Estimating Electric Motor Temperatures With Deep Residual Machine Learning

Wilhelm Kirchgässner, Oliver Wallscheid, *Member, IEEE*, and Joachim Böcker, *Senior Member, IEEE*

***Abstract*—Most traction drive applications lack accurate temperature monitoring capabilities, ensuring safe operation through expensive oversized motor designs. Classic thermal modeling requires expertise in model parameter choice, which is affected by motor geometry, cooling dynamics, and hot spot definition. Moreover, their major advantage over data-driven approaches, which is physical interpretability, tends to deteriorate as soon as their degrees of freedom are curtailed in order to meet the real-time requirement. In this article, deep recurrent and convolutional neural networks (NNs) with residual connections are empirically evaluated for their feasibility on predicting latent high-dynamic temperatures continuously inside permanent magnet synchronous motors. Here, the temperature profile in the stator teeth, winding, and yoke as well as the rotor's permanent magnets are estimated while their ground truth is available as test bench data. With an automated hyperparameter search through Bayesian optimization and a manual merge of target estimators into a multihead architecture, lean models are presented that exhibit a strong estimation performance at minimal model sizes. It has been found that the mean squared error and maximum absolute deviation performances of both, deep recurrent and convolutional NNs with residual connections, meet those of classic thermodynamics-based approaches, without requiring domain expertise nor specific drive train specifications for their topological design. Finally, learning curves for varying training set sizes and interpretations of model estimates through expected gradients are presented.**

***Index Terms*—Deep learning, functional safety, monitoring, neural networks (NNs), permanent magnet synchronous motor (PMSM), supervised machine learning, temperature estimation, thermal management.**

I. Introduction

THE permanent magnet synchronous motor (PMSM) is an appropriate option in various industry applications due to its high torque and power densities along with a high degree of efficiency. However, temperature-sensitive components are prone to failure under high thermal stress and must be monitored. Among those, the stator winding insulation and the permanent magnets are particularly noteworthy for being susceptible to heat, as the former may melt and the latter tends to irreversibly demagnetize in overload operation. Nowadays, hot spots in the stator are monitored with thermal sensors, which are irreplaceably embedded although their functionality degrades with age. Moreover, thermal sensors cannot be applied in the rotor due to its rotation and intricate structure without considerably high effort that would make production costs soar. Hence, a costly safety margin in embedded material is the today's measure against excessive heat. Consequently, the overload potential of a PMSM is never fully utilized. This problem, together with the constant cost pressure in the automotive sector, drives the investigation of sufficiently accurate real-time temperature estimation methods.

A. State of the Art

Various temperature estimation techniques were proposed and developed for decades. Among the most successful, lumped-parameter thermal networks (LPTNs) denote equivalent circuit diagrams approximating inner heat transfer by building on thermodynamic theory [2], [3]. Most notably, they are noninvasive, require no additional hardware, and achieve high estimation accuracy with comparably few model parameters, which is necessary for the real-time requirement. Then again, designing a high-performance, low-order LPTN is only possible by incorporating experimentally measured data, which marginalizes the LPTN's physical interpretability. In addition, short LPTN runtimes tradeoff the amount of monitorable hot spots directly. At the same time, expert domain knowledge is mandatory for a feasible LPTN parameterization, which strongly depends on the motor geometry and the cooling system.

Among estimation techniques that infer thermal behavior from electrical parameter estimates, there are to be mentioned those based on fundamental wave flux observers [4] and on high-frequency signal injection [5], [6]. Like LPTNs, both require sophisticated modeling efforts, i.e., the precise identification of the motor and inverter. Moreover, whilst for the former approach it has been shown that estimation accuracy suffers from lower accuracy at low motor speed, the latter technique monitors the high-frequency impedances of both, stator winding and the permanent magnets, to infer the thermal condition, which comes with additional losses on the system and may

Manuscript received August 9, 2020; revised October 16, 2020; accepted December 3, 2020. Date of publication December 17, 2020; date of current version March 5, 2021. This work was supported by the German Research Foundation (DFG) under Grant BO 2535/15-1. This paper was presented in part at the 2021 International Electric Machines and Drives, Hartford, CT, USA [1]. Recommended for publication by Associate Editor J. Hur. *(Corresponding author: Wilhelm Kirchgässner.)*

The authors are with the Department of Power Electronics and Electrical Drives, Paderborn University, D-33095 Paderborn, Germany, and also with the Automatic Control Department, Paderborn University, D-33095 Paderborn, Germany (e-mail: kirchgaessner@lea.upb.de; wallscheid@lea.upb.de; boecker@lea.upb.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2020.3045596.

Digital Object Identifier 10.1109/TPEL.2020.3045596

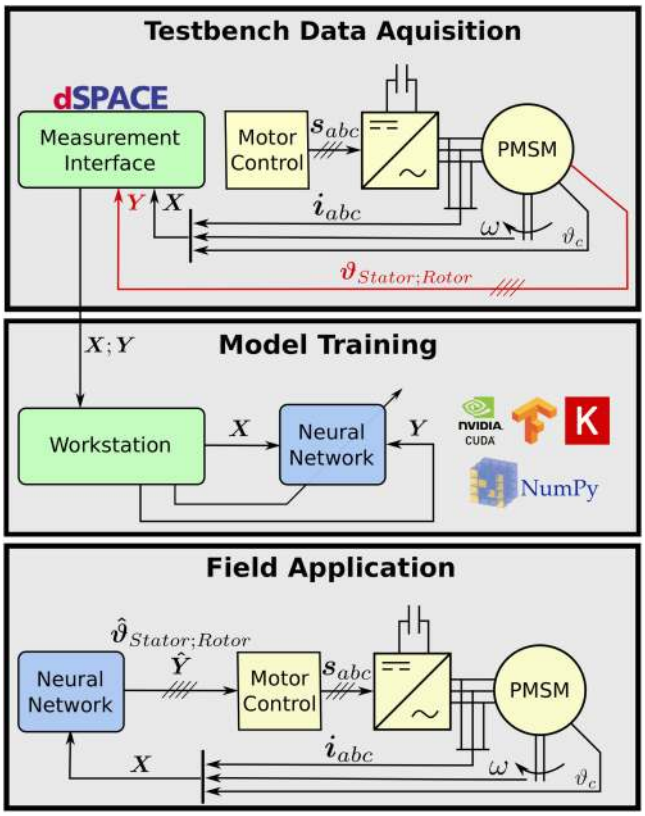

Fig. 1. Simplified scheme of the whole process from data acquisition at the test bench over model training up to the applied temperature monitoring in the field.

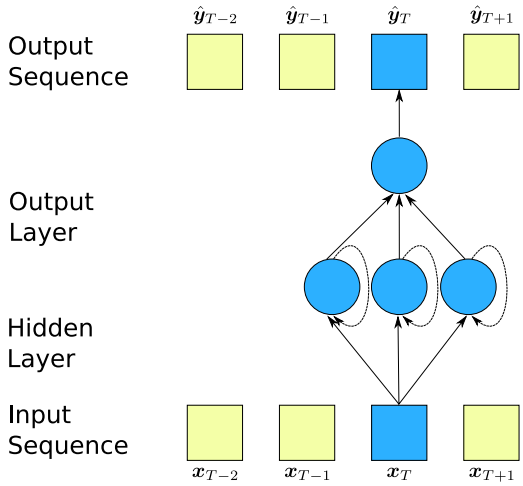

Fig. 2. Sequence learning with RNNs. For each observation $\boldsymbol{x}_T$ at time T there is a prediction $\hat{\boldsymbol{y}}_T$. Past information is retained through the RNN's memory cells and recurrent connections.

prove unsuitable in some applications due to electromagnetic interference [7].

A neural network (NN) or black-box approach, however, is independent of motor sheet information through being data-driven, i.e., being fitted on empirical measurements exclusively, and, for the same reason, does not suffer estimation degradation during operation points where physical model assumptions are violated. Especially recurrent NNs (RNNs) and convolutional NNs (CNNs) image the state of the art in various computer science domains, and are particularly useful in many sequence learning tasks [8], [9]. Their function approximation properties bode well for achieving high estimation accuracy without having to retrieve domain expertise. Fig. 1 sketches the big picture from the process of recording data up to the final application of NNs for online temperature monitoring.

B. Contribution

Introducing deep learning to the temperature estimation task in embedded systems comes with the following main challenges.

1) In contrast to LPTNs, many hyperparameters on top of the actual model parameters (weights) determine the training outcome. These need to be tuned automatically in order to not lose the autonomy of self-adaptivity as opposed to "design by an expert."
2) Generally, increasing the model order improves modeling capacities but compromises real-time computation. An appropriate tradeoff is to be opted for.
3) Having trained an NN, its estimates are difficult to interpret. It is desirable to be able to track deviations back to their causative input signals.
4) Real-world data are a limited resource as it costs time and money. How much data is necessary for an expedient regression accuracy?

Building on [1], this article investigates contemporary NN topologies on their ability to approximate latent and highly fluctuating temperature profiles inside PMSMs from commonly available electrical and thermal quantities. Hyperparameters that define the NN architecture and their weights' optimization as well as regularization are extensively optimized in a Bayesian search. This article sets itself apart from prior work by proposing a unifying model estimating all target temperatures with a minimal amount of parameters. Moreover, learning curves for varying training set sizes and input feature attribution by means of expected gradients to the eventual target temperature estimates are presented. Subject of all investigations is the data set[1] from [1] and [10]. In the following, we will briefly report on the utilized machine learning pipeline including basics of the investigated NN topologies. All code and more details are available at [11] to assist related work.

II. NN Architectures

A. Recurrent Architectures With Memory Blocks

Various breakthroughs in research domains such as speech recognition [12], machine translation [13], or wherever long-term memory is required [14] were reported to be successful due to RNNs and their particular recursive structure. Their recursive property enables preservation of past observations (see Fig. 2),

[1][Online]. Available: www.kaggle.com/wkirgsn/electric-motor-temperature

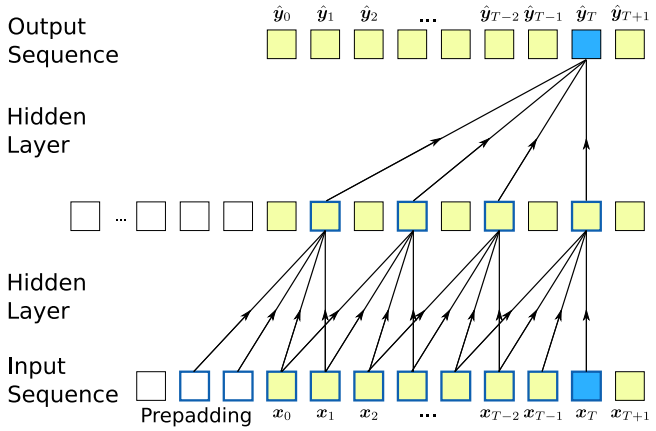

Fig. 3. Sequence learning with TCNs. Each prediction $\hat{\boldsymbol{y}}_T$ is informed by the current observation $\boldsymbol{x}_T$ and a finite set of past observations (framed blue), denoting the network's receptive field.

which is disregarded by most other machine learning algorithms, e.g., feed-forward NNs (FNNs). A crucial building block of popular RNN topologies is the so-called long short-term memory (LSTM) block that applies gating functions to its forward and recurrent connections [15]. A further simplified version of an LSTM is the gated recurrent unit (GRU) [16]. The major challenge in training recurrent NNs is the exploding and vanishing gradient problem, which is addressed by these memory blocks. Both memory block types were reported to perform similarly well on various sequence learning tasks without any significant advantage being observed over the other [17].

Although RNNs have been studied on the temperature estimation task in PMSMs before in [18], they are coevaluated in this article's hyperparameter search in order to ensure comparability to the following newer topology.

B. Temporal Convolutional Networks

RNNs were dominating the field of sequential learning problems for years and, thus, have been commonly associated with them. However, CNNs, which were prevalent in the image processing domain, have been discovered to be of superior behavior over RNNs recently, even in classic long-term memory sequential problems [9], [19]. The architectural details rendering this possible were proposed by [9], who coin this certain topology temporal convolutional network (TCN), see Fig. 3. It inherits recent advances of applications on sequential data, e.g., causal and dilated convolutions, but distilled to a simpler form than previously facilitated.

Causality in a convolution denotes the correlation of past observations only, i.e., no contemplation of future observations at any single point in time. The term dilation pertains to the fact that receptive fields of a filter are distributed apart from each other over the sequence rather than being adjacent, effectively expanding the TCN's scope. This enables TCNs to detect crucial events further in the past, potentially outperforming the long memory capability of established RNN memory blocks. The receptive field depends on the number of hidden layers n_l, the length of the filter l_{kernel} and the dilation rate d in each layer. Since each filter is convolved with the full sequence over a sliding window of length w, it is said that a TCN has shared weights in contrast to an equivalent fully connected FNN. This reduces the total amount of trainable model parameters. All activations in subsequent layers are the sum of all filters at that time. Padding with zero ensures that layers and the CNN-output are of the same size as the input. A key distinguishment to RNNs is the fact that CNNs and TCNs do not maintain inner memory cells.

TABLE I
TEST BENCH PARAMETERS

Motors	device under test	load motor
Type	PMSM	induction machine
Power rating (nom./max.)	22/52 kW	160/210 kW
Voltage rating	177 V	380 V
Current rating (nom./max.)	110/283 A	293/450 A
Torque rating (nom./max.)	110/250 N m	380/510 N m
Pole pair number	8	1
Cooling	water-glycol	forced air
Inverter	3×SKiiP 1242GB120-4DW	
Typology	voltage source inverter 2-level, IGBT	
Inverter interlocking time	3.3 µs	
Controller hardware	dSPACE	
Processor board	DS1006MC, 4 cores, 2.8 GHz	

C. Residual Connections

Residual or skip connections denote the element-wise addition of the activations at the input $\boldsymbol{x}$ and output of a layer or group of layers $f(\cdot)$ within a network to a new output o [20]. Effectively, partial flowing information takes a shortcut and bypasses a set of NN layers

$$o = \boldsymbol{x} + f(\boldsymbol{x}). \tag{1}$$

This scheme can be repeated for subsequent layers arbitrarily. The effect on the accumulated gradients is such that the NN weights are trained to not reconstruct the target trajectory from scratch anymore, but rather to learn the difference between the target temperature profile and the corresponding input signals.

III. Test Setup and Validation

The dataset consists of 140 h multivariate measurements sampled at 2 Hz from a three-phase automotive traction PMSM (52 kW) mounted on a test bench, which is torque-controlled while its motor speed is determined by a speed-controlled load motor 210 kW and fed by a two-level IGBT inverter (Semikron: 3xSKiiP 1242GB120-4DW). Furthermore, a dSPACE DS1006MC rapid-control-prototyping system is deployed on the test bench. All measurements were recorded by dSPACE analog-digital-converters, which have been synchronized with the control task. The most important test bench parameters are compiled in Table I. Fig. 4 shows the test bench with an exemplary PMSM of similar characteristics.

Fig. 4. Test bench with an exemplary PMSM of similar characteristics.

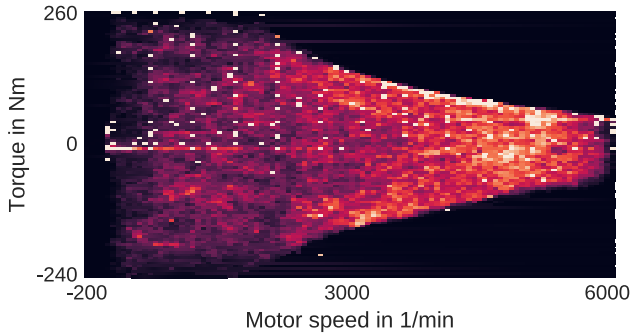

Fig. 5. Frequency of set operation points. Brighter areas are visited more often.

All temperatures are measured with embedded thermocouples (type K—class 1), and the rotor temperature, which is represented by the average permanent magnet surface temperature across four sensors, is transmitted via a telemetry unit. Further details on the test setup can be found in [2].

An operation point heatmap can be seen in Fig. 5. Among excitation profiles with longer constant operation evident as bright dots in the upper half-plane (motoric mode), a substantial amount of profiles denote random walks covering the full operation range in order to capture as much thermal behavior as possible.

All input and output signals are compiled in Table II. It is to be noted that only those quantities are considered as input that are readily available in most mass-produced drive applications like automotive or automation through their electronic control units and corresponding sensors, which is the reason why, for example, torque is not included here.

In all experiments, Keras [21] with a Tensorflow-backend is utilized. Experiment runtimes are reported for computations on NVIDIA GeForce GTX1080Ti graphic cards.

TABLE II
MEASURED INPUT AND TARGET PARAMETERS

Parameter name	Symbol	Parameter name	Symbol
Measured inputs		Measured target temperatures	
Ambient temperature	ϑ_a	Permanent magnet	ϑ_{PM}
Liquid coolant temperature	ϑ_c	Stator teeth	ϑ_{ST}
Actual voltage d/q-axes	u_d, u_q	Stator winding	ϑ_{SW}
Actual current d/q-axes	i_d, i_q	Stator yoke	ϑ_{SY}
Motor speed	n_{mech}		
Derived inputs			
Voltage magnitude	u_s		
Current magnitude	i_s		
Electric apparent power	S_{el}		
Joint interaction #1	$i_s \cdot \omega$		
Joint interaction #2	$S_{el} \cdot \omega$		

A. Data Preprocessing and Feature Engineering

Data representations are to be standardized mandatorily for NN training. Here, the mean is subtracted from each feature, and all observations are further scaled to exhibit unit standard deviation in the training set.

The exponentially weighted moving average (EWMA) and standard deviation (EWMS) of all input signals $\boldsymbol{x}$ are adhered to the original input space from Table II. As a result, the following two quantities are available for every signal x_t at each time step t as additional regressors

$$\mu_t = \frac{\sum_{i=0}^{t} w_i x_{t-i}}{\sum_{i=0}^{t} w_i} \quad \text{and} \quad \sigma_t = \frac{\sum_{i=0}^{t} w_i (x_i - \mu_t)^2}{\sum_{i=0}^{t} w_i} \tag{2}$$

where $w_i = (1-\alpha)^i$ with $\alpha = 2/(s+1)$ and s being an arbitrary span. Multiple, differently spanned EWMAs and EWMS' lead to various smoothed versions of all time-series and their standard deviation, which denote strong linear regressors [10]. Only four possible values for the span in EWMA and EWMS calculation are simultaneously explored in order to limit memory demand. To this end, the total amount of input quantities denotes 108 features, which is the same for all experiments.

B. Cross Validation

The mean squared error (MSE) is reported between the predicted and real temperature profile (ground truth). The full dataset is split into the following three parts: The training set for training NNs; a validation set of around five hours on which the generalization error is monitored after each iteration over the training set to apply early stopping [8]; and a seven-hours test set to eventually cross validate an NN's true generalization error on. All NN weights are optimized via the backpropagation algorithm (more specifically, Adam [22]), and the learning rate is divided by ten after there is no improvement anymore in training set loss for ten consecutive epochs. This amounts to a geometrically decreasing learning rate schedule, which is known to prevent learning from premature convergence [8].

IV. BAYESIAN OPTIMIZATION OF HYPERPARAMETERS

Most machine learning algorithms encompass not only parameters that are tuned during data fitting, but also preconfigured

TABLE III
HYPERPARAMETER SEARCH INTERVALS AND RESULTS

Search intervals			Search results			
Symbol	Hyperparameter	Interval	$\boldsymbol{\Xi}$	$\boldsymbol{\Phi}$	$\boldsymbol{\Psi}$	$\boldsymbol{\Omega}$
n_l	No. hidden layers	[1,7]	2	1	4	2
n_{units}	No. hidden units	[4, 256]	256	4	121	126
α	ℓ_2 regularization rate	$[10^{-9}, 10^{-1}]$	10^{-9}	0.1	10^{-8}	10^{-9}
β	Dropout rate	[0.2, 0.5]	0.37	0.5	0.29	0.35
η	Initial learn rate	$[10^{-7}, 10^{-2}]$	$1.4 \cdot 10^{-3}$	10^{-2}	$1.4 \cdot 10^{-4}$	10^{-4}
s_1	Span 1	[500, 1500]	500	1500	620	500
s_2	Span 2	[2000, 3000]	2204	2000	2915	2161
s_3	Span 3	[4000, 6000]	6000	4000	4487	4000
s_4	Span 4	[7000, 9000]	9000	7000	8825	8895
	RNN only					
arch	Architecture	LSTM, GRU or residual LSTM	Res. LSTM	Res. LSTM		
c_n	Grad. clip-norm	[0.25, 15]	9.4	0.25	-	-
c_{max}	Grad. clipping	$[10^{-2}, 1]$	0.076	0.01	-	-
σ_{GN}	Gradient noise	$[10^{-9}, 10^{-2}]$	10^{-9}	10^{-2}	-	-
tbptt	TBPTT length	[1, 128]	42	128	-	
	TCN only					
arch	Architecture	Plain or Residual	-	-	Residual	Residual
l_{kernel}	Kernel size	[2, 7]	-	-	6	2
w	Window size	[8, 128]	-	-	32	33

hyperparameters that affect overall training. Commonly, grid search or random search is conducted to find a well-performing set of hyperparameters. In this article, however, the search takes advantage of previously evaluated points in the hyperparameter space via Bayesian optimization [23]. Hyperparameters are optimized with the scikit-optimize framework [24] where Gaussian processes act as surrogate model.

Four independent searches are conducted. Search $\boldsymbol{\Xi}$ optimizes RNNs on stator temperatures exclusively, whereas search $\boldsymbol{\Phi}$ stands for RNNs on rotor temperatures; $\boldsymbol{\Psi}$ for TCNs on stator temperatures; and $\boldsymbol{\Omega}$ for TCNs on rotor targets. Table III compiles the considered hyperparameters, their search interval as well as their optima. In the following, not already mentioned entries are illuminated: Regularization is controlled via weight decay [25], which denotes an ℓ_2 norm penalty term on the weight matrices; dropout [26], [27], which randomly nullifies unit activations within layers; gradient clipping and normalization [28], which mitigates the exploding and vanishing gradient problem; and gradient noise on layer activations [29]. For each layer, there are n_{units} nodes. The truncated backpropagation through time (TBPTT) length is unique to RNNs, and defines the ratio of forward passes per backward pass during backpropagation.

A. Experimental Results

Table IV compiles a summary of the four searches. In absolute numbers, the found optima for both, RNNs and TCNs, achieve state-of-the-art scores on the task of temperature estimation for the stator and rotor on the designated test set. The MSE of all found optimal models is below 4k^2 and the L_∞-norm is also always below 9k, surpassing the accuracy of a recent state-of-the-art LPTN (MSE: 5.73k^2 [3]). However, this increased regression accuracy comes with magnitudes of more model parameters, which is below 50 for the mentioned LPTN.

It becomes evident that the TCN searches achieved better performing models in terms of the MSE than the RNN searches. Notably, one of the four searches ended up with an especially small topology with an only slightly degraded score compared to the other optima. This indicates that also fewer parameters in an NN offer sufficient modeling capacity, which is investigated in the following sections.

TABLE IV
SEARCH RUNTIME AND MODEL PERFORMANCE

Search	$\boldsymbol{\Xi}$	$\boldsymbol{\Phi}$	$\boldsymbol{\Psi}$	$\boldsymbol{\Omega}$
Topology	RNN	RNN	TCN	TCN
Target group	stator	rotor	stator	rotor
Total iterations	88	147	91	93
Runtime in days	> 25	> 38	> 42	> 51
Characteristics of the best model found				
Performance (MSE in K^2)	2.78	3.26	1.91	1.52
Model parameters	>850k	1.9k	>320k	>67k

V. OPTIMIZATION INTO A UNIFIED MODEL

Having analyzed the hyperparameter searches, it is further elaborated on a single model predicting all four temperatures instead of either stator or rotor quantities in an attempt to further reduce the model size. While adding a mere fourth target neuron to the final stator layer for the rotor estimation was not successful, introducing multiple heads with their own distinct layers has proven fruitful. Fig. 6 illustrates such a topology for a TCN. Note that four branches—one for each component—are likewise possible but would increase the model size significantly for a merely slight accuracy gain. Merging the stator and rotor estimation into one single model with shared layers directly reduces the required amount of parameters in contrast to two separate models that are to be run in parallel.

A. Line Search on Number of Units Per Layer

Since relatively small models of this certain topology already show strong performance, an optimal tradeoff between

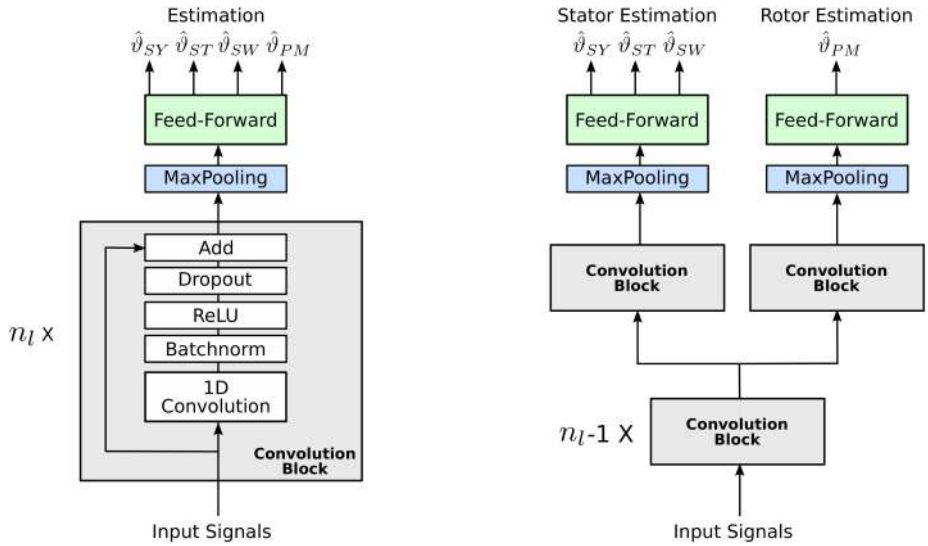

Fig. 6. Left: Single-head TCN. Right: multihead TCN with two heads.

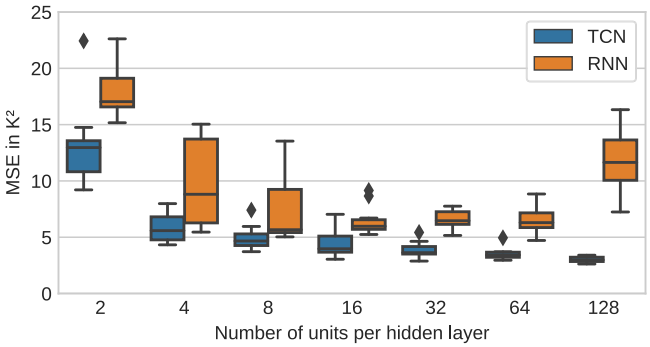

Fig. 7. Test set estimation accuracy based on the mean over all target temperatures with varying model sizes. Each topology was evaluated ten times with different random seeds.

model size and estimation performance is desirable. Pursuing an insightful Pareto-front, the number of neurons per layer is varied from 2 to 128, keeping the amount of layers fix at one joint residual hidden layer and one subsequent multihead layer, followed by a dense output layer each, as shown in Fig. 6. All other hyperparameters are kept as has been found optimal in the previous Bayesian optimization. The resulting performance curve is depicted in Fig. 7. It becomes evident that in general, again, the TCN performs better than the RNN in terms of the overall MSE as well as training consistency. Moreover, the mean estimation performance tends to improve with more hidden units per layer for the TCN, while the RNN has an optimal plateau within the contemplated interval. The deteriorating performance in case of the RNN for more than 64 units might be due to overfitting, which starts to take place with increasing model complexity. Since the regularization parameters are kept constant here, regularization strength for the RNN seems to not suffice, whereas the TCN retains its generalization ability. The optimal tradeoff between small model sizes and consistently high estimation performance appears to be at 16 hidden units per layer for the considered two-headed architecture.

B. Architecture Details and Performance

In conclusion, the optimal model architectures consist of the following.

1) TCN: One shared convolution block of 16 kernels of filter size 2 applied on a four-second sliding time window over 108 input features; two convolution blocks of 16 dilated kernels of filter size 2 each for the stator and rotor part applied on top; the stator head is completed with a three-neuron-layer for three stator signals and the rotor head with a one-neuron-layer, correspondingly. The graphical representation is best seen in Fig. 6 with $n_l = 2$.
2) RNN: One shared LSTM layer of 16 neurons applied on 108 input features per time step. The output of this layer is added with the output of a 16-neuron FNN layer applied on the current time step as residual connection. Similar to the TCN's two-head system, two dedicated residual LSTMs with 16 neurons each are applied on the shared LSTM layer that finish in a three-neuron FNN and a one-neuron FNN for stator and rotor, respectively.

This amounts to around 7 k trainable parameters for the TCN and 14 k for the RNN, which denote significant model size reductions compared to the previously found architectures, where 387 k and 852 k parameters, respectively, were necessary (see Table IV). Consequently, runtimes decrease as described in the following section. The optimal unified models achieve mean MSE scores across the four component temperatures of 3.42k^2 and 8.7k^2 for the TCN and RNN structure, respectively.

The estimation and ground truth of such composed models across the four targets is depicted in Fig. 8 for the TCN instance. It becomes evident that ϑ_{PM} and ϑ_{SY} are estimated the most precise in terms of both, the MSE and maximum absolute deviation

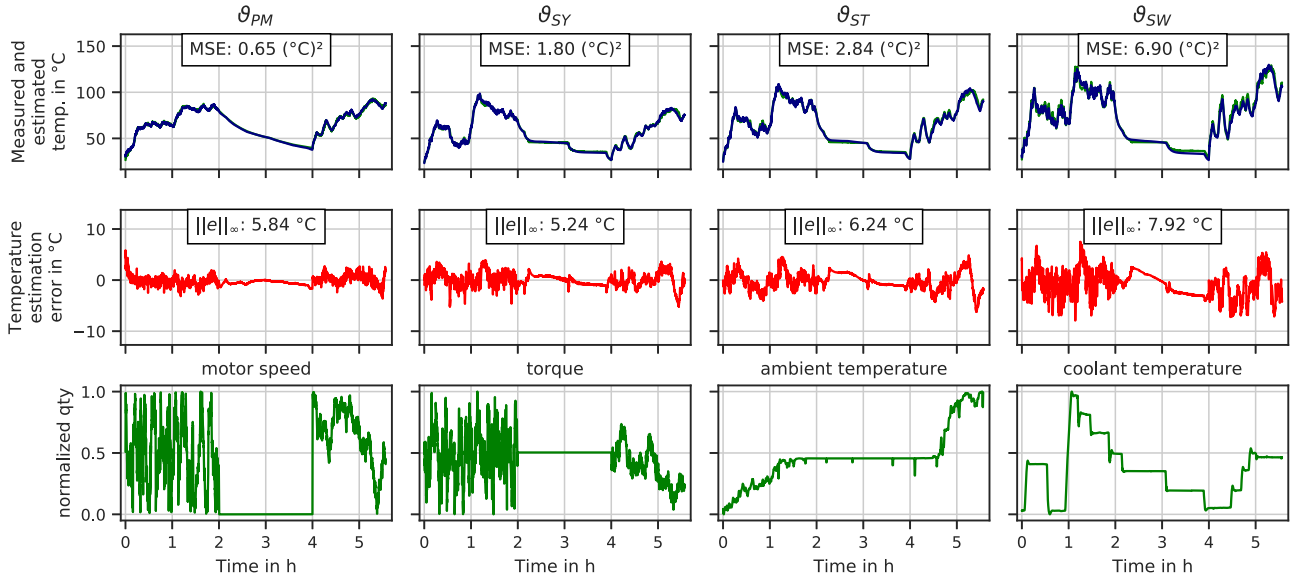

Fig. 8. Estimation on the first test set profile (out of two) with the optimal multihead TCN and 16 units per layer. Ground truth in green, prediction in blue, and the residual in red. The bottom four plots show selected input variables to highlight the high dynamics of the drive cycle.

$\|e\|_\infty$, while ϑ_{ST} is slightly weaker and ϑ_{SW} is estimated with the highest but still decent deviations.

Thus, it appears that the TCN tends to better predict those time series with a larger time constant in this particular setting. A possible reason for this is the certain choice of span values in EWMA/S generation. Reweighting target temperatures during training help to center this unbalanced accuracy across targets but comes with a worse overall MSE and $\|e\|_\infty$, therefore, it is not investigated further.

C. Runtime on Embedded-Like Hardware

Since NNs comprise substantially more parameters than conventional white- or grey-box models, all optimal topologies' runtime on embedded-like hardware has been quantified. In this article, a Raspberry Pi 3 Model B+ with Raspbian Stretch 9.8 and tensorflow 1.13 were utilized to measure runtime during inference. The models' inference runtime median across 10 000 iterations is shown in Table V along with the corresponding best model's training time on a GPU (RTX2060 Super) or CPU (i7-9800X), whichever was faster, for 100 epochs with tensorflow 2.3. Taking into account that the used dataset is sampled at 2Hz, an estimation would be necessary every half second at maximum, which is feasible even for the bigger NN topologies by a large margin.

TABLE V
RUNTIMES

Hyperparameter optimized models				
Search	Ξ	Φ	Ψ	Ω
Topology	RNN	RNN	TCN	TCN
Target group	stator	rotor	stator	rotor
Training time (GPU) in minutes	18	15	81	50
Median inference (RP3B+) time in ms	37.1	8.2	27.2	12.4
Required MFLOPs per estimate	2.24	0.004	21.9	11.7
Unified Models				
Topology	RNN		TCN	
Training time (GPU/CPU) in minutes	23		21	
Median inference (RP3B+) time in ms	14.4		8.5	
Required MFLOPs per estimate	0.03		0.07	

D. Interpreting Feature Importance With Shapley Additive Explanations (SHAP)

In the following, input parameter importance is interpreted with approximate SHAP [30]. Shapley values are a take on feature importance evaluation by identifying feature attributions to certain predictions through classic equations from cooperative game theory. SHAP values approximate Shapley values by computing them from a conditional expectation function of the original model. SHAP values themselves can be approximated for deep NNs through expected gradients, which combine SHAP and the integrated gradients algorithm [31]. Here, the full training set acted as background data to determine the model's expected estimation and the attribution is aggregated on deviations from that within estimations on the test set from Fig. 8. Violin plots of the six most impactful features in terms of their absolute SHAP value for the optimal TCN's feature attributions are shown in Fig. 9. Red and blue areas mark high and low feature values, respectively. High absolute SHAP values indicate a corresponding high impact on the actual estimation. As expected, for the stator head, the coolant and its filtered versions are among the most important input representations. More specifically, the coolant EWMA with a relatively short span of 11 min clearly dominates the attribution, which allows a

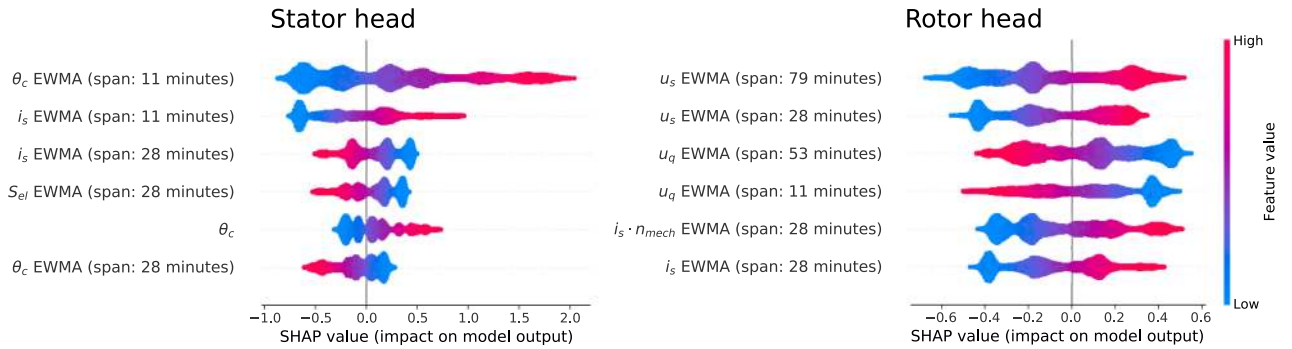

Fig. 9. Optimal multihead TCN's input attribution of the most impactful features. Left: stator head. Right: rotor head.

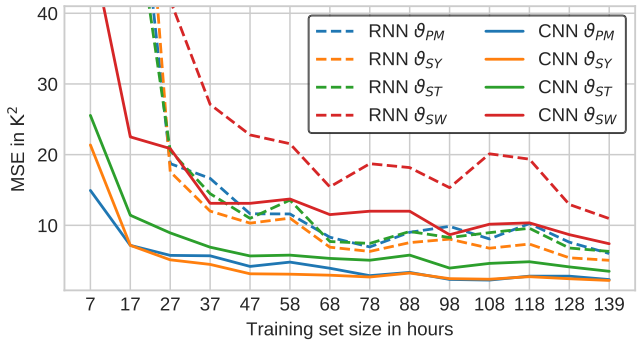

Fig. 10. MSE on the fix test set with an increasing training set size. Lines denote mean MSE across ten random subsets.

slight inference on the system's overall thermal properties. The rotor head, on the other hand, shows a more uniform distribution of absolute attributions, but still, the most impactful features are consistently among the voltage quantities. The coolant seems to have a relatively minor effect. Moreover, it becomes obvious that the rotor estimation relies more heavily on longer spans among the EWMAs than the stator head, which is explainable due to the rotor temperature's larger time constant. These observations give rise to the question, how differently filtered signals might be incorporated into an NN with dynamic spans. This remains an interesting investigation for future research.

E. Learning Curves

Industries that lack excessive amounts of data have an increased interest in ascertaining how much samples are needed to achieve a certain regression accuracy. NNs are known to exhibit a better scalability with more data than other machine learning (ML) methods, but an insufficient mapping in case of too few samples (underfitting) is likewise possible. Learning curves are plotted in Fig. 10. Here, the two-headed, optimal TCN is iteratively cross-validated against the constant-size test set whereas the available training set is virtually varied from 5% to 100% in order to assess the regression performance with random subsets of the original data. Randomness is induced by permuting the measurement session order from which the first profiles are considered ten-fold at full length to retain the time dependency. Overall, it becomes evident that the estimation performances start to converge to their minimum at around 63 h for most targets, except for the stator winding temperature estimation, which seems to not cease to benefit from more data. Note that the standard deviation, which is not depicted here, also decreases substantially with more data.

VI. Conclusion and Outlook

It has been shown that deep residual NNs that pertain to purely data-driven approaches, achieve state-of-the-art estimation performance on the task of monitoring important component temperatures in a PMSM in real-time, as measured on embedded-like hardware. Optimal architectures were found automatically during the Bayesian optimization of topology-defining hyperparameters, and manual model size reductions were achieved including the unification of all targets' estimates in one model. A method from representation importance literature for attributing model estimates to their most impactful input representation has been applied and shown useful. Eventually, it has been evaluated that decent performances are achievable also with around 60 hours of records, which denotes half the available data.

The presented data-driven modeling technique is by design of general form and can be easily adapted to different motor types, temperature targets (i.e., motor components) or even neighboring domains (e.g., battery systems) without the necessity of domain expert knowledge, provided that there exists a statistical relationship between input and output signals.

It is still an open question how well NNs may generalize across different motors from the same or even different manufacturers (transfer learning). Investigating this requires an enormously large dataset exhibiting this diversity, and is yet to be recorded.

References

[1] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Deep residual convolutional and recurrent neural networks for temperature estimation in permanent magnet synchronous motors," in *Proc. IEEE Int. Elect. Mach. Drives Conf.*, 2019, pp. 1439–1446.

[2] O. Wallscheid and J. Böcker, "Global identification of a low-order lumped-parameter thermal network for permanent magnet synchronous motors," *IEEE Trans. Energy Convers.*, vol. 31, no. 1, pp. 354–365, Mar. 2016.

[3] E. Gedlu, O. Wallscheid, and J. Böcker, "Permanent magnet synchronous machine temperature estimation using low-order lumped-parameter thermal network with extended iron loss model," 2020, *TechRxiv11671401*.

[4] O. Wallscheid, A. Specht, and J. Böcker, "Observing the permanent-magnet temperature of synchronous motors based on electrical fundamental wave model quantities," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 3921–3929, May 2017.
[5] S. D. Wilson, P. Stewart, and B. P. Taylor, "Methods of resistance estimation in permanent magnet synchronous motors for real-time thermal management," *IEEE Trans. Energy Convers.*, vol. 25, no. 3, pp. 698–707, Sep. 2010.
[6] M. Ganchev, C. Kral, H. Oberguggenberger, and T. Wolbank, "Sensorless rotor temperature estimation of permanent magnet synchronous motor," in *Proc. Ind. Electron. Conf. Proc.*, pp. 2018–2023, 2011.
[7] O. Wallscheid, T. Huber, W. Peters, and J. Böcker, "Real-time capable methods to determine the magnet temperature of permanent magnet synchronous motors—A review," in *Proc. 40th Annu. Conf. IEEE Ind. Electron. Soc.*, 2014, pp. 811–818.
[8] I. A. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/
[9] S. Bai, J. Z. Kolter, and V. Koltun, "Empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv 1803.01271* .
[10] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Empirical evaluation of exponentially weighted moving averages for simple linear thermal modeling of permanent magnet synchronous machines," in *Proc. 28th Int. Symp. Ind. Electron.*, 2019, pp. 318–323.
[11] W. Kirchgässner, "Deep-PMSM," 2020. [Online]. Available: https://github.com/upb-lea/deep-pmsm
[12] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *Proc. Int. Joint Conf. Neural Netw.*, 2005, vol. 4, no. 5, pp. 2047–2052.
[13] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," 2014, *arXiv 1410.8206*.
[14] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv: 1506.00019*.
[15] F. A. Gers and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw.*, 1999, vol. 2, pp. 1–19.
[16] K. Cho *et al.*,, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv: 1406.1078*.
[17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.
[18] 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.
[19] A. van den Oord *et al.*, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.
[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.
[21] F. Chollet and Others, "Keras," 2015. [Online]. Available: https://keras.io
[22] D. P. Kingma and J. Ba, "Adam: A. Method for Stochastic Optimization," 2014, *arXiv:1412.6980*.
[23] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A. review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.
[24] T. Head and Others, "Scikit-Optimize," 2018. [Online]. Available: https://scikit-optimize.github.io/
[25] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," *Adv. Neural Inf. Process. Syst.*, vol. 4, pp. 950–957, 1992.
[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A. simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
[27] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," 2015, *arXiv :1512.05287*.
[28] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," in *Proc. 30th Int. Conf. Mach. Learn.*, pp. 1310–1318, 2012.
[29] A. Neelakantan *et al.*, "Adding gradient noise improves learning for very deep networks," 2015, *arXiv :1511.06807*.
[30] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.
[31] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," 2017, *arXiv :1703.01365*.

Wilhelm Kirchgässner received the B.Sc. degree from the OWL University of Applied Sciences and Arts, Lemgo, Germany, in 2013, and the M.Sc. degree (honours) in electrical engineering in 2016 from the Paderborn University, Paderborn, Germany, where he is currently working toward the doctoral degree in electrical engineering.

He is currently working as a Research Associate for the Department of Power Electronics and Electrical Drives, Paderborn University. His research focuses on applying machine learning to electrical drive technologies, in particular thermal modeling of highly utilised traction drives from observed data. He was a Software Engineer with the Continental developing automotive applications in embedded systems from 2016 to 2018.

Oliver Wallscheid (Member, IEEE) received the bachelor's and master's degrees (hons.) in industrial engineering and the doctorate degree (hons.) in electrical engineering from Paderborn University, Paderborn, Germany, in 2010, 2012, and 2017, respectively.

Since then, he has worked as a Senior Research Fellow for the Department of Power Electronics and Electrical Drives, Paderborn University. He is currently serving as acting Professor of the Automatic Control Department, Paderborn University. His research includes system identification and intelligent control methods for technical systems in the areas of power electronics, drives and decentralized grids. His recent work focuses on hybrid methods utilizing machine learning (black box approaches) and classical engineering techniques (white box approaches).

Joachim Böcker (Senior Member, IEEE) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from the Berlin University of Technology, Berlin, Germany, in 1982 and 1988, respectively.

From 1988 to 2001, he was with the Allgemeine Elektrizitäts-Gesellschaft (AEG) and Daimler Research as the Head of the Control Engineering Team of the Electrical Drive Systems Laboratory. In 2001, he started his own business in the area of control engineering, electrical drives, and power electronics. He is currently Head of the Department with Power Electronics and Electrical Drives, Paderborn University, Paderborn, Germany, where he has been a Professor since 2003. His research interests include electrical drives, particularly for EVs and hybrid electric vehicles (HEVs), energy management strategies for vehicles and smart grids, and converters for power supplies, EV chargers, and renewables.