dataset. We compare with the MLP model as a baseline, which shows our model still outperforms it on all settings. Comparing four data settings in one dataset, models perform the best on the dataset in *w.-env-w.-bev* setting, which shows that both environmental and behavioral types of data have contributed to the prediction task. Moreover, models performs better in *w.o.-env-w.-bev* setting than *w.-env-w.o.-bev* setting. Models have the highest error on the dataset with no action data. Error decreases more by adding driver behavior data than adding environment behavior data. Driver behavior information offers more important information for SoC consumption than environmental information.

C.2 More Inpterpretation Cases

We show more selected intepretation cases in Figure 3.

D EXPERIMENT DETAILS

D.1 Detailed Description of The Datasets

Vehicle Energy Synthetic Data (VESD). The VESD dataset is a synthetic EV operating dataset based on a real physical model [27], containing the records of battery parameters including battery voltages, internal resistance, and capacity. We consider the effects of environmental and behavioral factors on the road in data synthesizing [21], e.g., road friction factor, aerodynamic factor, and acceleration rate. We generate 100 battery profiles with different battery initialization states and collect 3000 data points of length $T = 10$.

We generate a set of EV operating data based on a real physical model from [27], containing the records of battery parameters, including battery voltages, internal resistance, and capacity. The model of the battery is described by the relationships between currents and voltages measured inside the battery. This model can be described as follows:

- The circuit battery voltage V_t which is composed of an average equilibrium potential V_e, a hysteresis voltage V_h, and other voltages on resistances and capacitances;
- Internal resistance R_i consisting of the Ohmic resistance R_0 and the polarization resistances, R_{pa} and R_{pc}. R_{pa} represents effective resistance characterizing activation polarization and R_{pc} represents the effective resistance characterizing concentration polarization;
- effective capacitances C_{pa} and C_{pc}, which are used to describe the activation polarization and concentration polarization, and used to characterize the transient response of the battery.

The discrete system for the circuit model of a battery could be expressed as follows. Timestamp subscripts are added to indicate the time-dependent variables.

$$\begin{aligned}
V_{pa}^k &= V_{pa}^{k-1}\exp(-\Delta t/R_{pa}C_{pa}) \\
&\quad + IR_{pa}(1-\exp(-\Delta/R_{pa}C_{pa})) \\
V_{pc}^k &= V_{pc}^{k-1}\exp(-\Delta t/R_{pc}C_{pc}) \\
&\quad + IR_{pc}(1-\exp(-\Delta/R_{pc}C_{pc})) \\
V_h^k &= V_h^{k-1}\exp(-\alpha\Delta t) - (1-\exp(-\alpha\Delta t))V_{h,\max} \\
V_t^k &= V_e(s^k) - IR_0 - V_{pa}^k - V_{pc}^k + V_h^k \\
s^k &= s^{k-1} - \eta_i(I)\Delta t/C_N
\end{aligned}$$

where s^k represents the SoC value, I denotes the current through the battery, $V_{h,\max}$ represents the maximum hysteresis voltage of the battery, α is the hysteresis coefficient, and η_i stands for columbic efficiency. Note that we fix the $\{R_{pc}, R_{pa}, C_{pa}, C_{pc}\}$ which describes the characteristics of the battey as constants taken from the paper and set current $I = 30A$. We generate these state variables in time interval $\Delta t = 0.1s$. the equilibrium potential V_e is a function of SOC which is determined as $V_e = 5.2s^4 - 0.86s^3 - 12s^2 + 15s + 59$.

We also consider the effects of environmental and behavioral factors on the road in the data synthesizing processes, such as road friction factor, aerodynamic factor, acceleration rate, and others [21]. In this paper, the energy cost of the EV is modeled by the power demand:

$$s^k = s^{k-1} - P_{eng}\Delta t \tag{29}$$

$$P_{eng} = F_{eng} * v \tag{30}$$

$$F_{eng} = F_{acc} + F_f + F_{air} + F_g \tag{31}$$

where The engine power demand, P_{eng}, is a function of velocity and acceleration, F_{eng} is the force produced by the engine required to overcome forces acting on the vehicle at a given speed v.

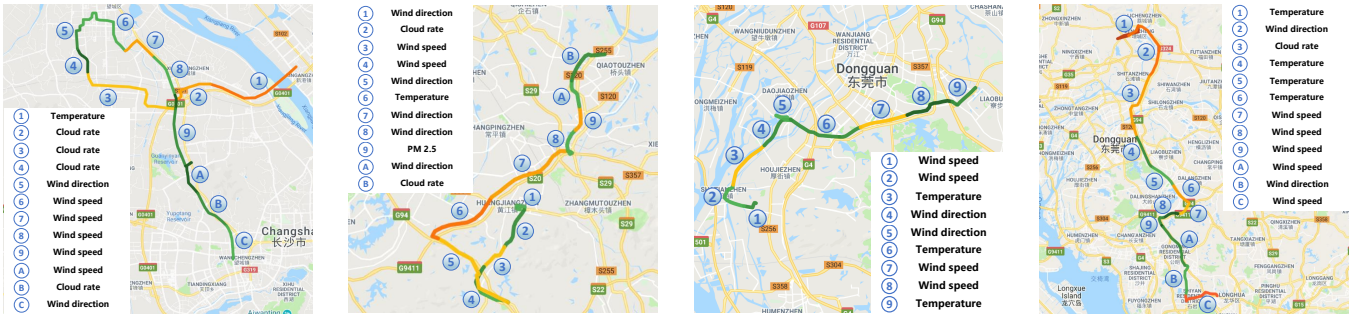

Figure 3: Four sampled driving trajectories. Driving trajectories in various colors indicate the expected SoC drop in a fixed time length. Dark red indicates a severe SoC drop and light green indicates a moderate SoC drop. Environmental factors affecting SoC change most significantly at every time step are labeled to the corresponding trajectory.

$\{F_{acc}, F_f, F_{air}, F_g\}$ are the forces decomposed into components due to acceleration, friction, air resistance and gravitation. The forces are determined by the environment and driver operations described as follows:

$$F_{acc} = ma \tag{32}$$

$$F_f = c_{rr} mg \tag{33}$$

$$F_{air} = \frac{1}{2} c_d A_f \rho v^2 \tag{34}$$

$$F_g = mg \sin(\theta) \tag{35}$$

where $\{m, g, A_f, \}$ represents vehicle mass, gravity coefficient, and frontal area of the car which are considered as constant, $\{v, a, c_{rr}, c_d, \rho, \theta\}$ denotes the velocity, the acceleration rate of the vehicle, rolling friction coefficient, aerodynamic drag coefficient, air density and the slope of the road, which are taken as environmental and driver-specific factors in the EV dynamic process. These factors are sampled from a reasonable normal distribution at each time step to affect the SoC value.

The overall SoC is affected by the battery model and these factors together: $s^k = s^{k-1} - \eta_i(I)\Delta t / C_N - -\lambda P_{eng} \Delta t$ where λ is the ratio of contribution of other factors outside the battery. We generate 100 times battery profiles with different battery initialization state and collect 3000 datapoints of length $T = 10$.

Real-World Human Driving Records (RWHDR) . The RWHDR dataset is a real-world, large-scale human driving dataset of EVs collected from a transportation company. This dataset contains 33311 EV driving records of different drivers in over 50 cities. Each EV driving record is multidimensional time series data collected in a frequency of 0.1Hz (sample once every 10 seconds) with recording time length varying from 100 seconds to over 8000 seconds. We choose 32940 records that are recorded for at least 1000 seconds. All data are anonymized, desensitized, and aggregated. After attaching corresponding weather information, vehicle information, and driver information, each recording contains data of four major sources including vehicle sensor data, driving control signals (such as speed), driver information (such as driver age), and weather information (such as temperature). Furthermore, All data is collected in the form of **GB/T-32960** international standard. To test model prediction ability in various time lengths, we generate three different sets of driving track data of different sampling rates. Three sets of training data have sampling rates of 50s, 100s, and 200s. Corresponding time lengths of datapoints in each track data are $T = 20, 15$, and 10 (lasting 1000s, 1500s, and 2000s), respectively. We denote these three generated datasets as *T-20-Rate-5, T-15-Rate-10*, and *T-10-Rate-20* where the first number indicates the number of time steps in one track datapoint and the second number indicates the sampling rates of samples. In these datasets, we notice that there are 68% sharp changes of delta SoC from 0 to 1 at one time step in *T-20-Rate-5*, 50% sharp changes in *T-15-Rate-10*, and 22% in *T-10-Rate-20*. Delta SoC variances is 0.7159 in *T-20-Rate-5*, 1.4001 in *T-15-Rate-10*, and 3.1331 in *T-10-Rate-20*.

We separate data into three parts for our model implementation: (1) State Data: state data contains continuous feature data from vehicle sensor and is used as observation in model implementation. Sensor data includes motor and motor controller temperature, motor voltage, current, resistance, motor npm and rmp, etc. (2) Action Data: action data contains continuous feature data from environment and driver behavior. Environment information involves humidity, cloud rate, temperature, wind direction, wind speed, and pm2.5. Driver behavior data includes driver control information (speed, brake, and acceleration) and driver personal information (driver age, license year, and driver gender). (3) Delta SoC: delta SoC is SoC (State of Charge) change at each time step.

D.2 Implementation Details

We implement two kinds of models for our proposed model: one with emission function parameterized by a neural network, EKFN-M, and one with identical emission, EKFN-I. We parameterize the transition network by a 3-layer MLP with `ReLU` activations on the hidden layer and linear activations on the output layer for both implementations. For the emission network of EKFN, we parameterize it by a 3-layer MLP with the same structure as the transition network. Moreover, we use `RandomForestRegressor` from `sklearn` [2] in python for random forest implementation. We use `stats-toolbox` [23] in python for the VAR implementation. For Kalman filter implementation, we use `pykalman` [3] in python. For the neural network model, we use 3-layer MLP with the same structure as the transition network. For LSTM and PLSTM model, we use one layer with 128 neurons to model the time-series and then apply a soft-max regressor on top of the last hidden state to do regression.

As for DMM, we use the same network structure proposed in [15]. To perform the prediction task mentioned above, we use the final outputs of predicted state data and given action data to get the next predicted state data for regression models. For generative models, we use a prediction method that the models already have to make the prediction. To ensure a fair comparison, we use roughly the same amount of parameters for all models. For experiments on the described dataset, train/valid/test sets were split as 60/10/30, and we report the average Mean Squared Error (MSE) on the test dataset. Note that we train all the deep learning models with the Adam optimization method [12] and use a validation set to find the best weights and report the results on the held-out test set. All the input variables are normalized to be of 0 mean and 1 standard deviation.