GiNet: Integrating Sequential and Context-Aware Learning for Battery Capacity Prediction

IEEE VTC Spring 2025

Sara Sameer[1](Presenter), Wei Zhang[1], Xin Lou[1] ,
Qingyu Yan[2] , Terence Goh [3] , Yulin Gao [3]

1. Singapore Institute of Technology
2. Nanyang Technological University
3. ST Engineering

Paper Code

Table of Content

Why Batteries Matter More Than Ever?

Headline Highlight: Existing EV Batteries may last upto 40% longer than expected - *Standford News, 2024*

- Electric Vehicles, Smartphones, Solar energy – all depend on battery reliability.
- Lower battery replacement = lower total cost of ownership.
- Battery estimation models must adapt to real-world variations.
- Need models that understands the real-time driving and battery degradation.

Why State of Charge is Crucial?

SINGAPORE INSTITUTE OF TECHNOLOGY

- State of Charge (SoC) estimation is crucial for:
  - Optimizing battery performance
  - Preventing overcharge or deep discharge,
  - Maximizing the battery’s life.
- Conventional approaches like Support Vector Regressors (SVR), Kalman filters and LSTM suffer from inability to capture temporal patterns and dependencies in high dimensional time-series data.

Panasonic 18650PF Li-ion Battery Data for SoC estimation

Attribute	Description
Battery Type	2.9Ah Panasonic 18650PF cell
Charging Profile	Cycled at 1C rate at 25degC
Discharging Temperatures	25degC, 10degC, 0degC, -10degC, and -20degC
Drive Cycles Profiles	Cycle 1, Cycle 2, Cycle 3, Cycle 4, US06, HWFET, UDDS, LA92, NN
Features	Current(I), Voltage(V), Temperature(T)

Research motivation for Transformers Model

SINGAPORE INSTITUTE OF TECHNOLOGY

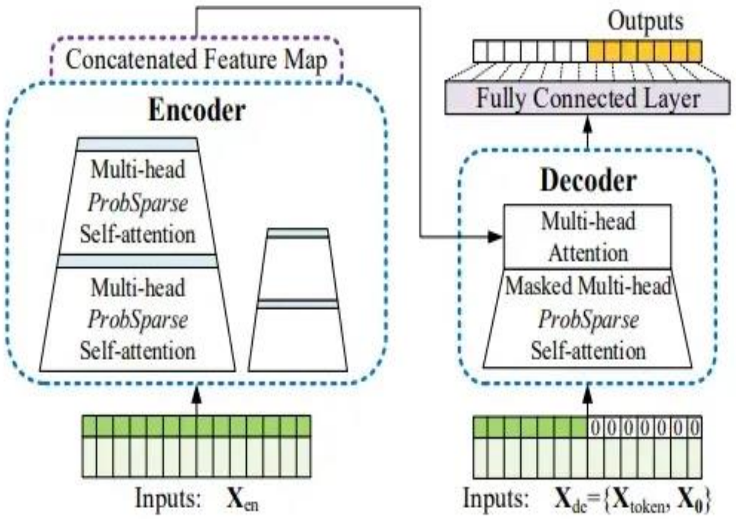
Transformers:

- Capture long-range dependencies
- No need for recurrent layers
- Suited for complex temporal patterns

Informer: Efficient Transformer Variant

- Handles long sequences
- Uses ProbSparse attention
- Optimized for high-dimensional time series

Encoder:			N
Inputs	1x3 Conv1d	Embedding ($d = 512$)	
ProbSparse Self-attention Block	Multi-head ProbSparse Attention ($h = 16, d = 32$)		
	Add, LayerNorm, Dropout ($p = 0.1$)		
	Pos-wise FFN ($d_{\text{inner}} = 2048$), GELU		4
	Add, LayerNorm, Dropout ($p = 0.1$)		
Distilling	1x3 conv1d, ELU		
	Max pooling (stride $= 2$)		
Decoder:			N
Inputs	1x3 Conv1d	Embedding ($d = 512$)	
Masked PSB	add Mask on Attention Block		
Self-attention Block	Multi-head Attention ($h = 8, d = 64$)		2
	Add, LayerNorm, Dropout ($p = 0.1$)		
	Pos-wise FFN ($d_{\text{inner}} = 2048$), GELU		
	Add, LayerNorm, Dropout ($p = 0.1$)		
Final:			
Outputs	FCN ($d = d_{\text{out}}$)		



Informers – Embedding

$$X = \{x_1, x_2, \ldots, x_n\}, \quad x_t = [\text{Temperature}_t, \text{Current}_t, \text{Voltage}_t] \qquad (1)$$

$$E_t = f_{\text{embed}}(x_t) + f_{\text{temporal}}(t) + f_{\text{spatial}}(x_t) \qquad (2)$$

$$\mathbf{X}_{\text{temporal}} = \mathbf{E}_{\text{hour}} + \mathbf{E}_{\text{weekday}} + \mathbf{E}_{\text{day}} + \mathbf{E}_{\text{month}}$$

$$\mathbf{X}_{\text{value}} = \text{Conv1D}(\mathbf{X}_{\text{input}})$$

$$\text{PE}(t, 2i) = \sin\left(\frac{t}{10000^{2i/d_{\text{model}}}}\right)$$

$$\text{PE}(t, 2i+1) = \cos\left(\frac{t}{10000^{2i/d_{\text{model}}}}\right)$$

Embeddings:

1. Value Embedding (Token Embedding): Converts I, V and T into higher dimensional data.

2. Positional Embedding: It encodes the position of each time step in the sequence.

3. Temporal Embedding: It encodes the time stamp into time aware encoding.

Informers - Architecture

$$Q = EW_Q, \quad K = EW_K, \quad V = EW_V \tag{3}$$

$$A = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \qquad \text{OR} \qquad A_{\mathrm{ProbSparse}} = \mathrm{softmax}\left(\frac{Q'K^\top}{\sqrt{d_k}}\right)V \tag{4}$$

$$Q' = \mathrm{Top}_u(\|Q\|) \tag{5}$$

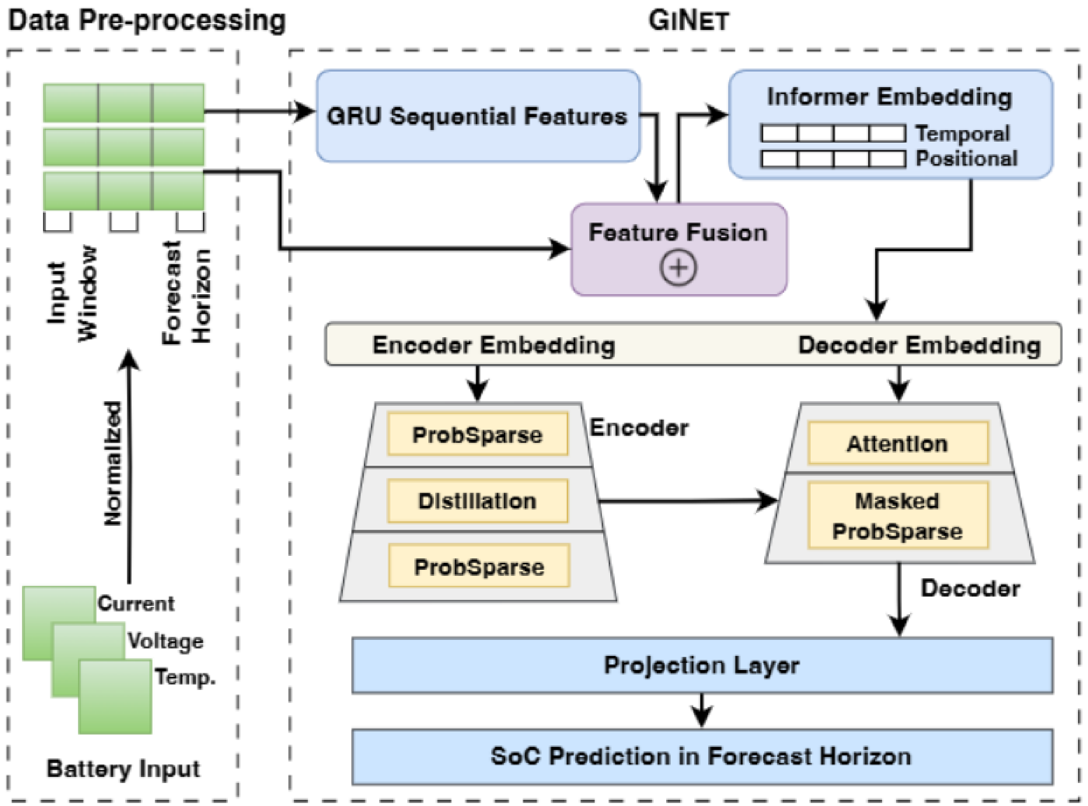
🔑 Input embeddings are transformed into **queries (Q)** and **keys (K)** using learned weights
- These define **how much attention** each input should give to others

🧠 Instead of computing attention for all key-value pairs, **ProbSparse Attention** samples only the most relevant ones, based on probability
- Making attention faster and lighter for long sequences

🔁 Instead of using a large model during inference, **distillation** trains a student model to mimic the behavior of a powerful teacher model
- Preserving accuracy while reducing computational cost

GINet

Work on SoC Estimation



GINET: GRU-enhanced Informer for battery capacity prediction

Feature Fusion: Combines sequential (GRU) & contextual (Informer) insights

Optimized Architecture: Tailored integration for battery monitoring data

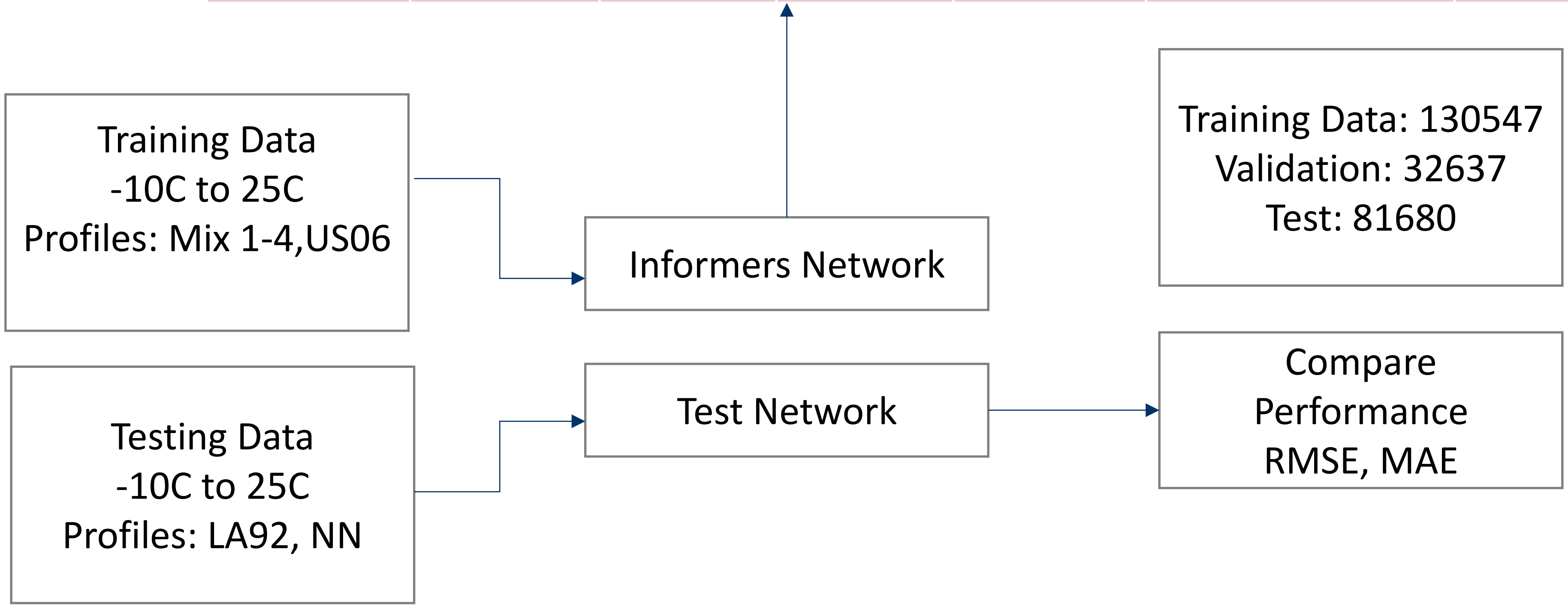
Performance:

- **MAE:** 0.11 (high accuracy)
- **Improvement:** +76% (vs. GRU), +27% (vs. Informer)

Impact: Enhanced SoC prediction efficiency

Experimental Setup

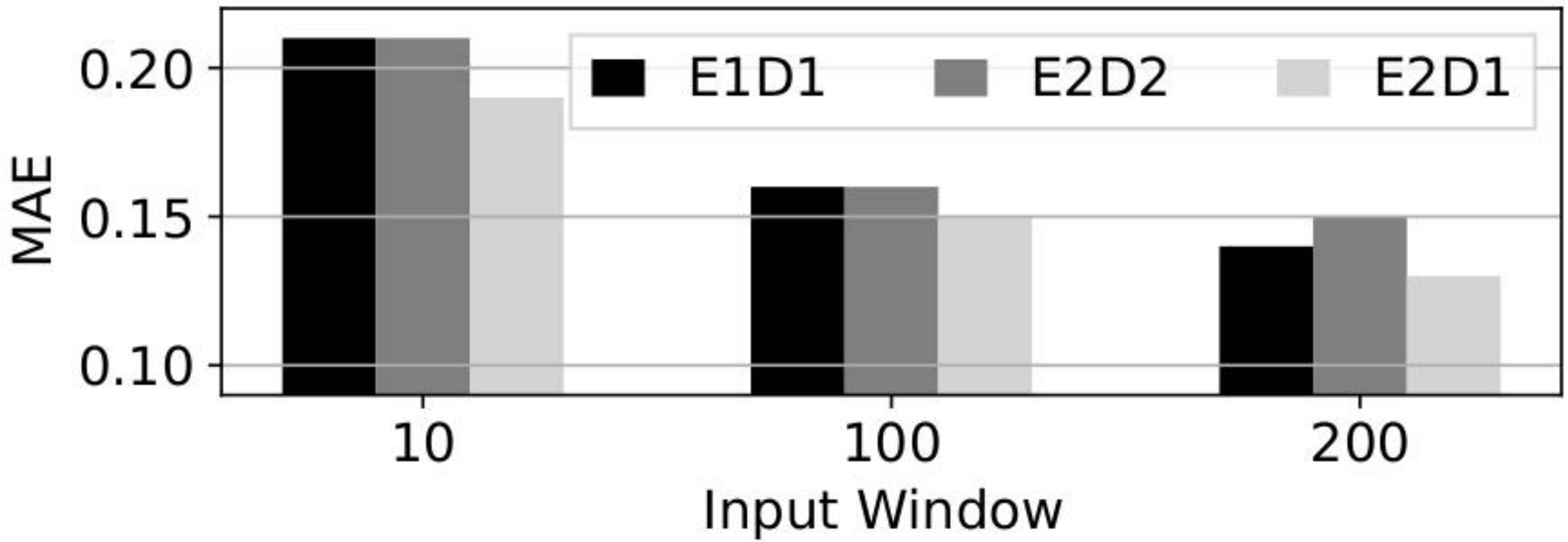
d_model	n_heads	e_layers	d_layers	Drop out	Learning Rate	Epoch	Batch Size
512	8	2	1	0.1	0.0001	20	32



Results

Forecast Horizon		25			10		
Input Length		200	100	10	200	100	10
LSTM	RMSE ↓	0.30	0.31	0.43	0.29	0.30	0.42
	MAE ↓	0.26	0.27	0.36	0.26	0.27	0.33
GRU	RMSE ↓	0.27	0.29	0.30	0.26	0.28	0.29
	MAE ↓	0.24	0.25	0.27	0.24	0.25	0.25
Informers	RMSE ↓	0.20	0.21	0.22	0.19	0.21	0.21
	MAE ↓	0.18	0.18	0.20	0.17	0.17	0.18
GINET	RMSE ↓	0.15	0.18	0.22	0.14	0.16	0.17
	MAE ↓	0.13	0.15	0.19	0.11	0.14	0.15

Sensitivity Analysis – Encoder Decoder Layers



Directions for Future Research

- Integrate physical models to fuse domain knowledge
- Quantify prediction confidence with uncertainty aware forecasting.
- Optimize the model for on device SoC estimation.

Any Questions?
Thank You!