Learning the State-of-Charge of Heterogeneous Fleets of Distributed Energy Resources with Temporal Residual Networks

Mustafa Matar[a], Hani Mavalizadeh[a], Sarnaduti Brahma[a], Mads R. Almassalkhi[a], Safwan Wshah[b]

[a] *Department of Electrical and Biomedical Engineering, The University of Vermont,*
[b] *Department of Computer Science, The University of Vermont,*

Abstract

With increased use of renewable energy such as wind and solar, electric power generation is experiencing increased variability and uncertainty, which drives larger imbalances between the electric demand and supply. To mitigate this challenge, one can use distributed energy resources to beget flexible demand from coordinating fleets of smart electric water heaters (EWH) and residential (kW-scale) batteries. To effectively coordinate and characterize such a large and heterogeneous fleet of distributed energy resources (DERs), a common abstraction is denoted a virtual battery (VB). While the state of charge (SoC) of individual DERs (e.g., EWHs's water temperature) can be easily measured, determining the SoC of a controlled virtual battery aggregation is a technically challenging task due to the fleet's heterogeneous nature, characterized by nonlinear, stochastic, partial differential equations with time-varying parameters. In this paper, a data-driven approach is presented that utilizes a deep-learning-based Temporal Residual Causal Network to determine the SoC for a heterogeneous fleet of DERs, updated using only available end-use measurements. Unlike existing literature that generally relies on complex physics-based models, our deep learning (DL) model is trained using practical input-output data. The simulation results demonstrate that accurate estimation can be achieved with a low computational burden, considering a range of parametric variations at the device and fleet levels, such as fleet population size, background demand, DER device parameters, and coordinator communication losses. The results suggest that the proposed approach has appropriate generalization and robustness properties for practical, real-time control settings.

Keywords: State-of-charge estimation, virtual battery, Packetized Energy Management, distributed energy resources, thermostatically controlled loads, deep learning, temporal residual causal network.

1. Introduction

With the increasing penetration of renewable generation like solar photovoltaic (PV), there is added uncertainty and variability in the supply that requires careful consideration to avoid compromising reliable operations of the electric power grid [1]. Recently, many schemes for coordinating distributed energy resources (DERs), e.g. thermostatically controlled loads (TCLs) like air conditioners and water heaters, electric vehicles (EVs), and energy storage systems (ESSs), have been proposed to provide a range of ancillary services, such as primary frequency control [2] and frequency regulation [3]. In [4], an frequency regulation method is proposed that minimizes the impact of PV uncertainty. Resources such as TCLs also represent a source of stored (thermal) energy that is flexible (i.e., it can defer energy consumption/supply without impacting the quality of service (QoS) for end-users). [5] proposes a method to quantify the aggregate flexibility for a fleet of TCLs. When aggregated and coordinated, thousands of TCLs can function as a single battery-like resource or grid asset, known as a virtual battery (VB) model [6]. Such a VB model can be a useful abstraction of a (large) collection of DERs for the purpose of effectively dispatching DERs *en masse*. A VB model typically includes a scalar state-of-charge (SoC) measured in megawatt-hours (MWh), as well as upper and lower power bounds measured in megawatts (MW) and energy capacity bounds measured in MWh. The SoC represents the average DER energy level of the DER fleet, the power bounds inform the maximum and minimum deviation in power consumption (from some established baseline or an admissible operating range) that can be consumed or provided by the devices, while the capacity bounds provide limits for the SoC. To effectively design coordination and control schemes using the VB model, it is necessary that the VB parameters are accurately estimated or identified.

Different studies have explored the identification of VB models for DERs. For example, in [7], the charge rate limits and capacity parameters are identified for a collection of TCLs, whereas in [6], a detailed model of the load and its control system is used to generate a VB model for a residential heating, ventilation, and air conditioning (HVAC) system. In [8], the VB parameters such as self-dissipation rate, and energy capacity are obtained via simulation using a first-order VB model by repeatedly solving an optimal control problem that minimizes the power tracking error for the aggregate. However, a challenge with these methods is that they assume the availability of full end-use device-specific parameters, which are often unknown in practice.

An alternative to model-based approaches for VB model identification, e.g., [9], is data-driven methods using, for example, machine learning or, specifically, deep learning [10]. Machine learning approaches are used for a wide variety of applications ranging from state of health for Lithium-Ion batteries [11, 12], recommender system [13] to music education [14]. Deep learning involves training a neural network using operational data from DERs to obtain the VB model. Recent works have explored identifying VB parameters using deep learning. For example, in [15], a transfer learning-based stacked autoencoder is used to calculate the virtual battery state of a given ensemble of flexible TCLs from available end-use measurements. In [16], a variational autoencoder-based deep learning algorithm is proposed to identify the probability distribution of the parameters of a stochastic VB model, such as self-dissipation rate, and power and energy capacities. The limitation of with these works is that they assume that the coordinator has direct controllability and full observability of all devices' state information, which is not practical in real-time implementations. Therefore, the aforementioned identification methods are difficult to adapt to practice without incurring high communication overhead, as they need real-time data streaming from all devices to the coordinator. In this paper, we explicitly consider a scenario, where the coordinator can neither directly control devices nor has full access to device state information. Instead, the coordinator uses an indirect scheme, whereby a device asynchronously (i.e., based on its own local clock and need for energy) requests access to the grid, and the coordinator either accepts or denies it. With the asynchronous nature of this implementation, the coordinator receives a set number of requests per second (request rate) and accepts a proportion of these requests (acceptance rate) to change the aggregate power of the fleet. With only this indirect request and acceptance rates from the fleet and the aggregate fleet power (i.e., a few scalar values), the previous literature on VB identification is not applicable. This highlights the need to develop a framework for such an indirect DER coordination scheme.

This work aims to adapt a deep learning methodology to identify the SoC of the VB when the DERs are participating in a specific indirect coordination scheme called Packetized Energy Management (PEM) [17]. In PEM, DERs asynchronously and stochastically send grid access requests to a coordinator, which, if accepted, permits the device to consume or supply power for a pre-specified, fixed epoch, called the packet length. The requests are either accepted or denied by the coordinator in real-time to regulate the aggregate net consumption of the fleet based on a provided power reference signal that can change over time. PEM has been extensively studied and modeled under assumptions on homogeneity in parameters for TCL, EV, and ESS populations [17, 18, 19]. The key distinction of this work is that it is a data-driven methodology for a heterogeneous DER fleet, making it a useful tool in a practical setting where exact parameter values are unknown, and only a distribution of parameter values is known. Moreover, even though the model requires limited data for training, it is a surprisingly generalizable estimate and applies across a range of operating conditions.

In this work, the focus is on the development of a novel machine learning-based framework to accurately and robustly estimate the SoC for a population of *heterogeneous* TCLs coordinated under PEM. The ML framework relies on just the following six scalar time-series data streams that represent the total number of 1) incoming requests, 2) accepted requests, 3) expired packets, and 4) ON devices, 5) the reference power signal, and 6) the aggregate fleet power output. Thus, the main contributions of the work are as follows:

- A novel Deep Learning (DL)-based method is proposed to estimate SoC of a virtual battery composed of a heterogeneous fleet of DERs using Packetized Energy Management (PEM) in real-time with high accuracy and minimal inputs.

- A Temporal Residual Causal Network (TRCN) model is presented which is computationally lightweight and requires training only once under ideal conditions. It does not need to be retrained under significant parameter variation, making it suitable for practical implementation.

- Simulation-based studies validate the proposed DL-based model's effectiveness under various practically relevant fleet compositions and device parameter variations, such as DER background demand, coordinator communication losses, and heterogeneous DER device parameters.

The rest of the paper is organized as follows: Section 2, an overview of Packetized Energy Management is provided. Section 3 discusses the *virtual* *battery* model. Section 4 describes the proposed methodology. The experimental setup is described in section 5. In section 6, the simulation results are conducted using TRCN. Concluding remarks are provided in section 7.

2. Packetized Energy Management

In this section, a brief overview of PEM for diverse DERs [17, 19] is presented. Two types of DERs are used, EWHs and batteries. In PEM, a DER with a local state-of-charge (SoC) $z_n[k]$ (e.g., temperature for an EWH or state of charge for a battery), is designed to operate within a deadband $[\underline{z}_n, \overline{z}_n]$ to maintain a certain level of consumer comfort. The dynamic state for EWH n is given by the following equation:

$$z_n[k+1] = z_n[k] + \Delta t \left(\frac{\eta_n P_{\mathrm{n}}^{\mathrm{rate}}}{c_p \rho L_n} \phi_n[k] - \frac{z_n[k] - T_{\mathrm{a}}[k]}{\tau_n} - \frac{Q_n[k]}{c_p \rho L_n} \right) \tag{1}$$

where $c_p = 4.186$ [kJ/kg-$^\circ C$] is the specific heat constant for water, τ_n is the standing loss time constant to ambient temperature, ρ is the water density close to $50^o C$, L_n is the tank capacity in [Liters], η_n is the efficiency, P_n^{rate} is the power rating in [kW], and $\phi_n[k]$ is a binary variable determining if device n is on or off. T_a is the ambient temperature in [$^\circ C$], Δt is the discretization time-step in [s], and $Q_n[k]$ is the heat loss from the tank due to water usage. The dynamic model of batteries is summarized by the following equation:

$$z_n[k+1] = z_n[k] + \Delta t(-\eta_n^{\mathrm{st}} z_n[k] + \phi_n[k] P_n^{\mathrm{rate}} \eta_n) \tag{2}$$

where $\phi_n[k]$ is +1 if the device is discharging, and is -1 if device n is charging at time k. If the device is in standby mode, $\phi_n[k] = 0$. The efficiencies for standing losses and charging are η_n^{st} and η_n, respectively.

Each DER measures its local SoC, $z_n[k]$. If the SoC is outside the deadband, $z_n[k] \notin [\underline{z}_n, \overline{z}_n]$, the DER automatically and temporarily opts out of PEM to guarantee Quality of Service (QoS) and reverts to a conventional control mode until the SoC is returned within limits after which it returns to PEM operation. If the SoC is within the deadband, $z \in [\underline{z}_n, \overline{z}_n]$, the DER probabilistically requests the PEM coordinator to either consume power from the grid (charging) or inject power into the grid (discharging) for a pre-specified epoch. The epoch corresponding to the energy packet is called packet length and denoted as δ_{p}. The requests are given by the following cumulative distribution function:

$$P_{\mathrm{req}}^{\mathrm{ch}}(z_n[k]) := 1 - e^{-\mu(z_n[k])\Delta t}, \tag{3}$$

where $\mu(z_n[k]) > 0$ is a rate parameter dependent on the local SoC and is defined as,

$$\mu(z_n[k]) \\ = \begin{cases} 0, & \text{if } z_n[k] \geq \overline{z}_n \\ m_R \left(\frac{\overline{z}_n - z_n[k]}{z_n[k] - \underline{z}_n} \right) \cdot \left(\frac{z_n^{\text{set}} - \underline{z}_n}{\overline{z}_n - z_n^{\text{set}}} \right), & \text{if } z_n[k] \in (\underline{z}_n, \overline{z}_n) \\ \infty, & \text{if } z_n[k] \leq \underline{z}_n \end{cases} , \tag{4}$$

where $m_R > 0$ [Hz] is a design parameter that defines the mean time-to-request (MTTR) for $z_n[k] = z_n^{\text{set}}$. A similar expression follows for $\mu(z_n[k])$ and $P_{\text{req}}^{\text{dis}}(z_n[k])$ in the case of *discharging* packets.

The energy packet requests then are sent to a coordinator asynchronously. Each request is either accepted or rejected based on aggregate demand and a market or grid reference signal. When a request for an energy packet is accepted, after δ_{p} seconds, the packet is expired, i.e., the DER stops charging or discharging. Since the coordinator knows how many devices are accepted at each time, it can accurately track the total number of charging $N_{\text{ON}}^{\text{ch}}[k]$ and discharging devices $N_{\text{ON}}^{\text{dis}}[k]$ in real time. The coordinator also receives the number of devices that opt out during each time step, $N_{\text{opt}}[k]$. Figure 1 illustrates the closed-loop feedback system for PEM. Each DER's normalized SoC is defined as $E_n[k] := (z_n[k] - \underline{z}_n)/(\overline{z}_n - \underline{z}_n)$. When $E_n[k] > 1$ or $E_n[k] < 0$ DER n experiences discomfort (e.g., too hot or too cold) and notifies the Aggregator that it opts out of PEM (temporarily). Thus, to preserve customers' quality of service, there is a desire on behalf of the Aggregator to keep the DERs' SoCs away from either extreme. Therefore, at any given time k, it is very valuable to know or estimate the mean of the distribution of SoCs (i.e., a key *state of the fleet*). Clearly, with full DER information available, the average SoC can be simply obtained by averaging over all DERs: $E_{\text{avg}}[k] := \sum_{n=1}^{N} E_n[k]/N$. However, the Aggregator does not have access to the local, individual SoC measurements, $E_n[k]$. Instead, it needs to estimate this average $\tilde{E}_{\text{avg}}[k]$ either using model-based methods or data-driven methods. In the next section, a procedure from [20] is outlined that summarizes the nonlinear and stochastic relationship between Aggregator inputs and outputs and the fleet's average SoC to illustrate the challenge with a physics-based approach to modeling and to motivate the proposed data-driven, learning-based approach.

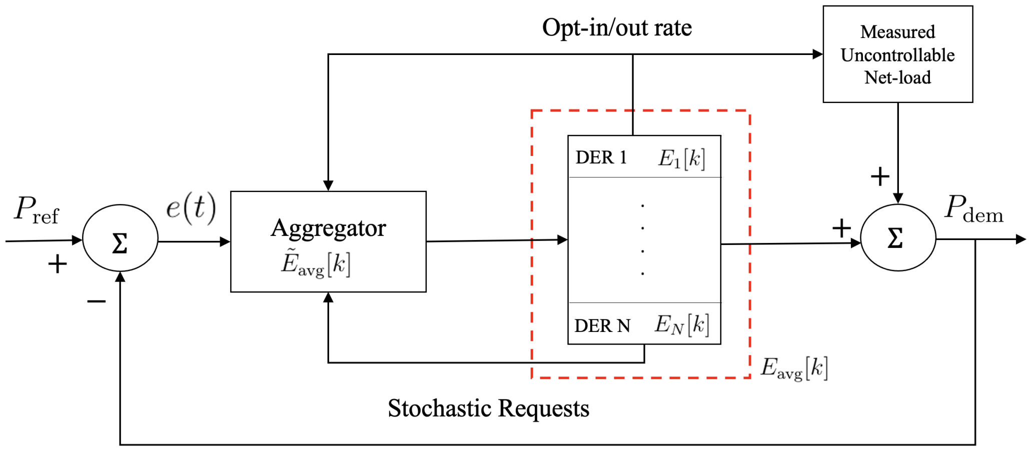


Figure 1: Closed-loop feedback system used for PEM with P_{ref} provided by the grid or market operator and the aggregate net-load P_{dem} measured by the Aggregator. The Aggregator uses the tracking error, opt-in/out rates and request rates to estimate the fleet's average SoC, $\tilde{E}_{\text{avg}}[k]$. The fleet's *actual* average SoC is given by the average of the distribution of the DERs' SoCs, $\{E_n[k]\}_{n=1}^{N}$ (in red dashed box), and denoted $E_{\text{avg}}[k]$.

3. Virtual battery model

In this section, we motivate the proposed data-driven approach to estimate the average energy or SoC of a DER fleet, $\tilde{E}_{\text{avg}}[k]$ by summarizing the related physics-based modeling and estimation approach introduced in [18, 20]. These past results accurately estimate $\tilde{E}_{\text{avg}}[k]$ under specific fleet conditions (i.e., *homogeneous* device and control parameters). using only the aggregate power consumption of the fleet, the incoming charging and discharging packet requests, and total number of devices consuming/injecting power and opting out – all received and tracked by the coordinator. From this estimate, the authors use a physics-based predictive model of a fleet of homogeneous DERs into a so-called PEM *virtual battery* or *PEM-VB*. The PEM-VB is characterized by four salient dynamics states:

- Average energy, $E_{\text{avg}}[k]$.
- Total number of charging DERs, $N_{\text{on}}^{\text{ch}}[k]$.
- Total number of discharging DERs, $N_{\text{on}}^{\text{dis}}[k]$.

- Total number of Opt-outs, $N_{\text{opt}}[k]$.

These states are coupled through the incoming packet requests. For example, the higher $E_{\text{avg}}[k]$ is, the higher the fleet's devices SoCs, which leads to a lower aggregate request rate (e.g., as EWHs heat up, they need less energy and how a lower probability of requesting a packet). Consequently, a lower request rate limits the fleet's ability to ramp up its aggregate power.

Under homogeneous fleet parameter assumptions, one can consider the fleet's average power via Equation 1 and, by assuming a constant average hot water consumption, μ_Q, get the following expression for estimating the average temperature for a EWH fleet, $\tilde{E}_{\text{avg}}[k]$:

$$\tilde{E}_{\text{avg}}[k+1] = (1 - \frac{\Delta t}{\tau})\tilde{E}_{\text{avg}}[k] + \frac{\Delta t T_{\text{a}}}{\tau} - \frac{\Delta t \mu_Q}{c_p \rho L} + \frac{\eta \Delta t P^{\text{rate}}(N_{\text{on}}^{\text{ch}}[k] + N_{\text{opt}}[k])}{c_p \rho L N}, \tag{5}$$

The change in $\tilde{E}_{\text{avg}}$ depends on the number of charging EWHs as well as background demand. T_a is assumed to be constant in this paper. Similar expression can be obtained for a battery fleet using Equation 2. Note that above expression is only valid when the parameters $\tau, L, P^{\text{rate}}, \underline{z}$, and $\overline{z}$ are common across all devices in the fleet (i.e., the homogeneity assumption). Clearly, the average SoC increases the more ON and opt-out devices there are, which are states driven by the rate of accepted requests. The requests only come from devices in standby mode (i.e., not in ON (charging or discharging) and not in opt-out modes) and are driven by $\tilde{E}_{\text{avg}}[k]$ and the request probabilistically in Equation 3. Therefore, the number of charging requests received by the coordinator during the interval k is,

$$x_{\text{r}}^{\text{ch}}[k] = P_{\text{req}}(\tilde{E}_{\text{avg}}[k])(N - N_{\text{on}}^{\text{ch}}[k] - N_{\text{on}}^{\text{dis}}[k] - N_{\text{opt}}[k]) \tag{6}$$

A similar expression can be obtained for the estimated number of discharging requests, $x_{\text{r}}^{\text{dis}}[k]$. Define $\beta_{\text{ch}}[k]$ and $\beta_{\text{dis}}[k]$ as the ratio of accepted charging and discharging requests during interval k, respectively. Then $\beta_{\text{ch}}^{-}[k]$ and $\beta_{\text{dis}}^{-}[k]$ are the proportion of expired charging and discharging requests during interval k, respectively. The dynamics of the number of ON (charging and discharging) devices can be expressed as:

$$N_{\text{on}}^{\text{ch}}[k+1] = N_{\text{on}}^{\text{ch}}[k] + \beta_{\text{ch}}[k] x_{\text{r}}^{\text{ch}}[k] - \beta_{\text{ch}}^{-}[k] N_{\text{on}}^{\text{ch}}[k], \tag{7}$$

$$N_{\text{on}}^{\text{dis}}[k+1] = N_{\text{on}}^{\text{dis}}[k] + \beta_{\text{dis}}[k] x_{\text{r}}^{\text{dis}}[k] - \beta_{\text{dis}}^{-}[k] N_{\text{on}}^{\text{dis}}[k]. \tag{8}$$

From this, it is clear that the number of ON devices increases if the coordinator accepts more (new) device requests than are expiring (i.e., completing their packet).

Finally, to capture the total number of devices opted out at timestep k, we consider the number of devices opting out and the number of devices opting back in as follows:

$$N_{\text{opt}}[k+1] = N_{\text{opt}}[k] + \mathcal{E}_{\text{opt-out}}[k] - \mathcal{E}_{\text{opt-in}}[k] \tag{9}$$

where $\mathcal{E}_{\text{opt-out}}[k]$ and $\mathcal{E}_{\text{opt-in}}[k]$ are the number of opt-outs and opt-ins during timestep k, respectively.

In [20], the model described by Equation 5-Equation 9 is applied to a homogeneous fleet of EWHs. An extended Kalman filter (EKF) was then developed to accurately estimate $\tilde{E}_{\text{avg}}$. However, under heterogeneous conditions and for a collection of mixed DERs (e.g., batteries and EWHs), the model to estimate $\tilde{E}_{\text{avg}}$ is not applicable. One common work-around is to decompose the fleet into different homogeneous groups and then model each group as a VB [3]. However, such approaches rely on the assumption that each device's set of parameters are accurately known, which is impractical [8]. Thus, in the case of a heterogeneous and diverse fleet of DERs, the above methods are not directly applicable, modeling becomes challenging and complex, and no guarantees exist on the observability of $\tilde{E}_{\text{avg}}$. With that in mind, in the next section, a practical data-driven method is presented to accurately estimate $\tilde{E}_{\text{avg}}$ for a heterogeneous and diverse fleet under a variety of operating conditions. In particular, we will show that the data-driven approach is practical, yet has favorable generalization and robustness properties.

4. Methodology

Due to the complexity of data, direct control over DERs is difficult to manage in near real-time as the DER coordinator usually does not have full access to device local measurements. To enable responsive DER control at scale, indirect schemes should be used, which do not offer full observability of states. This work develops an accurate estimation approach for SoC of the VB model for the PEM scheme. The proposed approach addresses the challenge of limited observability of states in the PEM scheme, by using the available end-use measurements to estimate the SoC of the VB.

The proposed data-driven approach requires the following time-series data: the number of charging requests $x_r^{\text{ch}}[k]$, the number of discharging requests $x_r^{\text{dis}}[k]$, the proportion of charging requests accepted by coordinator $\beta_{\text{ch}}[k]$, the proportion of discharging requests accepted by coordinator $\beta_{\text{dis}}[k]$, proportion of devices that finish charging packets $\beta_{\text{ch}}^{-}[k]$, proportion of devices that finish discharging packets $\beta_{\text{dis}}^{-}[k]$, the total number of DERs in charge mode $N_{\text{on}}^{\text{ch}}[k]$, the total number of DERs in discharge mode $N_{\text{on}}^{\text{dis}}[k]$, number of opt-outs $N_{\text{opt}}[k]$, and power reference signal that PEM is tracking. It uses 240 samples at 2-second resolution to estimate the SoC of the VB model, with a Moving Average (MA) filter having a window size of 30 seconds and a sampling rate of 2 seconds.

The deep learning approach estimates the average SoC of the VB model for the PEM scheme, which is an effective tool for complicated computing tasks. Recent advances in deep learning have led to impressive successes in a wide range of energy applications, such as model parameter calibration, forced oscillation localization, detection of GPS spoofing attacks on PMU and electricity demand forecasting [21, 22, 23, 24].

This research investigated both residual and classical deep learning models for the estimation of SoC of the VB. The proposed DL-model establishes a mapping between VB coordinator information such as the number of accepted requests, number expiring packets and so on, and the SoC of the VB model. The investigated deep learning models are described below. The purpose of the investigation is to compare the performance and robustness of these models in estimating the SoC of the VB and select the best-performing model for the PEM scheme.

Data-driven approaches may generate results that contain noise or outliers, which can affect the PEM. Therefore, it is common practice to apply filtering techniques to reveal the underlying trend. In this work, a MA filter was used, which works by taking a window of data and averaging it to produce a single output. The advantage of using the MA filter is that it preserves the causality of the data, meaning that the filtered output at a particular time point is only dependent on the input data up to that time point. Overall, the use of filtering techniques like MA filter is essential in data-driven approaches to improve the reliability of the estimated results, allowing for better decision-making.

4.1. Baseline Approaches

The performance of the proposed approach was compared with the baseline benchmark approaches such as Convolutional Neural Networks (CNNs), and feed-forward Neural Networks (FFNNs). DL-model should be accurate and precise to ensure reliable operation of the electric power grid. The comparison was done by evaluating the performance of the approaches using various metrics as explained in Section 4.4.

4.1.1. Feed-Forward Neural Networks

Feed-forward Neural networks [25] are relatively straightforward and widely used structures in deep learning. They are called feed-forward because the information flows through the network in one direction, from the input layer to the output layer, without looping back. However, FFNN faces many issues such as loss of neighborhood information, over-fitting the data, and vanishing and exploding gradient problems, which refer to the gradients of the error function becoming too small or too large during the training process, which can slow down or prevent the model from converging [26]. Additionally, FFNNs can have a large number of parameters to optimize, which can make the training process computationally expensive and time-consuming.

4.1.2. Convolutional Neural Networks

In recent years, the CNN-based approach has been widely utilized in various fields, including image and signal processing. A typical CNN model consists of multiple layers, including convolutional, non-linearity, pooling, and fully connected layers [27]. However, when dealing with long input measurements, such as those with a high sampling rate, the CNN models may require large kernels to increase the receptive field, which can make the optimization of model parameters challenging. One of the major obstacles in training deep CNNs is the vanishing gradient problem. This can impede the convergence of the CNN model. To mitigate this problem, several solutions have been proposed, such as batch normalization, dropout, and residual connections have been proven to be effective in addressing the vanishing gradient problem and improving the performance of CNN models [28].

4.2. Temporal Causal Convolutional Networks

The architecture of the proposed TRCNs for SoC estimation is illustrated in Figure 2. Key features of TRCNs is the utilization of dilated convolution operations and residual connections [26, 29], which solves the limitations of classical deep learning approaches for SoC estimation, and allows for an exponential increase in the receptive field with the number of layers, thus enabling the model to achieve a large receptive field with a relatively small number of layers. Additionally, these feature enable TRCNs to effectively handle

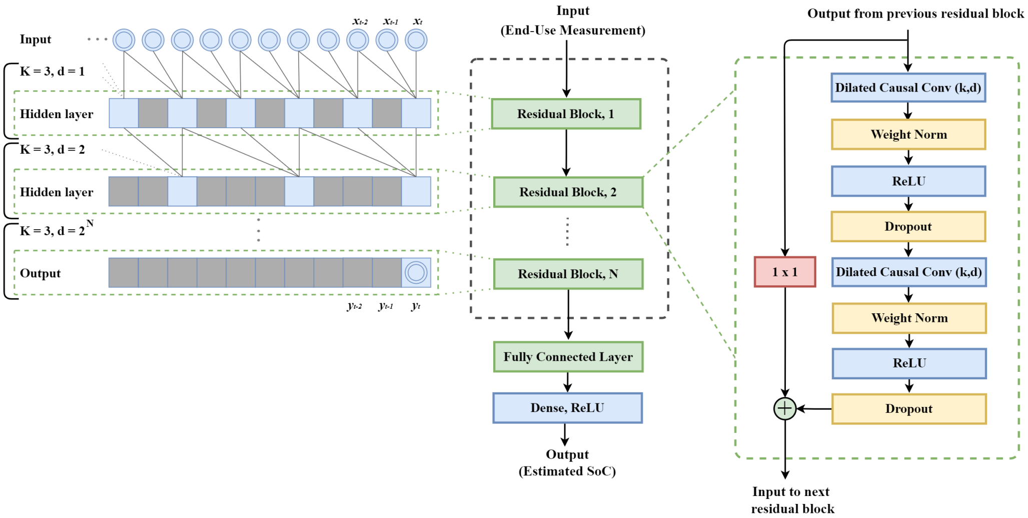


Figure 2: The network architecture of TRCN-based model.

long-range temporal dependencies , which is crucial for accurate prediction of SoC.

To process the input sequence, a series of residual blocks are employed, incorporating dilated convolution operations and residual connections, with increasing dilation rates. Additionally, Weight Normalization is utilized to normalize the weights of the neural network layer. This technique enhances the convergence of the network during training by decreasing the internal covariate shift. Furthermore, it reduces the network's sensitivity to the initial weights, resulting in a more stable optimization process. To avoid overfitting and enhance the neural network's generalization performance, Dropout, a deep learning regularization technique, is applied. During training, dropout randomly drops out (i.e., sets to zero) some of the neurons in a layer. The final step involves using a fully connected layer to perform regression and obtain the current state of charge.

TRCNs have a backpropagation path that different from the temporal direction of the sequence. The input data bypass the convolution operation through skip connections, and the outputs of both the skip connection and the convolution operation are then added together to form the output of the entire block. This mechanism allows the gradients to flow more efficiently through the network, thereby improving the overall performance of the model.

When designing a TRCN for online application, a key consideration is the principle of causality, which implies that the output of the network y_{T} at any given time step T should only depend on the convolutions of elements of the input sequence $x_0, ..., x_{\mathrm{T}}$. In this way, only the previous history of the measurements is used to predict the label at anytime T. That's meaning there is no information "leakage" from the future to the past.

$$(y_0, ..., y_{\mathrm{T}}) = f(x_0, ..., x_{\mathrm{T}}) \tag{10}$$

So, TRCNs satisfies the causal constraint that y_{T} depends only on $x_0, ..., x_{\mathrm{T}}$ and not on any future inputs, making the TRCNs suitable for real-time execution. In the sequence modeling, the purpose of learning is to discover a network that minimizes some expected loss between actual outputs and predictions.

The utilization of a causal convolutional structure is necessary to prevent future information leakage. However, applying a causal convolution directly to deal with long time series problems poses a significant challenge, as it limits the ability of the network to look back at historical data, with a linear size in the depth of the network [30]. To overcome this limitation, the proposed approach employs dilated convolutions that enable an exponentially large receptive field without increasing the model parameters [31]. Unlike standard convolution, where each kernel covers the same range as its length, dilated convolution's kernel skips input samples with a fixed step to cover a longer range, as shown in Figure 2. The receptive field size of the TRCN is dependent on the network depth N, filter size K and dilation factor d, thus making the TRCN deeper and larger is crucial to obtain sufficiently large receptive field.

4.3. Hyper-Parameters Optimization

In order to achieve optimal performance for any deep learning-based model, the selection of appropriate hyper-parameters is crucial, and several hyper-parameters were considered to tune. Random Search [32] was used to find the network's optimal hyper-parameter combinations in this study. The optimal hyper-parameters for optimizer type, learning rate, filter size, and number of residual blocks were found to Adam [33], 2×10^{-5}, 3, and 5 respectively. The experiments were conducted using Python 3.8 and the Tensorflow v2.8.0.

Table 1: DER fleet parameters

Parameter	Distribution
Power rating for EWHs and batteries (kW)	$P_{\text{rate,n}} \sim \mathcal{U}(3.2, 4.8)$
Tank size for EWHs	$L_n \sim U(240, 360)$
Battery efficiency	$\eta_n \sim \mathcal{N}(0.95, 0.03)$
Battery capacity (kWh)	$\overline{z}_n \sim \mathcal{N}(13.5, 2)$

4.4. Evaluation Criteria

In this work, Root Mean Square Error (RMSE), and mean absolute error (MAE) were utilized as metrics to measure the difference between the estimated and actually observed SoC value. These metric are defined as follows:

$$RMSE = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\Big(E_{\text{avg}}[k] - \tilde{E}_{\text{avg}}[k]\Big)^2} \tag{11a}$$

$$MAE = \frac{1}{K}\sum_{k=1}^{K}\Big|E_{\text{avg}}[k] - \tilde{E}_{\text{avg}}[k]\Big| \tag{11b}$$

where K is the total number of samples and $E_{\text{avg}}[k]$ and $\tilde{E}_{\text{avg}}[k]$ are the actual and estimated SoC at time k. The RMSE indicates the robustness of the estimation while the MAE indicates the accuracy of the estimation [34].

5. Experimental Setup

The following experiments are conducted to verify and evaluate the performance, generalization capability and robustness of the proposed approach. Two different fleet are used to verify the results. *i* heterogeneous fleet of 500 EWHs. *ii* heterogeneous fleet of 250 EWHs and 250 batteries. The fleet is tracking a scaled power reference signal with nominal power (P^{nom} = 200KW). The packet length, δ_p is 3 minutes for all packets while $\mu = 3$ minutes. The Parameters for the DER fleet are presented in Table 1.

The estimation of SoC based on deep learning can be divided into two distinct phases: model training and model testing. In the model training phase, several experiments were conducted in a PEM environment under ideal conditions i.e, no parametric variations. This was done to generate a three-day training data set. A 70-30% split was employed, 70% of the available training data were chosen for training and 30% for validation. The main computational load demanded by these DL-models happens during training phase, making it feasible for implementation for online applications. In the testing phase, testing data were generated under various demand profiles, and various parametric variations. A brief description of these experiments is provided in the following subsections.

5.1. Data-Driven Models Comparison

The first set of experiments focuses on identifying the optimal DL architecture for addressing the SoC estimation problem. This is accomplished by employing a consistent evaluation methodology to compare the performance of various DL-models. In this experiment, a comparison study was conducted between TRCNs, as described in subsection 4.2, and baseline DL-models, namely FFNNs, and CNNs. To ensure consistency, the models were trained using the same data and underwent the same validation and testing procedures.

5.2. Robustness Against Different Parametric Variations

In order to assess the the robustness and generalization capability of proposed DL-based model, the model was tested under a variety of conditions in this experiment. The model was evaluated under ideal conditions with unseen power reference signal, as well as under practical conditions that simulate real-world scenarios [35, 36]. These scenarios include varying population sizes, background demands, device parameters, and coordinator communication loss rates. This allows for an assessment of the model's ability to adapt and perform well in different scenarios.

5.3. Influence of Training Data Size

In order to investigate the impact of training data size on the performance of SoC estimation, several experiments were conducted. The same deep learning architecture and testing sets were used to ensure a fair comparison. The goal of this study was to examine how the size of the training dataset affect the performance of the proposed deep learning-based model, providing insight into the relationship between training dataset and model performance.

5.4. Robustness against diverse population

In order to evaluate the robustness of the proposed approach under a diverse fleet of devices , such as water heaters (EWHs) and energy storage systems (ESSs), this experiment was conducted. A VB was used as an energy-based aggregate model of a heterogeneous fleet of 250 EWHs and 250 ESSs. The fleet was designed to track a scaled power reference signal.

To address the complexity of the heterogeneous diverse fleet, the proposed deep learning model was restructured to incorporate additional end-use measurements such as $x_r^{\text{dis}}[k]$, $\beta_{\text{dis}}[k]$, $\beta_{\text{dis}}^{-}[k]$, and $N_{\text{on}}^{\text{dis}}[k]$ using 240 samples at 2-second resolution to estimate the SoC of the VB model. The model was evaluated in this experiment with unseen AGC signals, providing insight into the model's ability to handle diverse and complex fleets of devices.

6. Numerical Results and Discussion

This section presents the numerical results obtained from the experiments described in section 5, all experiments reported in this section used the proposed TRCN model.

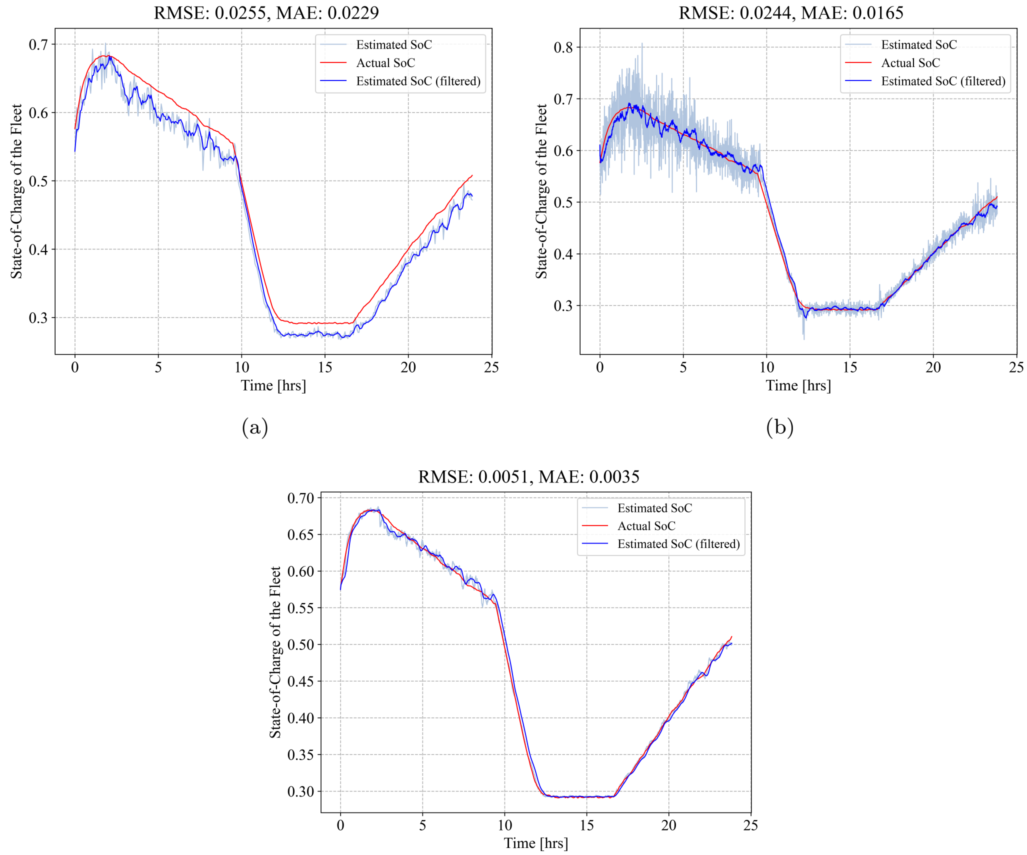
6.1. Data-Driven models comparison

Accurate and precise are two terms that are commonly used in the field of data analysis, machine learning, and statistical modeling. Accurate refers to how close the estimated values are to the actual values. In other words, if the estimated values are close to the actual values, then the model is said to be accurate. Precision, on the other hand, refers to the consistency of the estimated values. If the estimated values are close to each other, then the model is said to be precise, even if it is not necessarily close to the true value.

In order to determine the optimal DL architecture for the SoC estimation, the performance of the proposed deep network was evaluated against baseline DL-models, namely FFNNs, and CNNs. The results showed that while the FFNN approach yielded accurate estimation, it lacked precision. On the other hand, the CNN approach produces precise estimation, but they were inaccurate, as shown in the comparison results presented in Table 2 and Figure 3. Thus, both the FFNNs and the CNNs models were not considered for the implementation. The proposed approach demonstrated an accurate and precise performance, which means that the estimated values are close to the actual values. In addition, the results are precise which means that the estimated error values are close to each other. Therefore, the proposed estimator is able to capture the underlying dynamics effectively.

Table 2: Performance Comparison of CNN, FFNN, and TRCN Models

	RMSE	MAE	Accurate	Precise
CNN	0.0255	0.0229		✓
FFNN	0.0244	0.0165	✓	
TRCN (Proposed)	0.0051	0.0035	✓	✓



(a)

(b)

(c)

Figure 3: Results of SoC estimation under same testing scenario using: (a) CNNs; (b) FFNNs; (c) TCNs.

6.2. Robustness against parameter variation

This section presents different parametric variations tests which used to evaluate the SoC estimation robustness and generalization capability.

6.2.1. Population Size

The proposed approach was trained on a heterogeneous fleet of 500 EWHs. However, in practical scenarios, the population size may change during the operation, making it essential for the model to be robust model to population size variation. Therefore, N in (6) is set to 500 for training, and then the population size (N) is varied between -30% to $+50\%$. Which is a wide enough to test the proposed approach's ability to handle significant changes in population size. It covers a decrease of almost a third of the original population size and an increase of half the original population size.

The results indicate that the proposed model is generalizable and able to adapt to different population sizes without retraining the network. As the number of population diverges from the number of population the model was trained on, the estimated SoCs start to deviate from the true values as shown in Figure 5.

The estimation RMSE and MAE are both less than 1.5% for $[-10\%, +30\%]$ percentage of the change in population size, indicating that the trained network shows the ability to adapt untrained number of population. Figure 4 illustrates the results of SoC estimation under population of 550 heterogeneous devices, further demonstrating the model's ability to adapt to diverse population sizes.

6.2.2. Background Demand

Due to the unpredictability of the human activities such as taking shower, etc, the background water usage can not be predicted accurately and can diverge from the predicted values. In this subsection, different experiments are conducted to evaluate the model's robustness against parametric variation in the average water usage. Same distribution i.e., same CDF is used for determining the starting time of the events, but the average number of events per hour is changed between $[-45\%, +60\%]$.

The results indicate that the proposed model is robust to background demand variation as shown in Figure 6. The estimation RMSE and MAE are both less than 1.5% within $[-25\%, +30\%]$ of the trained mean background demand. Figure 7 illustrates the results of SoC estimation for one day with 15% increase of background demand, and background demand for the case.

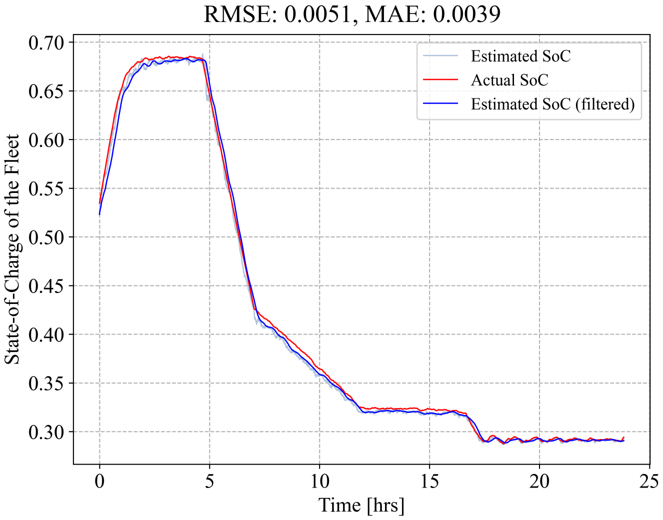


Figure 4: VB SoC estimation under population of 550 heterogeneous devices.

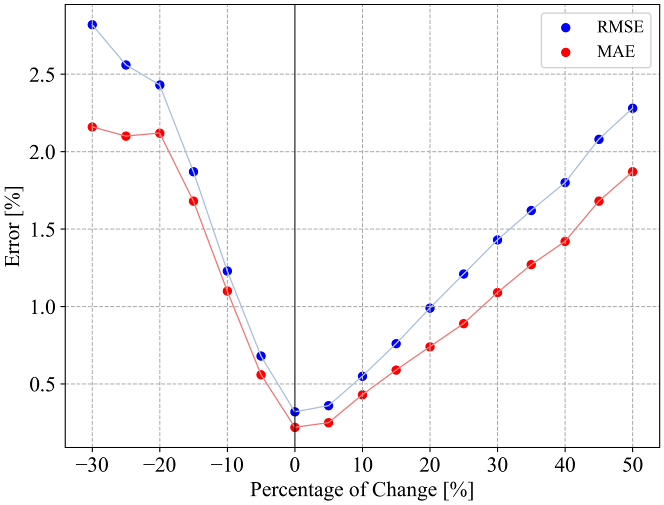


Figure 5: VB SoC estimation error vs percentage change in DER population.

6.2.3. Device Parameters

To further evaluate the robustness of the SoC estimator against variation in DER parameters, two experiments were conducted; i) parametric variation in tank size and ii) parametric variation in power rating of DERs.

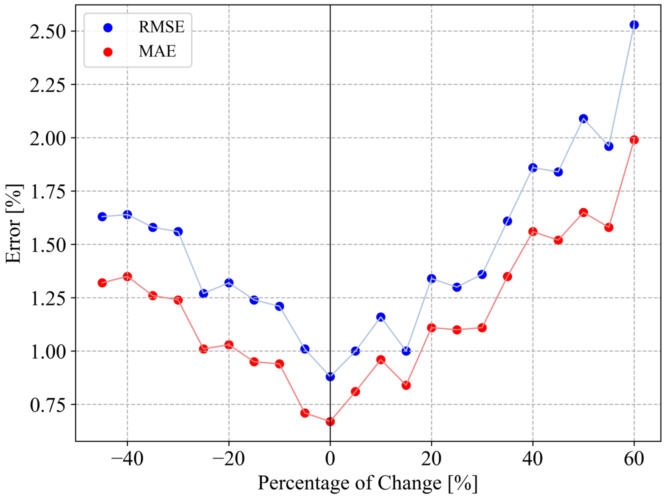


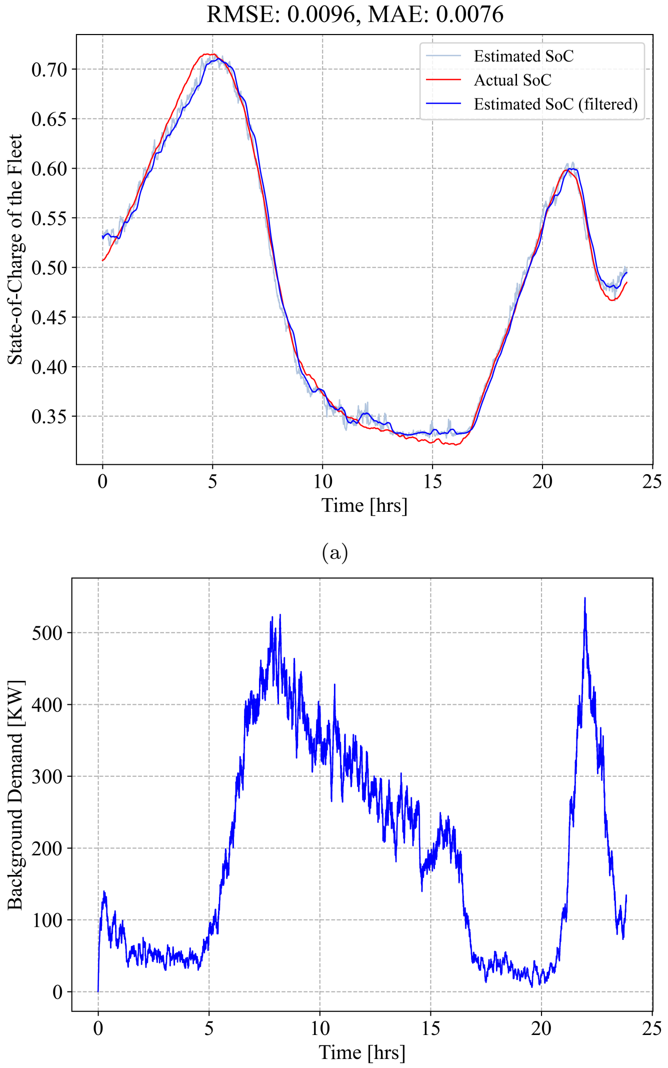
Figure 6: VB SoC estimation error vs percentage change in background demand.

To examine the robustness of the proposed method against variation in tank size, the estimator is trained on a set of random parameters $s \in S$, then the mean tank size is changed between -50% to $+80\%$, while other parameters are same as S. The result of $+20\%$ change is shown in Figure 8. It can be seen that RMSE and MAE are both less than 1.5% for tested range of average tank size. This shows the robustness of the proposed method against estimation error in tank size as shown in Figure 9.

Next, the variation in power rating is taken into account by using S parameters but this time changing the average value of DER power rating. In the training set, $P_{\text{rate}} \sim U(3.2, 4.8)kW$. In the testing set, $P_{\text{rate}} \sim U(4(1+a) - 0.8, 4(1+a) + 0.8)kW$, where $a \in [-30\%, +60\%]$. In fact, the distribution of P_{rate} is shifting with factor a, to study the impact of inaccuracy in estimating P_{rate} on our SoC estimator. The result of $+10\%$ change is shown in Figure 10. It can be seen in Figure 11 that the RMSE and MAE both are less than 1.5% for a wide range of -25% to $+40\%$ error in P_{rate}.

6.2.4. Communication errors

One of the significant practical issues in DER coordination are the communication errors. Communication error can happen in both directions, i.e., *i*. lost packets which means that some of the requests sent by DERs are not received by the coordinator or *ii*. Some of the decisions made by the coordinator are not received by DERs. In this subsection, the goal is to



(a)

(b)

Figure 7: Results in case of background demand variation: (a)VB SoC estimation; (b) Background demand profile.

demonstrate the robustness of the proposed method against communication errors. The results for lost 10% of packets and lost 10% of decisions are presented in Figure 12 and Figure 13, respectively.

In both scenarios as shown Figure 14 and Figure 15, the performance of

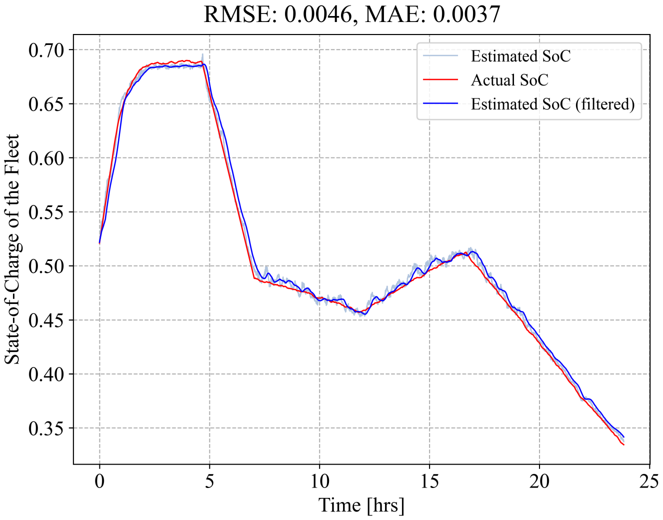


Figure 8: VB SoC estimation under a 20% increase in EWH tank size.

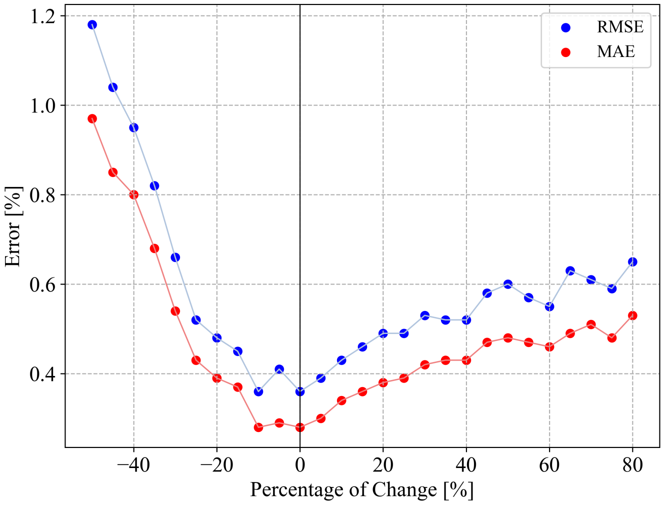


Figure 9: VB SoC estimation vs percentage change in EWH tank size.

the SoC estimator in terms of RMSE and MAE are acceptable up to 10% of requests/commands are lost. Table 3 shows a summary of above experiments.

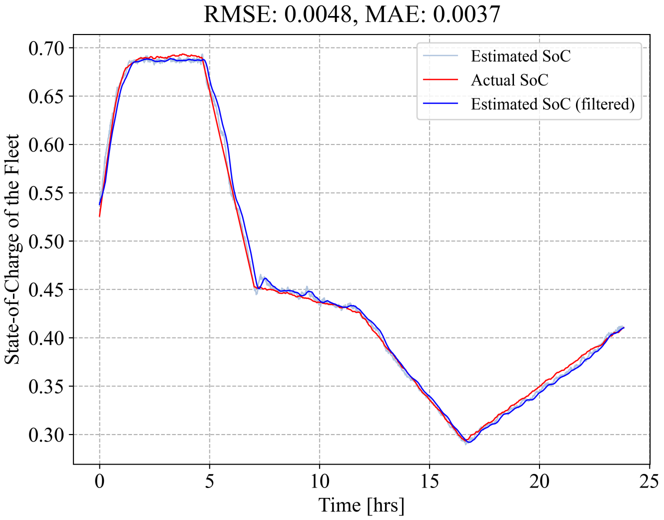


Figure 10: VB SoC estimation under a 10% increase in EWH power rating.

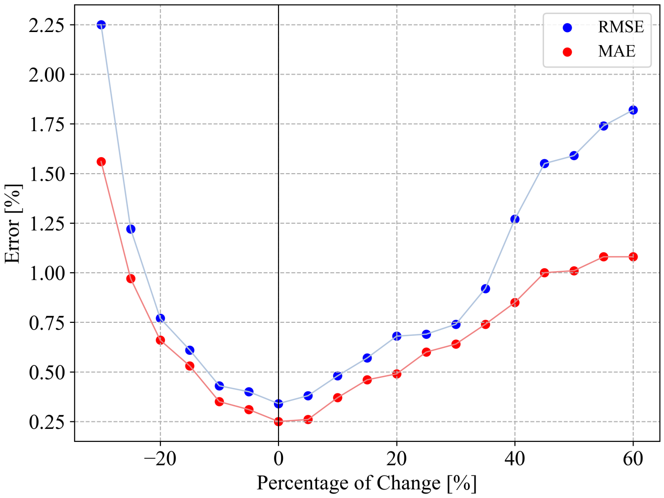


Figure 11: VB SoC vs percentage change in EWH power rating.

6.3. Time Efficiency

An essential metric to consider when running deep learning models for online applications is the computation time. In this research, a Windows PC with Intel(R) Core(TM) i7-9700 @3.00 GHz, 32GB memory, NVIDIA GeForce RTX 2080Ti was used to run the pre-trained models. The results

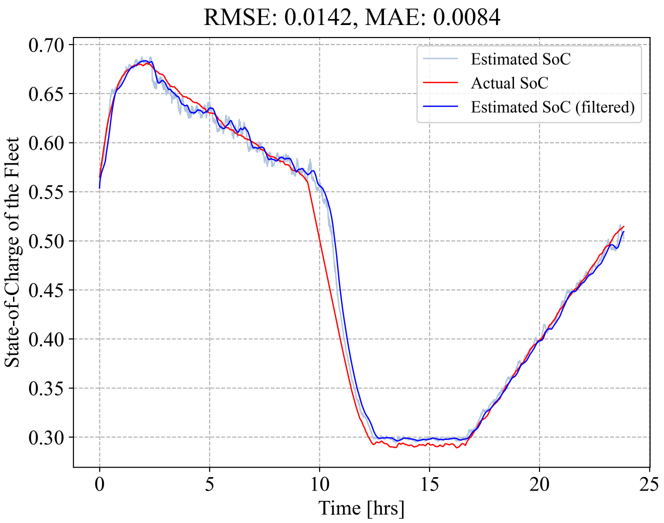


Figure 12: VB SoC estimation when 10% of packets are lost.

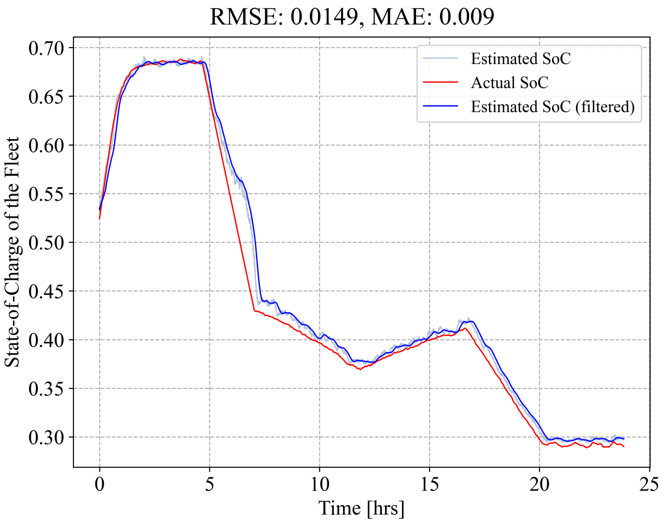


Figure 13: VB SoC estimation when 10% of charge/discharge decisions by the PEM coordinator are lost.

show that the average testing computing time was 28.4 milliseconds.

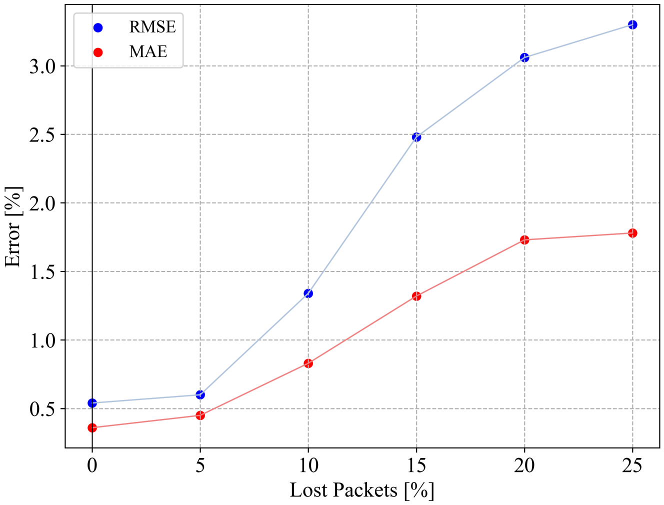


Figure 14: VB SoC estimation vs percentage of lost packets.

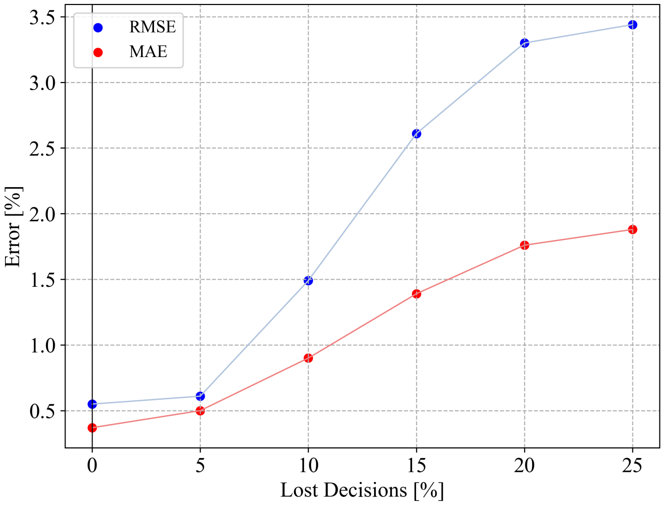


Figure 15: VB SoC estimation error vs percentage of lost charge/discharge decisions.

6.4. Influence of training data size

To achieve good SoC estimation, DL-based model require a certain amount of data for training. However, obtaining this amount of data can be challenging.

Overall, the results show that DL models benefit from more training data for efficiently training and generalization ability as shown in Figure 16.

Although there is a trend of improvement with more data, the effect of more training data does saturate after 72 hrs, indicating that a certain amount of data is sufficient for good performance.

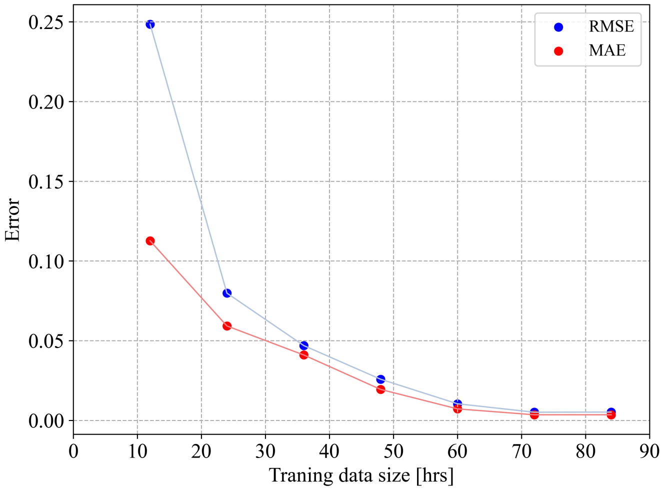


Figure 16: Proposed DL-model performance for different training data sizes.

6.5. Robustness against diverse population

The proposed DL-model was restructured to get more features as input. This section presents the evaluation of the model under diverse fleet of devices (e.g., EWHs, ESSs). The results of SoC estimation under diverse DER population is shown in Figure 17. The results of this study indicate that the model demonstrates robustness against diverse population, with high accuracy.

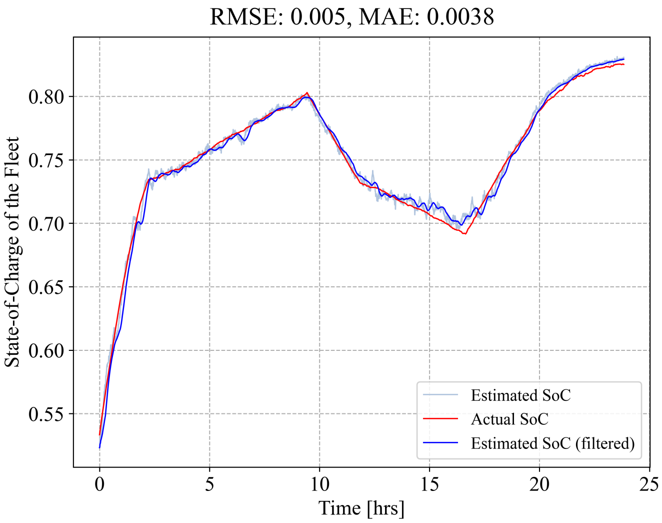


Figure 17: VB SoC estimation performance under diverse DER population.

Table 3: Summary of VB SoC Estimation Results.

Scenario	RMSE	MAE	Scenario	RMSE	MAE	Scenario	RMSE	MAE
Exp.: Population Size			-40%	0.0095	0.008	Exp.: Power rating of DERs		
-10%	0.0123	0.011	-35%	0.0082	0.0068	-25%	0.0122	0.0097
-5%	0.0068	0.0056	-30%	0.0066	0.0054	-20%	0.0077	0.0066
0%	0.0032	0.0022	-25%	0.0052	0.0043	-15%	0.0061	0.0053
5%	0.0036	0.0025	-20%	0.0048	0.0039	-10%	0.0043	0.0035
10%	0.0055	0.0043	-15%	0.0045	0.0037	-5 %	0.004	0.0031
15%	0.0076	0.0059	-10%	0.0036	0.0028	0%	0.0034	0.0025
20%	0.0099	0.0074	-5%	0.0041	0.0029	5%	0.0038	0.0026
25%	0.0121	0.0089	0%	0.0036	0.0028	10 %	0.0048	0.0037
30%	0.0143	0.0109	5%	0.0039	0.003	15%	0.0057	0.0046
Exp.: Background Demand			10%	0.0043	0.0034	20%	0.0068	0.0049
-25%	0.0127	0.0101	15%	0.0046	0.0036	25%	0.0069	0.006
-20%	0.0132	0.0103	20%	0.0049	0.0038	30%	0.0074	0.0064
-15%	0.0124	0.0095	25%	0.0049	0.0039	35%	0.0092	0.0074
-10%	0.0121	0.0094	30%	0.0053	0.0042	40 %	0.0127	0.0085
-5%	0.0101	0.0071	35%	0.0052	0.0043	Exp.: Lost packets		
0%	0.0088	0.0067	40%	0.0052	0.0043	0%	0.0054	0.0036
5%	0.01	0.0081	45%	0.0058	0.0047	5%	0.006	0.0045
10%	0.0116	0.0096	50%	0.006	0.0048	10%	0.0134	0.0083
15%	0.01	0.0084	55%	0.0057	0.0047	Exp.: Lost decisions		
20%	0.0134	0.0111	60%	0.0055	0.0046	0%	0.0055	0.0037
25%	0.013	0.011	65%	0.0063	0.0049	5 %	0.0061	0.005
30%	0.0136	0.0111	70%	0.0061	0.0051	10 %	0.0149	0.009
Exp.: Tank size of EWHs			75%	0.0059	0.0048			
-50%	0.0118	0.0097	80 %	0.0065	0.0053			
-45%	0.0104	0.0085						

7. Conclusion

This paper developed a deep learning approach to estimate the average SoC for a heterogeneous ensemble of EWHs and batteries operating under PEM. DL-based model capable of accurately estimating the SoC under different type of parametric variations such as population size, background demand, device parameter, and communication errors was presented. Comparisons were made among the different DL architectures and the results show the efficiency and good performance of the proposed method. The estimation of the PEM-VB SoC has been illustrated with an DL implementation for both EWH fleet and diverse fleet (including both EWHs and batteries) and the results show that in both cases the error RMSE is within 1.5%. Future work can focus on extending the current framework to include additional virtual battery parameters, such as upper and lower energy and power bounds. In addition, the proposed ML-based estimation method can be combined with predictive (physics-based) models to optimize the dispatch of DER fleets.

8. Acknowledgments

Computations were performed on the Vermont Advanced Computing Core supported in part by NSF award No. OAC-1827314. This work was in-part supported by the U.S. Department of Energy's Advanced Research Projects Agency-Energy (ARPA-E) award DE-AR0000694. M. Almassalkhi is co-founder of startup Packetized Energy, which is commercializing technology related to Packetized Energy Management.

Appendix A.

The following GitHub repository contains the data used in this document: https://github.com/mmatar3/VB_data

References

[1] P. Denholm, J. Cochran, Balancing area coordination: Efficiently integrating renewable energy into the grid, Tech. rep., National Renewable Energy Laboratory (2015).
URL https://www.nrel.gov/docs/fy15osti/63037.pdf

[2] H. Mavalizadeh, L. A. Duffaut Espinosa, M. R. Almassalkhi, Decentralized frequency control using packet-based energy coordination, in: 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2020, pp. 1–7. doi:10.1109/SmartGridComm47815.2020.9302972.

[3] L. A. D. Espinosa, A. Khurram, M. Almassalkhi, Reference-tracking control policies for packetized coordination of heterogeneous der populations, IEEE Transactions on Control Systems Technology 29 (6) (2021) 2427–2443. doi:10.1109/TCST.2020.3039492.

[4] J. Zhang, P. Wang, N. Zhang, Frequency regulation from distributed energy resource using cloud-edge collaborations under wireless environments, IEEE Transactions on Smart Grid 13 (1) (2022) 367–380. doi:10.1109/TSG.2021.3109006.

[5] A. R. Coffman, N. Cammardella, P. Barooah, S. Meyn, Aggregate flexibility capacity of tcls with cycling constraints, IEEE Transactions on Power Systems 38 (1) (2023) 52–62. doi:10.1109/TPWRS.2022.3160071.

[6] J. T. Hughes, A. D. Domínguez-García, K. Poolla, Identification of virtual battery models for flexible loads, IEEE Transactions on Power Systems 31 (6) (2016) 4660–4669. doi:10.1109/TPWRS.2015.2505645.

[7] J. L. Mathieu, M. Kamgarpour, J. Lygeros, G. Andersson, D. S. Callaway, Arbitraging intraday wholesale energy market prices with aggregations of thermostatic loads, IEEE Transactions on Power Systems 30 (2) (2015) 763–772. doi:10.1109/TPWRS.2014.2335158.

[8] S. P. Nandanoori, I. Chakraborty, T. Ramachandran, S. Kundu, Identification and validation of virtual battery model for heterogeneous devices (2019). arXiv:1903.01370.

[9] L. A. D. Espinosa, A. Khurram, M. R. Almassalkhi, A virtual battery model for packetized energy management, in: 2020 59th IEEE Conference on Decision and Control (CDC), 2020, pp. 42–48. doi:10.1109/CDC42340.2020.9304065.

[10] F. Chollet, Deep learning with Python, Simon and Schuster, 2021.

[11] M. Zhang, Y. Liu, D. Li, X. Cui, L. Wang, L. Li, K. Wang, Electrochemical impedance spectroscopy: A new chapter in the fast and accurate estimation of the state of health for lithium-ion batteries, Energies 16 (4) (2023). doi:10.3390/en16041599.
URL https://www.mdpi.com/1996-1073/16/4/1599

[12] M. Zhang, D. Yang, J. Du, H. Sun, L. Li, L. Wang, K. Wang, A review of soh prediction of li-ion batteries based on data-driven algorithms, Energies 16 (7) (2023). doi:10.3390/en16073167.
URL https://www.mdpi.com/1996-1073/16/7/3167

[13] Z. Y. Khan, Z. Niu, A. S. Nyamawe, I. ul Haq, A deep hybrid model for recommendation by jointly leveraging ratings, reviews and metadata information, Engineering Applications of Artificial Intelligence 97 (2021) 104066.

[14] X. Yu, N. Ma, L. Zheng, L. Wang, K. Wang, Developments and applications of artificial intelligence in music education, Technologies 11 (2) (2023). doi:10.3390/technologies11020042.
URL https://www.mdpi.com/2227-7080/11/2/42

[15] I. Chakraborty, S. P. Nandanoori, S. Kundu, Virtual battery parameter identification using transfer learning based stacked autoencoder (2018). arXiv:1810.04642.

[16] I. Chakraborty, S. P. Nandanoori, S. Kundu, K. Kalsi, Stochastic virtual battery modeling of uncertain electrical loads using variational autoencoder, in: 2020 American Control Conference (ACC), 2020, pp. 1305–1310.

[17] M. Almassalkhi, L. A. Duffaut Espinosa, P. D. Hines, J. Frolik, S. Paudyal, M. Amini, Asynchronous Coordination of Distributed Energy Resources with Packetized Energy Management, Springer New York, New York, NY, 2018, pp. 333–361.

[18] L. A. Duffaut Espinosa, M. Almassalkhi, P. Hines, J. Frolik, System properties of packetized energy management for aggregated diverse resources, Power Systems Computation Conference (June 2018).

[19] L. A. Duffaut Espinosa, M. Almassalkhi, A packetized energy management macromodel with quality of service guarantees for demand-side

resources., IEEE Transactions on Power Systems 35 (5) (2020) 3660–3670. doi:10.1109/TPWRS.2020.2981436.

[20] L. A. Duffaut Espinosa, A. Khurram, M. Almassalkhi, A virtual battery model for packetized energy management, in: 59th IEEE Conference on Decision and Control (CDC), 2020, pp. 42–48.

[21] S. R. Khazeiynasab, J. Zhao, I. Batarseh, B. Tan, Power plant model parameter calibration using conditional variational autoencoder, IEEE Transactions on Power Systems 37 (2) (2021) 1642–1652.

[22] M. Matar, P. G. Estevez, P. Marchi, F. Messina, R. Elmoudi, S. Wshah, Transformer-based deep learning model for forced oscillation localization, International Journal of Electrical Power & Energy Systems 146 (2023) 108805.

[23] F. Almutairy, L. Scekic, M. Matar, R. Elmoudi, S. Wshah, Detection and mitigation of GPS spoofing attacks on Phasor Measurement Units using deep learning, International Journal of Electrical Power & Energy Systems 151 (2023) 109160.

[24] J. Bedi, D. Toshniwal, Empirical mode decomposition based deep learning for electricity demand forecasting, IEEE access 6 (2018) 49144–49156.

[25] D. Svozil, V. Kvasnicka, J. Pospichal, Introduction to multi-layer feed-forward neural networks, Chemometrics and intelligent laboratory systems 39 (1) (1997) 43–62.

[26] M. Matar, B. Xu, R. Elmoudi, O. Olatujoye, S. Wshah, A deep learning-based framework for parameters calibration of power plant models using event playback approach, IEEE Access 10 (2022) 72132–72144. doi:10.1109/ACCESS.2022.3188313.

[27] S. Wshah, R. Shadid, Y. Wu, M. Matar, B. Xu, W. Wu, L. Lin, R. Elmoudi, Deep learning for model parameter calibration in power systems, in: 2020 IEEE International Conference on Power Systems Technology (POWERCON), 2020, pp. 1–6. doi:10.1109/POWERCON48463.2020.9230531.

[28] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions, Journal of big Data 8 (1) (2021) 1–74.

[29] Y. Cao, Y. Ding, M. Jia, R. Tian, A novel temporal convolutional network with residual self-attention mechanism for remaining useful life prediction of rolling bearings, Reliability Engineering & System Safety 215 (2021) 107813.

[30] J. Yan, L. Mu, L. Wang, R. Ranjan, A. Y. Zomaya, Temporal convolutional networks for the advance prediction of enso, Scientific reports 10 (1) (2020) 1–15.

[31] X. Yuan, S. Qi, Y. Wang, K. Wang, C. Yang, L. Ye, Quality variable prediction for nonlinear dynamic industrial processes based on temporal convolutional networks, IEEE Sensors Journal 21 (18) (2021) 20493–20503.

[32] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization., Journal of machine learning research 13 (2) (2012).

[33] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[34] F. Yang, W. Li, C. Li, Q. Miao, State-of-charge estimation of lithium-ion batteries based on gated recurrent neural network, Energy 175 (2019) 66–75.

[35] A. Khurram, M. Amini, L. A. D. Espinosa, P. D. Hines, M. Almassalkhi, Real-time grid and der co-simulation platform for validating large-scale der control schemes, arXiv preprint arXiv:2106.01993 (2021).

[36] S. Kundu, J. Hansen, J. Lian, K. Kalsi, Assessment of optimal flexibility in ensemble of frequency responsive loads, in: 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm), IEEE, 2017, pp. 399–404.