2026 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
APPLIED ARTIFICIAL INTELLIGENCE
AUG. 11-13, 2026 - NOVI, MICHIGAN

DEEP LEARNING-BASED ROTOR TEMPERATURE ESTIMATION FOR RARE-EARTH-FREE MOTORS

Farzaneh Tatari, PhD[1], Mohsen Mirza Aligoudarzi [2]

[1, 2] Drive System Design Inc., Farmington Hills, MI, US

ABSTRACT

This paper presents a deep learning-based approach for online rotor temperature estimation in electrically excited synchronous motors (EESMs). Accurate rotor temperature estimation is critical for ensuring safe operation, improving performance, and enabling reliable thermal management of electric traction motors. Recurrent neural network (RNN) architectures, including gated recurrent unit (GRU) and long short-term memory (LSTM) networks, are investigated to develop a data-driven thermal virtual sensor capable of capturing the temporal dynamics of motor operation. Experimental data collected from a 190 kW EESM prototype are used to train and evaluate the proposed models. A systematic training, testing, and 10-fold cross-validation framework is employed to assess prediction accuracy and generalization capability. The results demonstrate that the GRU-based model achieves higher prediction accuracy than the LSTM model while maintaining comparable inference latency. The proposed approach provides an efficient and lightweight solution for real-time rotor temperature estimation suitable for embedded motor control applications.

Citation: F. Tatari, M. M. Aligoudarzi, "Deep Learning for Virtual Sensor Modeling," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 11-13, 2026.

1. INTRODUCTION

The rapid shift toward sustainable transportation and the large-scale adoption of electric vehicles (EVs) have intensified the need for reliable thermal management of electric motors. Maintaining component temperatures within allowable limits is critical for reliability and safety. Proper thermal management also improves cooling efficiency, and maintains torque and power capacity [1]. Although embedded sensors can provide accurate real-time measurements, direct temperature sensing, particularly in rotating components such as the rotor, is often impractical due to mechanical constraints, wiring complexity, cost, and durability concerns. Consequently, model-based temperature estimation has become a practical and cost-effective alternative for monitoring the internal thermal state of electric machines [2]. While high-fidelity methods such as computational fluid dynamics (CFD) and finite-element (FE) thermal simulations provide accurate predictions, their computational burden makes them unsuitable for real-time embedded applications.

Motor temperature estimation methods can be broadly classified into three categories: indirect and observer-based approaches, lumped-parameter thermal networks (LPTNs), and supervised machine learning methods. Indirect approaches exploit the temperature dependence of electrical parameters. However, such methods are sensitive to parameter uncertainties, magnetic saturation, and loss modeling inaccuracies, which can lead to significant estimation errors [3]. Observer-based techniques extend this concept by embedding a motor thermal model within a state observer framework to reconstruct internal temperatures from measurable electrical signals. Although capable of high accuracy when the model is well-identified, their performance depends strongly on model fidelity and parameter calibration and may require periodic recalibration to compensate for aging and unmodeled effects [4-7]. LPTNs provide a physics-based alternative by discretizing the motor into interconnected thermal nodes representing components such as the stator, rotor, and bearings. Thermal resistances and capacitances approximate conductive and convective heat transfer paths, reducing the distributed heat equation to a set of ordinary differential equations. LPTNs are typically categorized as white-box or gray-box models. White-box formulations rely exclusively on geometry and material properties, whereas gray-box approaches integrate physical structure with experimentally identified parameters. Although computationally efficient and suitable for real-time implementation, accurate parameter identification often requires significant domain expertise and prototype data, and model development can be labor-intensive [8-10]. In recent years, supervised machine learning (ML), including deep learning (DL), has emerged as a data-driven alternative for motor temperature estimation [11-13]. These methods learn the nonlinear mapping between operating variables (e.g. speed, torque, and currents) and internal motor temperatures directly from data. Architectures such as ordinary least square (OLS) and multilayer perceptron (MLP) models [14], Convolutional neural networks (CNNs) [15], feedforward neural networks [16], long short-term memory (LSTM) and gated recurrent unit (GRU) networks [17], and temporal convolutional networks (TCNs) [18] have demonstrated strong predictive capability. While purely data-driven models may suffer from limited physical interpretability and can require substantial training data and model capacity, they offer notable advantages: improved generalization across motor types, reduced reliance on manual thermal circuit derivation, automation of the modeling workflow, and the ability to implicitly capture complex or unmodeled physical effects.

Electrically excited synchronous machines (EESMs) are gaining renewed interest in EV propulsion due to their rare-earth magnet-free architecture. Unlike permanent magnet synchronous machines, EESMs generate rotor flux via electrically excited copper windings, eliminating reliance on critical rare-earth materials. This reduces exposure to supply-chain constraints and cost volatility, making EESMs strategically attractive for large-scale EV deployment. In addition to their rare-earth–free advantage, EESMs offer high efficiency, high power density, wide speed operating range enabled by controllable field excitation, robust construction, and favorable acoustic performance. These characteristics position EESMs as a compelling solution for next-generation EV powertrains.

Accurate thermal modeling of EESMs is particularly important because rotor field excitation introduces distinct thermal dynamics compared to permanent-magnet machines. In [19], a low-order LPTN was developed for online thermal monitoring of an EESM, with separate identification of thermal resistances and capacitances. Similarly, [20] proposed a transient LPTN model for an air-cooled EESM for shape optimization. In our previous work [21], using classic ML, an OLS regression model was developed for estimating both rotor and stator temperatures for an EESM, where incorporating loss data as well as inputs' exponentially weighted moving averages and standard deviations improved prediction accuracy.

Motivated by the need for accurate, computationally efficient, and rare-earth–free motor thermal modeling, this paper presents, recurrent neural networks (RNNs) for EESM thermal modeling. Specifically, GRU and LSTM architectures are developed and systematically compared using experimental data from a prototype EESM to achieve accurate and fast real-time temperature estimation suitable for embedded implementation.

2. RECURRENT NEURAL NETWORKS (RNNs)

RNNs are neural architectures designed to model sequential and temporal data by maintaining an internal hidden state that evolves over time. Unlike feedforward networks, RNNs include recurrent connections that allow information from previous time steps to influence current outputs, making them well suited for time-series modeling and dynamical systems. Gated RNN architectures, including the long short-term memory (LSTM) network [22] and the gated recurrent unit (GRU) [23], introduced gating mechanisms that regulate information flow across time steps, enabling stable gradient propagation and improved learning of long-range temporal dependencies. In both LSTM and GRU networks, the training data is organized as sliding time-window sequences with length L where the sequence window can slide every couple of (milli)-seconds depending on the sample rate, memory and computational limitations. Each sequence window is treated as an independent training sample and processed by the GRU or LSTM network across its temporal steps.

In both LSTM and GRU, let $x(t) \in \mathbb{R}^{d_x}$ denote the input at time step t and $h(t) \in \mathbb{R}^{d_h}$ the hidden state where d_x and d_h are respectively the number of input features and network hidden units. $\sigma(\cdot)$ denotes the logistic sigmoid function, $\tanh(\cdot)$ is the hyperbolic tangent, and $\odot$ denotes element-wise multiplication. The matrices $W_{(\cdot)} \in \mathbb{R}^{d_h \times d_x}$, $U_{(\cdot)} \in \mathbb{R}^{d_h \times d_h}$, and biases $b_{(\cdot)} \in \mathbb{R}^{d_h}$ are trainable parameters.

2.1. *Long short-term memory (LSTM) networks*

An LSTM updates its states using three gates: input $i(t)$, forget $f(t)$, and output $o(t)$ gates, given as below.

$$i(t) = \sigma(W_i x(t) + U_i h(t-1) + b_i),$$

$$f(t) = \sigma(W_f x(t) + U_f h(t-1) + b_f),$$

$$o(t) = \sigma(W_o x(t) + U_o h(t-1) + b_o), \quad (1)$$

where hidden states updates containing

candidate cell $\tilde{c}(t)$ and cell (memory) $c(t) \in \mathbb{R}^{d_h}$ are as follows,

$$\begin{aligned} h(t) &= o(t) \odot \tanh(c(t)), \\ c(t) &= f(t) \odot c(t-1) + i(t) \odot \tilde{c}(t), \\ \tilde{c}(t) &= \tanh(W_c x(t) + U_c h(t-1) + b_c). \quad (2) \end{aligned}$$

The LSTM computational structure is shown in Figure 1. The input gate controls what new information is written into memory. The forget gate determines which information from the previous cell state should be retained and the output gate regulates how much of the memory state is exposed to the hidden state. This explicit memory mechanism enables LSTMs to capture long-term temporal dependencies.

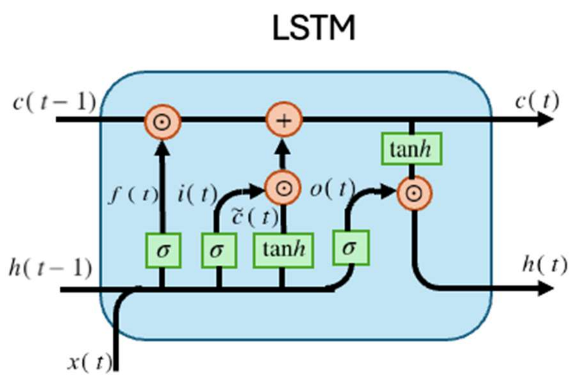

Figure 1: LSTM computational structure

For EESM rotor temperature estimation using LSTM network, LSTM layer and output layer are shown in Figure 2. The LSTM network processes a sequence of motor measurements $x(t)$ with sequence length of L and maintains internal hidden and cell states $(h(t), c(t))$ that capture the temporal dynamics of the system. At each time step the states are updated using the current inputs and the previous states. In this work the last hidden state of the sequence is passed to a fully connected output layer to estimate the rotor temperature. This structure allows the network to capture long-term thermal dynamics resulting from the historical data of the motor. During training, the LSTM network processes each input sequence using shared parameters across all time steps. After the forward pass generates the predicted rotor temperature, the error is propagated backward through the entire sequence using backpropagation through time, and the network parameters are updated using the Adam optimizer.

2.2. *Gated recurrent unit (GRU) networks*

Unlike LSTM, GRU does not maintain a separate cell state; instead, it combines memory and hidden representations into a single state variable. GRU uses two gates: update $z(t)$ and reset $r(t)$ gates, given below

$$\begin{aligned} z(t) &= \sigma(W_z x(t) + U_z h(t-1) + b_z), \\ r(t) &= \sigma(W_r x(t) + U_r h(t-1) + b_r), \quad (3) \end{aligned}$$

where the candidate hidden state $\tilde{h}(t)$ and hidden state $h(t)$ updates are as follows

$$\begin{aligned} \tilde{h}(t) &= \tanh(W_h x(t) + U_h(r(t) \odot h(t-1)) + b_h), \\ h(t) &= (1 - z(t)) \odot h(t-1) + z(t) \odot \tilde{h}(t). \quad (4) \end{aligned}$$

The GRU computational structure is shown in Figure 3.

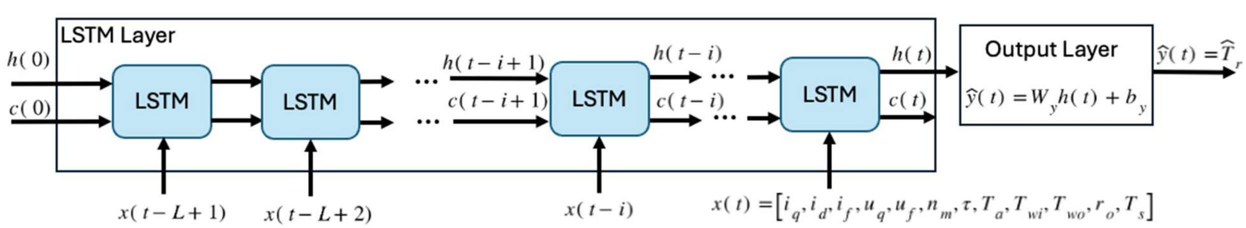

Figure 2: LSTM layer and output layer

The update gate $z(t)$ interpolates between retaining the previous hidden state and adopting the new candidate state. The reset gate $r(t)$ controls how much past information is used to compute the candidate state. This simplified gating structure reduces computational complexity while preserving the ability to model long-term dependencies. For EESM rotor temperature estimation using GRU network, GRU layer and output layer are shown in Figure 4. Similar to LSTM, the GRU network processes a sequence of motor measurements with length Lat each time t, $x(t-L+1), x(t-L+2), \ldots, x(t)$. At each time step the GRU updates the hidden state, which captures the temporal dynamics of the motor. The final hidden state of the sequence $h(t)$ is passed to a fully connected layer to estimate the rotor temperature.

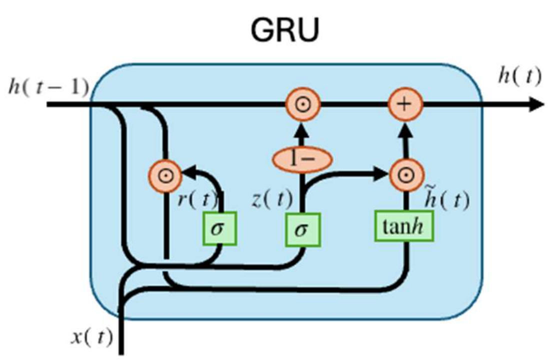

Figure 3: GRU computational structure

The recurrent hidden and cell states are initialized to zero at the beginning of every sequence, allowing the network to learn temporal dependencies within each window while preventing information leakage across different training samples.

2.3. *Weight Update Mechanism*

Both LSTM and GRU networks are trained using gradient-based optimization to minimize a predefined loss function, the mean squared error (MSE) in regression problems. Although their internal gating structures differ, the weight update mechanism follows the same fundamental training principle, backpropagation through time (BPTT), combined with a numerical optimizer. For a given input sequence, the recurrent network performs a forward pass across all time steps. At each time step t, the LSTM or GRU computes gate activations and updates its hidden (and, in the case of LSTM, cell) states using the current input, previous hidden state $h(t-1)$, and the network parameters (weight matrices and biases).

After propagating through the entire sequence, the network produces a predicted rotor temperature $\hat{y}(t) = \widehat{T}_r(t)$. The training loss is then computed using a regression criterion, typically

$$L = \frac{1}{N} \sum_{k=1}^{N} \left(\hat{y}_k - y_k\right)^2, \qquad (5)$$

where y_k is the measured rotor temperature and $N=8$ is the size of Mini Batch in the training process.

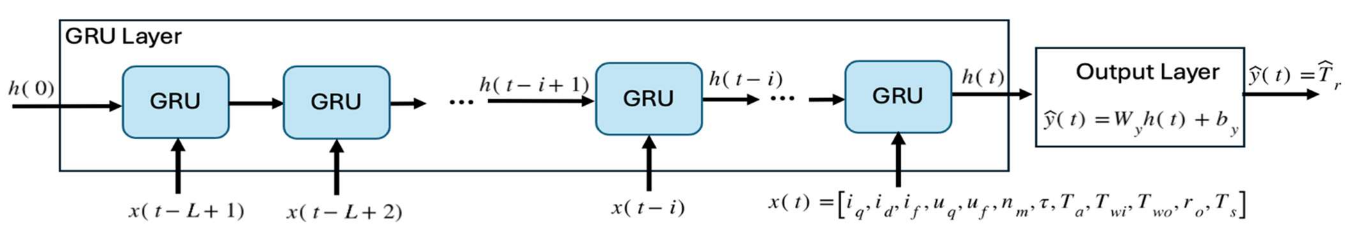

Figure 4: GRU layer and output layer

To update the weights, gradients of the loss function with respect to all trainable parameters are computed. Backpropagation through time recursively applies the chain rule to compute $\frac{\partial L}{\partial W}$ for each weight matrix W. For LSTM, gradients are computed with respect to input gate weights W_i, U_i, forget gate weights W_f, U_f, output gate weights W_o, U_o, and candidate cell weights W_c, U_c, output layer weights W_y as well as their associated biases. For GRU, gradients are computed with respect to update gate weights W_z, U_z, reset gate weights W_r, U_r, and candidate hidden state weights W_h, U_h, output layer weights W_y, along with their corresponding biases. Because the hidden state at each time step depends on previous states, gradients propagate backward through the entire sequence and accumulate contributions from all time steps.

After gradients are computed, the network parameters are updated using an optimization algorithm. In this work, the Adam optimizer is employed. Adam is an adaptive gradient method that maintains first- m_t and second-moment v_t estimates of the gradients

$$m_t = \beta_1 m_{t-1} + \left(1-\beta_1\right) g_t,$$
$$v_t = \beta_2 v_{t-1} + \left(1-\beta_2\right) g_t^2, \qquad (6)$$

where g_t is the gradient at iteration t, β_1 and β_2 $(0<\beta_i<1,\ i=1,2)$ are exponential decay rates controlling the moving averages of the first and second moments of the gradients. The parameter update rule is

$$W_{t+1} = W_t - \alpha \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \varepsilon}}, \qquad (7)$$

with learning rate α and bias-corrected moment estimates $\widehat{m}(t), \widehat{v}(t)$. Because the moment estimates are initialized at zero, bias-corrected estimates $\widehat{m}(t) = {}^{m_t}\!/\,1-\beta_1^{\,t}$ and $\widehat{v}(t) = {}^{v_t}\!/\,1-\beta_2^{\,t}$ are used in the parameter update to compensate for initialization bias during the early iterations [24]. Adam improves convergence stability and reduces sensitivity to gradient scaling, which is particularly beneficial in recurrent networks where gradients may vary significantly across time steps.

In every sequence sample training, the last hidden state $h(t)$ is passed to a fully connected output layer, which maps the hidden representation to the predicted output through a linear transformation

$$\widehat{y}(t) = W_y h(t) + b_y. \qquad (8)$$

3. EMPIRICAL DATA

At every time t, a data sample pair $(x(t), y(t))$ contains $x(t) = (x_1(t), \ldots, x_{d_x}(t))^T \in \mathbb{R}^{d_x}$, where $x_i(t),\ i = 1, \ldots, d_x$, represents a single input i measurement from an independent sensor in the vehicle's drive train and $y(t)$ is the target output, measured rotor temperature at time t. Each sample $x(t)$ contains motor measurement data, representing a single instance of data that is mapped to its associated $y(t)$. Because we can access this data through the electric control unit, no extra hardware needs to be deployed for deep learning models.

Inputs and target outputs for EESM thermal modeling using RNNs is listed in Table 1.

Table 1: Default inputs and output of the deep learning thermal model for EESM

Parameter type	*Parameter name*	*Selected*
Inputs	*Ambient temperature:* T_a	*Yes*
	WEG-In temperature: T_{wi}	*Yes*
	WEG-Out temperature: T_{wo}	*Yes*
	Oil temperature: T_o	***No***
	Oil flow rate: r_o	*Yes*
	Actual d-axis current: i_d	*Yes*
	Actual q-axis current: i_q	*Yes*
	Actual f-axis current: i_f	*Yes*
	Actual d-axis voltage: u_d	***No***
	Actual q-axis voltage: u_q	*Yes*
	Actual f-axis voltage: u_f	*Yes*
	Motor speed: n_m	*Yes*
	Motor torque: τ	*Yes*
	Stator windings temperature: T_s	*Yes*
Target Outputs	*Rotor windings temperature:* T_r	

The experimental dataset was collected from a three-phase, 190 kW EESM featuring a 54-slot/6-pole configuration and sampled at 10 Hz in multiple operating days [21]. The corresponding test bench setup is illustrated in Figure 5.

Figure 5: EESM in the test bench.

The dataset comprises approximately seven hours of operation, totaling more than 240,000 multi-dimensional samples. To enable deep learning of rotor thermal behavior, a dedicated temperature sensor was installed on the rotor for data collection purposes. This direct measurement provides ground-truth rotor temperature for model training and validation. Across the full dataset, the machine operates within a torque range of −150 to 200 Nm and a speed range of −1000 to 8000 rpm, ensuring that the learned models are exposed to a wide spectrum of operating conditions.

4. GRU and LSTM FOR ROTOR TEMPERATURE ESTIMATION

To investigate the suitability of gated recurrent architectures for rotor temperature estimation in EESMs, both GRU and LSTM networks are implemented within an identical training and evaluation framework.

The database is partitioned (at the day level) into training and test sets to prevent temporal leakage. All input features are normalized using the mean and standard deviation computed exclusively from the training data, thereby avoiding information leakage from the test set and ensuring consistent scaling across all sequences.

In the first step, in order to investigate how the addition of physics-informed data to default input data can improve the performance of a GRU network for rotor temperature estimation, all the derived and losses input data in [21] were fed to the GRU network along with default input data. This increased the number of inputs from 14 to 21. However, the simulation results showed that incorporating the physics-informed data did not noticeably improve the performance of the GRU network for rotor temperature estimation and it only increased the size of the network which would increase the computational burden during deployment. Therefore, we proceed to the feature selection with only considering default inputs (given in Table 1).

To reduce input redundancy and improve generalization, forward sequential feature selection is performed prior to network training. A ridge-regression-based root mean squared error (RMSE) criterion with 10-fold cross-validation is employed to identify the most informative subset of input features. The 12 selected features (given in Table 1) are then retained in the original time-domain sequences for final model training.

The learning problem is formulated as a sequence-to-one regression task with fixed-length sequences of 3000 samples (corresponding to 5 minutes of operation with 10 Hz sample rate). The sequence window slides every 5 seconds; therefore, the LSTM and GRU networks estimate rotor temperature every 5 seconds using the previous 3000 input samples. Each sequence is mapped to the rotor temperature at its final time step, allowing the network to learn the nonlinear temporal mapping between historical operating conditions and the current rotor temperature. This formulation enables both GRU and LSTM networks to capture long-term dependencies inherent in thermal processes.

The network architecture, in both LSTM and GRU, consists of a sequence input layer, a single recurrent layer and a fully connected output layer. The recurrent layer in GRU network contains 16 hidden units where increasing the number of hidden units in the recurrent layer does not significantly improve the performance of the rotor temperature prediction without overfitting. In order to have a fair comparison between GRU and LSTM, 13 hidden units were picked for LSTM network that leads to approximately the same number of tunable parameters within both GRU and LSTM networks. Therefore the differences in the LSTM and GRU performance can be attributed directly to the gating mechanism rather than to network depth or parameter scaling. Both GRU and LSTM networks are trained using the Adam optimization algorithm with a maximum of 200 epochs, a mini-batch size of 8, and an initial learning rate of $\alpha = 0.001$. The training data are shuffled at every epoch to enhance generalization.

5. MODEL EVALUATION AND SIMULATION RESULTS

Deep learning is inherently data-driven and requires a structured training and validation procedure to ensure reliable generalization. During training, model parameters are optimized iteratively using the training dataset by minimizing a predefined loss function. However, strong performance on training data alone does not guarantee predictive capability on unseen operating conditions. Therefore, a separate validation strategy is essential to assess generalization performance and mitigate overfitting.

In this work, k-fold cross-validation is employed to evaluate model robustness. The available dataset is partitioned into k subsets (folds). For each iteration, the model is trained using $k-1$ folds and evaluated on the remaining fold. This procedure is repeated k times such that each fold serves as the validation set once. The average performance across all folds provides an unbiased estimate of the model's generalization capability.

Model performance is quantified using standard regression metrics, including mean squared error (MSE), mean absolute error (MAE), and maximum absolute error (MaxAE) between predicted rotor temperature sequences and ground-truth measurements. The held-out test sets are used to directly compare the predictive accuracy of the proposed GRU and LSTM models under unseen operating conditions.

The key performance metrics of GRU and LSTM models for EESM rotor thermal modeling is presented here where all data processing and model development is

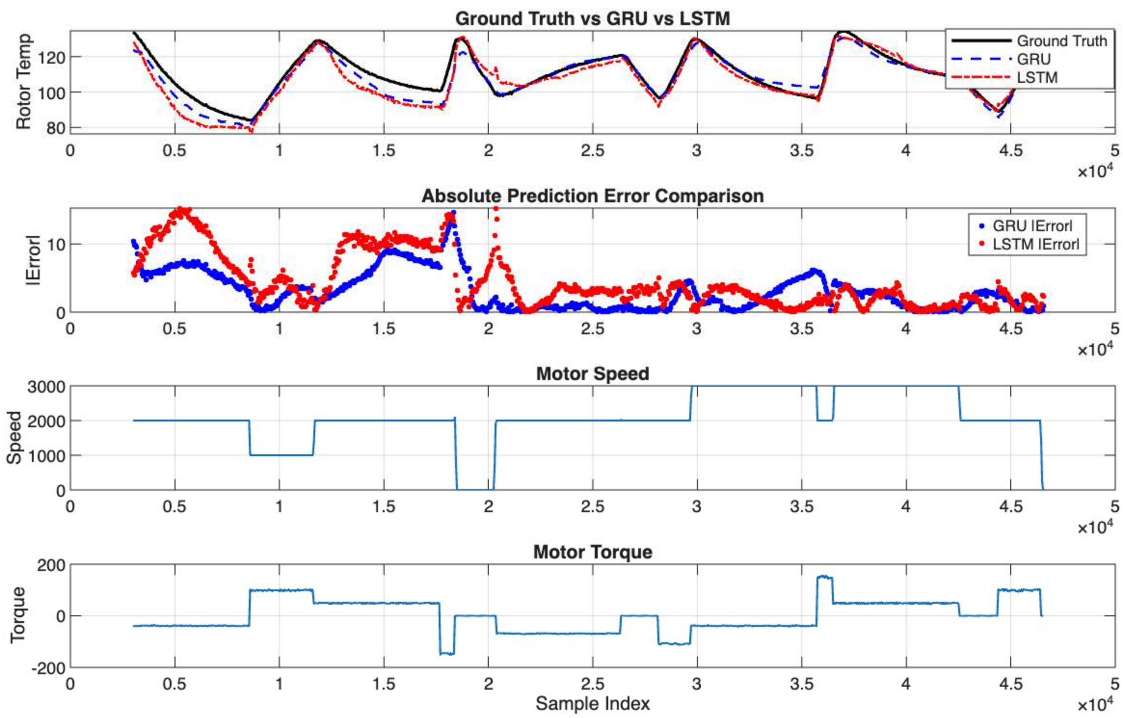

Figure 6: Motor speed, torque, GRU rotor temperature estimation for a measurement cycle (test set).

performed using MATLAB and its deep learning toolbox. Employing the experimental data and a 10-fold cross validation process, the MSE, MAE and MaxAE of the GRU and LSTM networks with loss inputs are given in Table 2.

Table 2: k-fold cross-validation testing results of GRU and LSTM for EESM rotor thermal modeling

	GRU rotor thermal model			*LSTM rotor thermal model*		
No.	MSE	MAE	MaxAE	MSE	MAE	MaxAE
1	4.0405	3.2505	9.0029	8.2982	7.2585	19.024
2	4.4784	3.3526	16.774	4.1296	2.9167	18.487
3	0.3863	0.3103	1.9997	0.5799	0.4491	2.4022
4	4.4003	3.7852	9.3368	5.4028	3.6238	21.87
5	0.6390	0.4549	2.9619	1.8334	1.2786	6.2767
6	0.2953	0.2407	1.0091	0.6065	0.4796	3.1613
7	3.8693	3.3421	18.185	5.3372	4.2944	19.984
8	0.7176	0.4819	3.6763	1.087	0.8088	3.0001
9	3.361	2.6405	9.0816	11.025	8.795	41.02
10	5.8365	4.7733	13.15	6.2362	4.3691	16.645
Ave.	2.8042	2.2632	8.5173	4.4535	3.4273	15.1870

The EESM rotor temperature estimation using GRU network has resulted in, the average MSE 2.8042, MAE 2.2632 and MaxAE 8.5173, and rotor temperature estimation using LSTM network has resulted in, the average MSE 4.4535, MAE 3.4273 and MaxAE 15.1870. The results indicate that the GRU network achieves better prediction accuracy than the LSTM network for EESM rotor thermal modeling.

Moreover, Figure 6 shows the GRU and LSTM rotor temperature estimator model performance for estimating the EESM rotor temperature for a measurement cycle where this cycle was unseen during training. Inference latency was evaluated by measuring the time required to generate one prediction for each test sample. A warm-up pass was performed prior to timing to avoid initialization overhead. For GRU and LSTM models, inference was measured using predict with mini-batch size equal to 1. The timing was repeated for all test samples, and the mean and standard deviation were reported.

All measurements were performed on the same hardware and under identical execution settings. GRU and LSTM with inference latency times are respectively, 0.0020 and 0.0019 seconds, indicating nearly identical execution speed. The low inference latency (~ 2 ms per prediction) demonstrates that the proposed models are suitable for real-time embedded motor control applications.

Comparing the existing results with the OLS method results in our previous study [21], here we have developed a lighter network in terms of the number of inputs and shorter sequence and also obtain better performance in comparison with [21].

6. CONCLUSIONS

This paper developed recurrent neural networks (RNNs) for online rotor temperature modeling of electrically excited synchronous motors (EESMs) which can improve EESM performance, safety, and operational reliability. Two RNNs, gated recurrent unit (GRU) and long short-term memory (LSTM) were studied where GRU outperformed LSTM in terms of performance and accuracy. The training and testing of GRU and LSTM were performed using experimental data where the final results of a 10-fold cross validation showed the performance of the given GRU and LSTM networks for EESM rotor temperature modeling.

7. REFERENCES

[1] O. Wallscheid and J. BÅNocker, “Derating of automotive drive systems using model predictive control,” IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics, 2017, pp. 31–36.

[2] O. Wallscheid, “Thermal monitoring of electric motors: State-of-the art review and future challenges,” IEEE Open Journal of Industry Applications, vol. 2, pp. 204–223, 2021.

[3] O. Wallscheid, T. Huber, W. Peters, and J. BÅNocker, “Real-time capable methods to determine the magnet temperature of permanent magnet synchronous motors - a review,” Annual Conference of the IEEE Industrial Electronics Society, 2014, pp. 811–818.

[4] G. Feng, C. Lai, J. Tjong, and N. C. Kar, “Non-invasive Kalman filter based permanent magnet temperature estimation for permanent magnet synchronous machines,” IEEE Trans. Power Electron., vol. 33, no. 12, pp. 10673–10682, Dec. 2018.

[5] G. Feng, C. Lai, W. Li, Z. Li, and N. C. Kar, “Efficient permanent magnet temperature modeling and estimation for dual three-phase PMSM considering inverter nonlinearity,” IEEE Trans. Power Electron., vol. 35, no. 7, pp. 7328–7340, Jul. 2020.

[6] K. Huang, B. Ding, C. Lai, and G. Feng, “Flux linkage tracking-based permanent magnet temperature hybrid modeling and estimation for PMSMs with data-driven-based core loss compensation,” IEEE Trans. Power Electron., vol. 39, no. 1, pp. 1410–1421, Jan. 2024.

[7] D. Liang, Z. Q. Zhu, J. Feng, Y. Li, and S. Guo, “Initial temperature determination for real-time thermal models in permanent magnet synchronous machines,” IEEE Trans. Transp. Electrific., vol. 11, no. 1, pp. 2204–2218, Feb. 2025.

[8] P.-O. Gronwald and T. A. Kern, “Traction motor cooling systems: A

literature review and comparative study," IEEE Trans. Transp. Electrific., vol. 7, no. 4, pp. 2892–2913, Dec. 2021.
[9] A. Boglietti, M. Cossale, S. Vaschetto, and T. Dutra, "Winding thermal model for short-time transient: Experimental validation in operative conditions," IEEE Trans. Ind. Appl., vol. 54, no. 2, pp. 1312–1319, Mar./Apr. 2018.
[10] O. Wallscheid and J. BÅNocker, "Global identification of a low-order lumped-parameter thermal network for permanent magnet synchronous motors," IEEE Transactions on Energy Conversion, vol. 31, no. 1, pp. 354–365, 2015.
[11] P. Sharma, A. Vijayvargiya, B. Singh, and R. Kumar, "Data driven temperature estimation of PMSM with regression models," in Proc. 2nd Int. Conf. Power, Control Comput. Technol., 2022, pp. 1–6.
[12] H. Jing et al., "Gradient boosting decision tree for rotor temperature estimation in permanent magnet synchronous motors," IEEE Trans. Power Electron., vol. 38, no. 9, pp. 10617–10622, Sep. 2023.
[13] X. Zhang, W. Lang, Y. Zhao, and C. Gong, "Transferable neural network method used for PMSM permanent magnet temperature estimation in electrical drive system," in Proc. Boao New Power Syst. Int. Forum Power Syst. New Energy Technol. Innov. Forum, 2024, pp. 507–513.
[14] W. KirchgÅNassner, O. Wallscheid, and J. BÅNocker, "Data-driven permanent magnet temperature estimation in synchronous motors with supervised machine learning: A benchmark," IEEE Transactions on Energy Conversion, vol. 36, no. 3, pp. 2059–2067, 2021.
[15] W. KirchgÅNassner, O. Wallscheid, and J. BÅNocker, "Estimating electric motor temperatures with deep residual machine learning," IEEE Transactions on Power Electronics, vol. 36, no. 7, pp. 7480–7488, 2020.
[16] J. Lee and J.-I. Ha, "Temperature estimation of PMSM using a difference-estimating feedforward neural network," IEEE Access, vol. 8, pp. 130855–130865, 2020.
[17] O. Wallscheid, W. Kirchgässner, and J. Böcker, "Investigation of long short-term memory networks to temperature prediction for permanent magnet synchronous motors," in Proc. Int. Joint Conf. Neural Netw., 2017, pp. 1940–1947.
[18] W. Kirchgässner, O.Wallscheid, and J. Böcker, "Estimating electric motor temperatures with deep residual machine learning," IEEE Trans. Power Electron., vol. 36, no. 7, pp. 7480–7488, Jul. 2021.
[19] E. Wang, P. Grabherr, P. Wieske, and M. Doppelbauer, "A low-order lumped parameter thermal network of electrically excited synchronous motor for critical temperature estimation," in 2022 International Conference on Electrical Machines (ICEM). IEEE, 2022, pp. 1562–1568.
[20] H. Spielmann, "Transient thermal lumped parameter model of an electrical excited synchronous machine with forced air cooling for shape optimization," in 2022 IEEE Vehicle Power and Propulsion Conference (VPPC). IEEE, 2022, pp. 1–7.
[21] F. Tatari, D. Trapp, J. Schneider and M. M. Aligoudarzi, "Data-driven Thermal Modeling for Electrically Excited Synchronous Motors - A Supervised Machine Learning Approach," 2024 IEEE Transportation Electrification Conference and Expo (ITEC), Chicago, IL, USA, 2024, pp. 1-6.
[22] S. Hochreiter, and J. Schmidhuber, "Long short-term memory", Neural

Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] K. Cho, B. van Merrienboer, and C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y.Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation", Conference on Empirical Methods in Natural Language Processing, 2014.

[24] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," ICLR, 2015.

8. CONTACT INFORMATION

Farzaneh Tatari
Senior Control Engineer,
Drive System Design,
37777 Interchange Dr, Farmington Hills, MI
Farzaneh.tatari@drivesystemdesign.com