IEEE Open Journal of
Intelligent Transportation
Systems

Received 16 March 2022; revised 3 May 2022; accepted 19 May 2022. Date of publication 23 May 2022; date of current version 3 June 2022.

Digital Object Identifier 10.1109/OJITS.2022.3177007

A Method of Developing Quantile Convolutional Neural Networks for Electric Vehicle Battery Temperature Prediction Trained on Cross-Domain Data

ANDREAS M. BILLERT, MICHAEL FREY, AND FRANK GAUTERIN

Institute of Vehicle System Technology, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

CORRESPONDING AUTHOR: A. M. BILLERT (e-mail: andreas.billert@partner.kit.edu)

This work was supported by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

ABSTRACT The energy consumption caused by battery thermal management of electric vehicles can be reduced using predictive control. A predictive controller needs a prediction model of the battery temperature, for example for different battery cooling and heating thresholds. In the proposed method, cross-domain data from simulation, vehicle fleet and weather stations were analyzed and processed as training data for a Convolutional Neural Network (CNN). The CNN took data from previous road segments and predictions for following road segments as input and predicted the change in battery temperature as quantile sequences over a prediction horizon. Properties of the collected cross-domain data sets were analyzed and considered during preprocessing, before 150 models were trained, of which the best performing model was further analyzed. Point-forecast metrics and quantile-related metrics were used for model comparison and evaluation. For example, the median prediction achieved a mean absolute error (MAE) of 0.27 °C and the true values were below the median prediction in 47% of the test data. Possible improvements of the method such as increasing data size, using more complex architectures as well as optimizing the horizon sizes were discussed. In conclusion, the method was able to well predict battery temperatures for different battery cooling thresholds.

INDEX TERMS Battery temperature, convolutional neural network, cross-domain data, machine learning, quantile forecasting.

NOMENCLATURE

LATIN SYMBOLS

A	Surface
c	Heat capacity
d	Distance
F	Force
h	Heat transfer coefficient
I	Current
k	Horizon size
L	Loss
m	Mass
n	Number (quantity)
P	Power
p	Occurrence
$\dot{Q}$	Heat transfer rate
R	Inner resistance
s	Drive distance
T	Temperature
t	Time
U	Voltage
v	Velocity
y	True value
$\hat{y}$	Predicted value.

GREEK SYMBOLS

α	Weight
δ	Partial derivative

The review of this article was arranged by Associate Editor Prof. Chi-Hua Chen.

Δ	Difference
η	Efficiency
μ	Quantile weight
ψ	Sharpness
σ	Standard deviation, also related to as smoothness (of quantile)
θ	Slope angle.

SUBSCRIPTS

0.5	Indication for 0.5 quantile also known as median
a	Ambient
avg	Average
b	Battery
c	Coolant
cons	Consumer
cool,start	Start of cooling
cwq	Constrained weighted quantiles
d,abs	Absolute difference
em	Electric machine
fit	Fitting
gen	Generated
i	Index i, related to horizon step
j	Index j, related to quantile
μ	Quantile weight
max	Maximum
MSE	Mean Squared Error
ocv	Open circuit voltage
p	Specific value (of heat capacity)
pb	Pinball
q	Quantile
res	Total drive resistance
s	Surrounding
tm	Thermal management
total	Total
vehicle	Vehicle
ws	Weather station.

SUPERSCRIPTS

q	Quantile.

ABBREVIATIONS AND ACRONYMS

ADAM	Adaptive Moment Estimation
BEV	Battery Electric Vehicles
BTMS	Battery Thermal Management System
CNN	Convolutional Neural Network
CORS	Crossover Rate Score
DOE	Design of Experiments
DPMA	Driving-Profile-Map-Attributes
DWD	Deutscher Wetterdienst (German weather service)
FFT	Fast Fourier Transformation
MAE	Mean Absolute Error
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MSE	Mean Squared Error
NN	Neural Network
QL	Qualifier
R^2	Coefficient of Determination
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RUS	Random Undersampling
SOC	State of Charge
SOH	State of Health
WS	Winkler Score.

I. INTRODUCTION

BATTERY thermal management has a significant impact on energy consumption thus on the range of battery electric vehicles (BEV) [1]. An efficient battery thermal management system (BTMS) has to find the optimal balance between low energy consumption and low impact of the battery temperature T_b on battery aging and power derating [2], [3]. For instance, higher battery temperatures result in accelerated aging [4], [5]. A battery temperature threshold for battery cooling is commonly used to avoid high battery temperatures [6], [7], [8].

Battery cooling might not be necessary if the battery temperature increase for the following drive horizon is not high thus does not significantly increase aging (e.g., due to a low power request). A shift to a higher battery cooling threshold avoids cooling and leads to less energy consumption. For example, a dynamic uphill drive can lead to an increase of T_b, such that the battery cooling threshold $T_{b,cool,start}$ would be reached before reaching the peak of the mountain. When driving downhill, even without active battery cooling, the increase in T_b is reduced due to the lower power request. In this case, battery cooling may be avoided already before reaching the mountain peak if T_b will not reach significant higher values without cooling. This can be implemented using a predictive controller instead of fixed thresholds. Such a controller needs the prediction of T_b for the following drive horizon for different $T_{b,cool,start}$ to choose the optimal value for $T_{b,cool,start}$.

The prediction of T_b for model predictive control can be done using a physics-based model, often combined with empirically determined parameters [3], [7], [9], [10], [11], [12], [13], [14]. Knowledge about the uncertainty of the prediction of T_b is important due to the non-linearity of the underlying physical processes and the uncertainty of the forecast of the route ahead. Including information about the prediction uncertainty can improve model predictive control [7]. Model predictive control of a BTMS can be extended by Neural Networks (NN), e.g., for the prediction of vehicle speed as input [2]. NNs can also be applied as soft sensors for State of Charge (SOC), State of Health (SOH) and power loss estimation, to take into account the complex modeling of battery cells and related heat transfers [15], [16], [17]. The trained NN performed better than the compared regression models [15]. The application of Deep Learning using Convolutional Neural Networks (CNN) provided good performance for SOH estimation in [18]. Predictions of the Nusselt number of a BTMS using NNs with more than one hidden layer also result in better accuracy than using a single-layer NN [19].

The nonlinear behavior of BTMS motivates the usage of NNs to directly predict the battery temperature T_b as part of a predictive controller, as done by Liu and Zhang [20], [21]. They designed an air-based BTMS with different modes of coolant flow (U-, Z- and J-type) and used simulations as training data. Park and Kim [22] implemented NNs for subsystems of BTMS which are connected according to the physical topology of the BTMS. Their NNs can be used for different BTMS modes, which is important for predictive control. Besides environmental quantities (e.g., speed), physical quantities of the BTMS components are used as inputs, such as mass flow rate and compressor power. Park and Kim [22] used measurements of driving profiles based on two drive cycles as training data. The prediction of T_b using NNs showed to be accurate and robust to external noise in [23], [24], [25]. Predictive control using NNs is also applied in other fields, such as power control of vehicles with combustion engine [26], motion planning for autonomous vehicles [27] and thermal or energy management of buildings [28], [29], [30], also using weather forecast data as input for Reinforcement Learning [31]. An integration of predictive BTMS (e.g., using NN for its predictions) into a cloud-based system [32] may be advantageous.

The development of Machine Learning (ML) models can be improved using Big Data [33], [34]. More (non-redundant) training data enable the development of more complex data-driven NN architectures (e.g., Deep Neural Networks) which can represent complex relations and lead to better performance [35], [36], [37]. Noise and redundancy in data require several steps of data analysis and preprocessing, e.g., data cleaning and feature engineering [33], [38], [39]. According to L'Heureux *et al.* [38] and Zheng [40], the value of a ML model can be improved using heterogeneous data from different domains (also called cross-domain data by Zheng), which requires suitable preprocessing and fusion.

An application of cross-domain data is the enrichment of ML by simulations [41], [42]. Simulation data can be used as training data to represent the causality of a particular behavior of the system. For instance, it can enrich measurement-based training data of a vehicle fleet that lack the desired system behavior, such as a novel variation of battery cooling and heating thresholds, unless additional, resource-expensive experiments are conducted. On the contrary, vehicle fleet data can provide a large amount of data in the domain of model application with energy demand under real driving conditions, instead of simulations based on artificial drive profiles [43]. Furthermore, vehicle fleet data can be joined by data from weather stations to improve the predictive control of a BTMS [44], which takes into account the impact of humidity and solar radiation on heat transfer and thermal loads [45], [46].

Predicting a quantity for a following sequence can be done by different types of NNs. Time-series properties and forecasting methods using NNs are described in [47], [48], [49], [50], [51]. Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN, e.g., Long-Short-Term-Memory) and Transformer models are stated as suitable for time-series prediction. CNNs can learn representations of locally invariant patterns with features in different hierarchical levels, which is important when dealing with non-linear dependencies within a large amount of multivariate, sequenced data [48], [52]. This cannot be achieved by classic (regression) ML models. CNNs can be designed as Multiple Input Multiple Output (MIMO) models, such that they are able to model multi-step multi-horizon forecasts, which provide better performance than comparable forecast methods (e.g., single-step and recursive strategies) [53]. CNNs can be improved by multivariate or multi-scale architectures [54], [55].

The practical value of a ML model (e.g., for predictive control) can be extended by predicting uncertainties, for example using probability distributions [50], [56]. In case of uncertainties that do not follow a Gaussian distribution, Quantile Regression NNs can be used to predict quantiles [57]. Quantile Regression can be applied on time-series forecasting with different types of NNs (including CNNs and RNNs), and can be improved using a custom quantile loss with trainable and constrained quantile weights (L_{cwq}) [58]. A combination of RNN with quantile regression for multi-horizon forecasting is given in [59] and provides better prediction intervals than the prediction of a Log-Gaussian distribution. Replacing the RNN by a CNN further improved the accuracy.

In this work, a method of developing CNNs with quantile predictions of the battery temperature for different battery cooling and heating thresholds is proposed. CNNs are used instead of classic regression models because of their ability to model non-linear physical behavior with locally invariant patterns as multi-step multi-horizon prediction of quantiles, which provide the prediction uncertainty for later usage in a predictive control of BTMS. The loss function includes L_{cwq} from [58] and the architecture provides two input channels, one for data from previous drive segments of the same drive (history channel) and one for predictions of route and weather (foresight channel). Training, validation and test data are composed of cross-domain data. A simulation data set provides the effect of different battery cooling and heating thresholds. A vehicle fleet data set provides drives under real road conditions and non-artificial profiles. It is joined by public data from a weather database to take into account the effect of humidity and radiation which are originally not included in the vehicle fleet data. All data sets are analyzed and processed before model training and evaluation. They consist of data points sampled by driven distance as segments of 250 m. Models with different hyperparameters are trained and compared using point-forecast and quantile-related metrics. Predictions of the best performing model

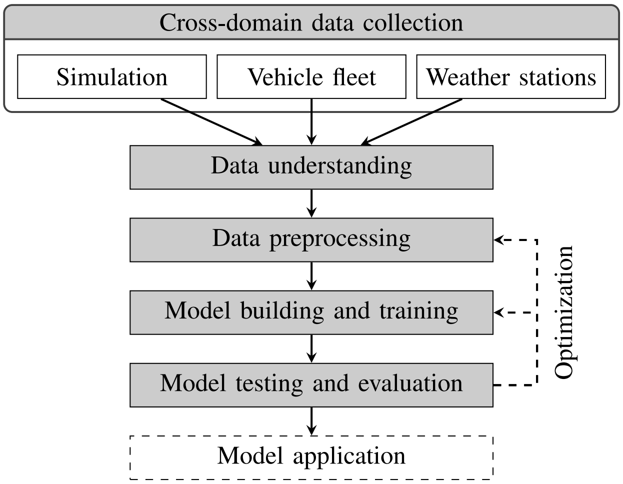


FIGURE 1. Machine Learning pipeline with cross-domain data as input.

on test data are further analyzed and discussed. Potentials and limitations of the method are presented and an outlook on improvements is given, such that the method can be further optimized in future works to provide accurate prediction models for a predictive control of BTMS.

II. METHOD

This chapter introduces the proposed method of developing prediction models for the battery temperature of electric vehicles. The steps of the method are structured in a pipeline as shown in Fig. 1. The collection of data from different domains is followed by the step of data understanding, in which plausibility, redundancies and limitations of the collected data are analyzed. Possible errors, implausibilities or gaps in the collected data need to be identified and then processed in the following step called data preprocessing. This step includes the preparation of data, such as feature engineering and data balancing, for the next step of model building and training. Trained models are tested and evaluated, which can lead to several optimization iterations to obtain a model that meets certain performance requirements necessary for its application. Each step will be further explained in the remain of this chapter. Examples provide a detailed view on the variation of cooling thresholds but can be similarly related to heating thresholds.

A. CROSS-DOMAIN DATA COLLECTION

The need for using cross-domain data is given by individual limitations of each data source. On the one hand, simulation data can provide the effects of a novel variation of battery cooling and heating thresholds, which are not present in vehicle fleet data. On the other hand, the available simulation model is simplified and therefore does not cover all physical dependencies. In contrast to that, vehicle fleet data provide customer-related drive profiles rather than artificial drive cycles, under real driving conditions and with all external influences present in the data.

In preparation of data collection, the required physical quantities need to be specified. For a change in battery temperature T_b over distance s, the physical dependencies are described by (1). They include the transfer rate of heat generated by the battery $\dot{Q}_{b,gen}$, heat transferred to the surroundings $\dot{Q}_s$ and heat transferred from or to the thermal management system $\dot{Q}_{tm}$. The change in T_b over s is also dependent on time t, velocity v, battery mass m_b and specific heat capacity of the battery $c_{p,b}$.

$$\frac{\partial T_b(s)}{\partial s} \approx \frac{\partial T_b(t)}{v \cdot \partial t} = \frac{\dot{Q}_{b,gen} + \dot{Q}_s + \dot{Q}_{tm}}{v \cdot m_b \cdot c_{p,b}} \tag{1}$$

The set of physical quantities that describe the thermal behavior of the battery can be obtained with the underlying equations for each of the three heat generation and transfer quantities $\dot{Q}$ (see Appendix A). The quantities are recorded on a distance base, both for simulation and vehicle fleet data.

Simulation data are obtained according to a Design of Experiments (DOE). It varies in cooling and heating thresholds, velocity and slope profile (e.g., from cycles), ambient temperature and initial battery temperature. Instead of obtaining a homogeneous variation, the focus of the DOE is on combinations that lead to differences in system behavior caused by battery cooling and heating thresholds.

Vehicle fleet data are collected and aggregated to distance-based segments, before they are sent to the backend database. There, the data set is joined by data from weather stations such as humidity and radiation. It is a one-on-one join using position, date and time between the vehicle and the closest active weather station with a location below 1000 m above sea level (to neglect weather stations located on mountains).

B. DATA UNDERSTANDING

Before using data sets for the training of ML models, their suitability, added value and limitations need to be understood. Average battery temperature and overall energy consumption are compared between simulations with different cooling and heating thresholds. The shift of start and end segment of battery cooling and heating is analyzed by cross-correlation of the battery temperature between simulations. Redundancies between physical quantities are identified using Pearson correlation. Inaccuracies due to distance-based segmentation are analyzed.

Vehicle fleet data may contain data gaps or implausible data due to sensor inaccuracies, sensor malfunctions or errors in the data collection process. This can be identified by data plausibility checks, for example considering value ranges or value monotony. For prototype vehicles, an identification of drives on a test track or test rig is done by position and velocity profile to filter them because of their artificial profiles. A drive on a test rig is identified if the position does not change or the velocity profile shows periodic behavior, which is analyzed using Fast Fourier Transformation (FFT). Occurrence distributions of values and rates of the collected data, especially of the battery temperature, can be used to understand the covered range available for the ML model. The analysis of Pearson correlation between signals is also possible for vehicle data to identify dependencies and possible redundancies.

C. DATA PREPROCESSING

Faulty and implausible data, which have been detected in the step of data understanding, need to be addressed during data preprocessing. Dependent on the physical quantity and the number of faulty data points, available and correct values are interpolated or placeholder values are used. Signals that contain placeholder values are extended by a qualifier (QL) to provide the information if a value is correct ($QL = 1$) or a placeholder ($QL = 0$). Battery temperature, ambient temperature and State of Charge (SOC) of the vehicle fleet data are processed with a moving average of five segments to smoothen the discrete value steps provided by the sensors.

The fusion of all collected data is conducted feature-based. Vehicle fleet data and simulation data are appended, such that the resulting table contains all segments of real and simulated drives as rows. Its columns are non-redundant physical quantities identified in the previous steps, including variants from mathematical operations and energetic Driving-Profile-Map-Attributes (DPMA) as defined and used in [60] and [61]. Possible redundancies due to these variants are checked using Pearson-correlation. The conducted feature engineering directly provides physically relevant quantities for ML training.

All segments are extended by following and previous values in a third dimension of the table, representing a history and a foresight horizon. The foresight horizon contains both foresight input data (e.g., estimated speed, slope, temperature) and the labels the model has to predict (i.e., the change in battery temperature). Hence, in this work, the prediction horizon equals the horizon of the foresight input data. An additional qualifier provides the information if a horizon segment exceeds the start or end of the drive. Each feature is centered and normalized into the range $[-1, 1]$.

All drives are assigned group numbers for a split into train, validation and test data. Data imbalance is addressed by Random Undersampling (RUS) as in [62], but according to an empirically defined occurrence curve. For training, every second data row is disregarded since neighboring segments contain a high overlap of history and forecast horizon data. This undersampling reduces overfitting and training time.

D. MODEL BUILDING AND TRAINING

Model building and training incorporates the choice of model architecture and training parameters. The considered architecture is shown in Fig. 2. It comprises of two input channels, one for each input horizon (history and foresight, see Section II-C). Both input channels include hidden layers prior to concatenation. After more hidden layers, one output channel per quantile is added. Each quantile output channel predicts the change in battery temperature as multi-step over a distance-based prediction horizon with respect to the battery temperature at the position of the prediction. The choice and shape of inputs and outputs is further subject of Section III-C. The model is built and trained using the TensorFlow/Keras package in version 2.5.0 for Python.

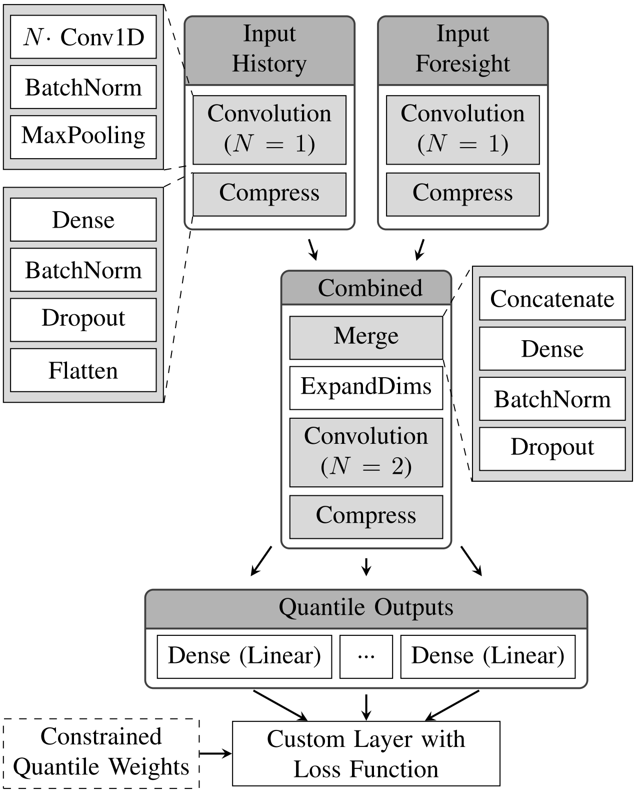


FIGURE 2. General architecture of the neural network. Repeating combinations of layers are presented as blocks (shaded rectangles), with custom names and a magnifier view for each block type. The layer names are given according to the used TensorFlow/Keras package, with BatchNormalization and MaxPooling1D shortened to BatchNorm and MaxPooling respectively. The number of Conv1D layers per convolution block is given by *N*. Inputs and outputs are further discussed in Section III-C, a complete list of input features is given in Appendix B.

To constrain the number of variants, the following architectural design choices are made: The choice of type and number of hidden layers is fixed. The number of layers is kept at a minimum considering the size of collected training data (see Section III). It will be increased in future works when more data is available. The number of nodes in the last layer before the concatenate layer is kept low, since otherwise the biggest share of trainable model parameters is located in the concatenate layer. Batch normalization layers are placed before dropout layers, as discussed in [63], [64], [65]. The number of nodes of the hidden layers is chosen decreasing from input to output channels. Hence, complex patterns and correlations of input features can be captured by a high number of nodes, whereas patterns of the predicted output quantity (i.e., T_b) can be described with fewer nodes, which results in a reduction of model parameters and training time with comparable prediction performance [66]. The relation of layer nodes is considered using a scaling factor, with a varying base number of nodes for the model.

The loss function is added as part of a final, customized layer. It is based on the work of Lopez-Martin *et al.* [58] who developed a quantile loss L_{cwq} (2) for time-series forecasting with quantile regression. It uses the pinball loss L_{pb} (3) between the predicted values $\hat{y}_i$ for the quantiles q_j and the true values y_i for a horizon size k and provides n_μ trainable quantile weights μ_j, including constraints which minimize quantile-crossing and improve point- and probablisitic forecasts. The constraints include softmax function and symmetry around the 50 percent quantile $q_{0.5}$ (median). A fixed initialization and a fixed order, descending from median to outer quantiles, are added in this work. This ensures that outer quantiles do not get higher weights than inner quantiles which can result in optimizing only part of the desired quantile output as pointed out by [58] for the unconstrained case. During training, the custom layer takes the true values as an additional input.

$$L_{cwq} = \frac{1}{k\left(2n_\mu + 1\right)} \sum_{j=0}^{2n_\mu} \sum_{i=1}^{k} \mu_j \cdot L_{pb}(i,j) \tag{2}$$

$$L_{pb}(i,j) = max\left[\left(q_j - 1\right)\left(y_i - \hat{y}_i^{q_j}\right), q_j\left(y_i - \hat{y}_i^{q_j}\right)\right] \tag{3}$$

The loss function is extended in this work by the Mean Squared Error (MSE) to enforce higher weight on the median forecast, independent of the distribution of quantile weights. The total loss L_{total} (4) contains an additional weight α_{MSE} as fixed hyperparameter.

$$L_{total} = (1 - \alpha_{MSE}) \cdot L_{cwq} + \frac{\alpha_{MSE}}{k} \cdot \sum_{i=1}^{k} \left(y_i - \hat{y}_i^{q_{0.5}}\right)^2 \tag{4}$$

Training parameters are related to the choice of optimizer, learning rate, mini-batch size and the patience value of early stopping. Learning rate and mini-batch size are varied as part of the model optimization as well. An overview of all optimization parameters is given in Table 1. Properties of input data, horizons and the general architecture are fixed for this work. This includes the usage of the actual collected data as forecast input data without noise which could occur due to inaccuracies of forecasting e.g., velocity and ambient temperature. The relation of nodes between layers is also fixed, with a variable base number as described above. A list of fixed parameters and their values is provided in Appendix C.

TABLE 1. Design parameters for model building and training.

Category	Parameter	Consideration in this work
Input data	Selection of data (data source, data balancing)	Fixed
	Selection of features (physical quantities, qualifiers)	Fixed
	Noise of forecast data	Fixed (no noise)
Horizons	Size of history and forecast horizon (input) and prediction horizon (output)	Fixed
Architecture	Order, type, number and parameters (e.g. nodes, kernel) of layers	• Fixed: order, type, number, kernel, relation of nodes between layers • Varied: base node number
	Regularization: dropout rate, batchnorm., max-norm.	• Fixed: batchnorm. and max-norm. • Varied: dropout rate
	Custom loss function: α	Varied
Training	Learning rate	Varied
	Mini-batch size	Varied
	Early stop patience	Fixed

E. MODEL TESTING AND EVALUATION

Any model resulting from the previous steps is tested and evaluated. Test data used for this purpose consist of drives of the vehicle fleet which have not been used as training or validation data, and simulations of drive profiles different to the profiles used as training or validation data. Evaluation is based on the following metrics: resulting total loss (4), point-forecast metrics and quantile-related metrics. Point-forecast metrics can be applied individually for each quantile, and are most important for the evaluation of the 50 percent quantile (median) as point-forecast. They include Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and the Coefficient of Determination (R^2) between prediction and true values. A prediction error distribution can be included for a detailed analysis. The standard deviation σ of the partial derivative of the prediction of each quantile q_j with respect to the distance per segment δs (250 m) is used as a metric to describe the smoothness σ_{q_j} of the predictions (5).

$$Smoothness\left(q_j\right) = \sigma_{q_j} = \sigma\left(\frac{\delta\hat{y}^{q_j}}{\delta s}\right) \tag{5}$$

Quantile-related metrics describe the consistency of quantile predictions (e.g., quantile-crossing, quantile-related probabilities). The relative occurrence of values within quantile $p(q_j)$ and quantile interval $p(q_j, 1 - q_j)$ is calculated and the absolute differences are taken as evaluation metrics (denoted by subscript d, abs). The metrics Winkler Score (WS), Crossover Rate Score (CORS) and Sharpness (denoted as ψ) are based on the definition in [58].

All metrics are calculated per quantile or quantile interval. The average metric values over all quantiles are taken for a comparison of models. Better performance is expected with lower values for each metric. For R^2, only the median prediction is taken for model comparison and higher values are better, with 1.0 being the upper limit. Based on the total loss and the evaluation metrics for predictions on the test data set, models with variation in hyperparameters can be compared and ranked. Dependent on the target application, the best suitable model can be selected according to the ranking. An exemplary analysis of predictions on test data can be used to explore the possibilities and limitations of using the model in target applications. The impact of different thresholds on quantile predictions of the battery temperature needs to be examined for the model's suitability for predictive control.

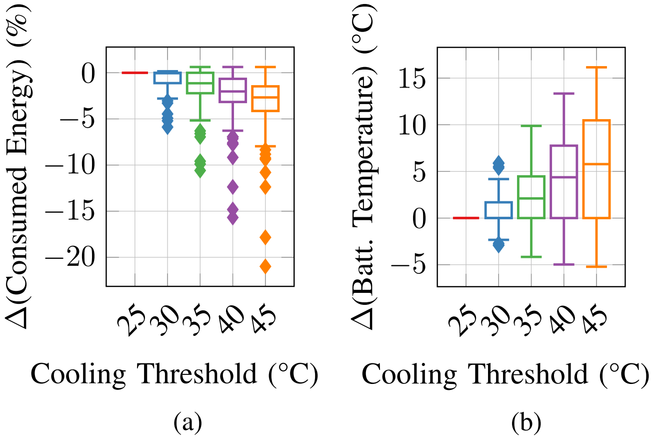


FIGURE 3. Effect of battery cooling thresholds on overall energy consumption (a) and on average battery temperature (b) in reference to a threshold of 25 °C.

III. RESULTS

This section provides the results of applying the proposed method for a variation of battery cooling thresholds. It is structured according to the steps given in the previous section.

A. CROSS DOMAIN DATA COLLECTION

The simulation data set used as training and validation data in this work consists of 844 simulations of drive profiles with one hour drive duration for five different cooling thresholds. Additional 16 simulations are used as test data. Vehicle fleet data are collected from a prototype fleet of 16 vehicles between March 2021 and October 2021. Drives that do not contain battery temperature data (e.g., due to malfunctions of the data collector script), with implausible mileage or with a drive distance of less than 5 km are neglected. This results in 1504 drives remaining for further analysis and processing in the next steps. They are joined by publicly accessible data from 147 weather stations of the German Weather Service (DWD) [67].

B. DATA UNDERSTANDING

The simulation data set is obtained to cover the effect of different cooling thresholds for the battery on its temperature and the overall energy consumption. This effect is shown in Fig. 3. An increase of the cooling threshold in steps of 5 °C results in a higher, average battery temperature and a lower energy consumption. Fig. 4 depicts cross-correlations of the battery temperature between each threshold and the reference threshold. It shows the tendency of the cooling start shifting to later segments for higher cooling thresholds. Both diagrams reveal that the simulation data cover the effect of cooling thresholds on energy consumption and battery temperature as required. A Pearson correlation of the battery current I_b is given in Table 2 as an example. The results show high correlations with other quantities, such that the battery current is not considered as a feature. From the listed quantities, only total driving torque and acceleration pedal are kept as features. Inaccuracies in energy and battery temperature due to distance-based sampling of simulation data (from 0.2 Hz to 250 m segments) are shown in Fig. 5. The observed inaccuracies are in acceptable ranges and there are no significant differences between the different cooling thresholds.

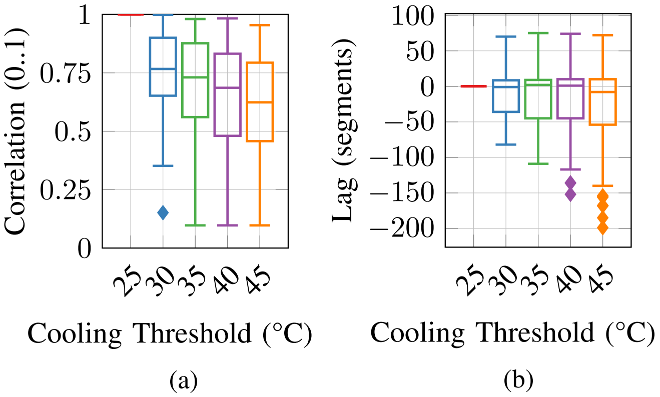


FIGURE 4. Effect of battery cooling thresholds on start of cooling represented by cross-correlation (a) with segment lag (b) of T_b in reference to a threshold of 25 °C. A negative lag means a later cooling start.

TABLE 2. Pearson-correlation of the battery current to other quantities (filtered for correlations greater than 0.9).

Simulation parameter	Battery current
Battery cell current	1
Battery power	0.998
Inverter current	0.996
Inverter power	0.993
Electric machine electric power	0.993
Electric machine torque	0.948
Torque of drive axle	0.947
Total driving torque	0.947
Acceleration pedal	0.921

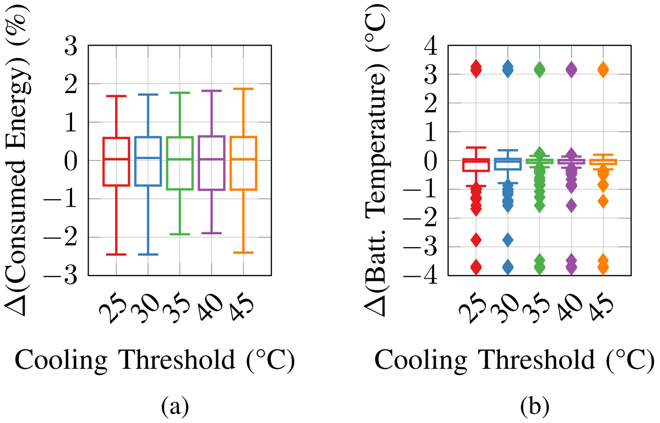


FIGURE 5. Inaccuracies of overall energy consumption (a) and average battery temperature (b) over battery cooling thresholds due to distance-based sampling.

Data sets from the vehicle fleet are analyzed with a focus on possible sensor signal inaccuracies and the occurrence of changes in battery temperature over different distance horizons. In the collected data, the battery temperature T_b is available in discrete steps of 1 K. Fig. 6 shows a distribution of the differences in T_b of segments with varying distance

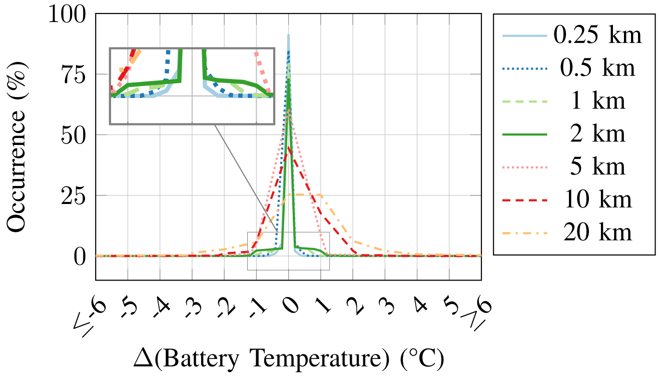


FIGURE 6. Envelope curves of the occurrence of differences in T_b between segments with different distance in between, obtained from the collected vehicle fleet data.

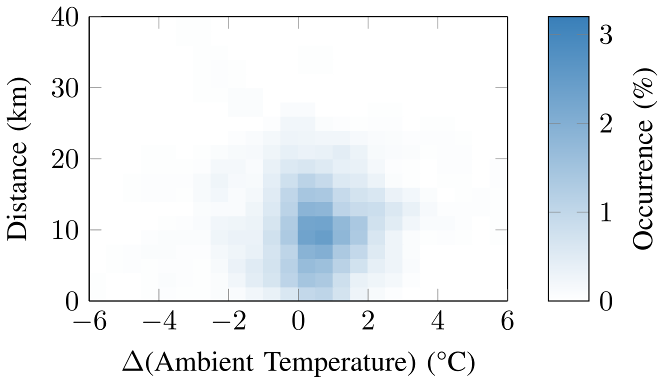


FIGURE 7. Distance between vehicle and closest weather station d_{ws} versus difference in measured ambient temperature ΔT_a (6). Data with $d_{ws} \geq 74$ km are excluded in this diagram.

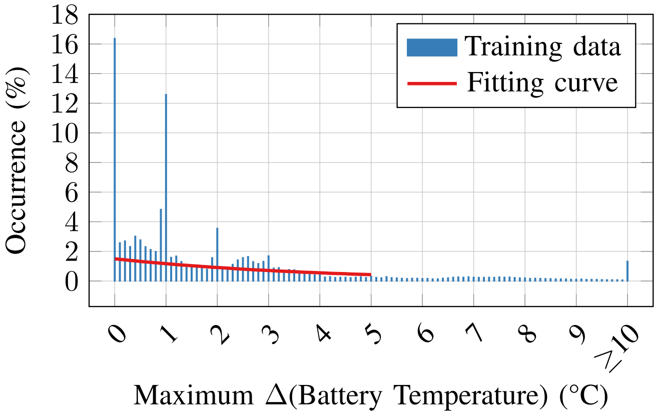


FIGURE 8. Occurrence of maximum absolute differences in T_b (7) over the forecast horizon of the collected training data points (segments) and fitting curve as defined in (8) with (9). High occurrence of discrete values (at 0 °C, 1 °C , 2 °C) is a result of sensor discretization.

in between. An increasing distance results in a flatter and wider distribution which means a higher occurrence of higher absolute differences in T_b. An increase in battery temperature occurs more often than a decrease, which is most notable for distances of 10 km or 20 km.

The weather data set is analyzed on a subset of 1000 randomly selected data points (i.e., segments). The first and last six segments of each drive are excluded from the random draw since they could be located in parking garages where different ambient temperatures might be measured. Since the available weather stations are spread over the country and not necessarily close to the moving vehicle, the distance between the vehicle and the closest weather station d_{ws} is taken into account. Segments with d_{ws} larger than 74 km are located outside of Germany and excluded from the analysis, because the closest German weather station is too far away.

A heat map of d_{ws} versus the difference in measured ambient temperature ΔT_a (6) between vehicle ($T_{a,vehicle}$) and weather station ($T_{a,ws}$) is given in Fig. 7. ΔT_a is in a range of -2.1 °C to 3.5 °C in 95% of the segments. Higher differences can be related to height differences since most weather stations are located on top of hills or smaller mountains (below the previously defined threshold of 1000 m above sea level). That is also the cause for $T_{a,vehicle}$ on average being larger than $T_{a,ws}$.

$$\Delta T_a = T_{a,vehicle} - T_{a,ws} \tag{6}$$

The step of understanding vehicle fleet data in combination with data from weather stations increases the awareness of limitations and potential for the usage as training data. Its results show a suitability of the collected data for a usage in the following steps.

C. DATA PREPROCESSING

The step of data processing leads to three data sets (training, validation and test) that are filtered and shaped such that they can be directly used for model building, training and evaluation. As a result, there are 1630 drives in the training data set, 359 drives in the validation data set and 245 drives in the test data set. The data sets are balanced according to a fitting curve derived by the segment distribution of occurring maximum differences $\Delta T_{b,max}$ (7) of the training data before fitting. It takes into account that there is a high share of drives with few or no change in battery temperature. The obtained distribution and occurrence fitting curve are shown in Fig. 8. The equation of the occurrence fitting curve p_{fit} is based on an exponential decay and defined by (8) and (9).

$$\Delta T_{b,max} = max(T_b) - min(T_b) \tag{7}$$

$$p_{fit}\big(\Delta T_{b,max}\big) = 0.015 \cdot exp\big(-0.25 \cdot \Delta T_{b,max}\big) \tag{8}$$

$$\Delta T_{b,max} \leq 5\,°\mathrm{C} \tag{9}$$

For different horizon sizes, the fitting curve might need to be adapted because of the dependence on the horizon size (see Section III-B). After fitting the distribution by randomly removing segments, the number of segments in training data is reduced by 55.2% and the number of drives by 3.1%. This means that the number of considered drives is not significantly reduced, but the number of segments of these drives. The test data contains 237 drives (with 9395 data points) after balancing.

Feature engineering and selection is based on physical dependencies and redundancies identified by Pearson correlation. As a result, all data sets include 43 history features (17 foresight features) which are based on 19 (6) physical quantities, 8 (4) features added by mathematical operations, 14 (5) qualifier signals and, both for history and foresight, an upper battery cooling threshold $T_{b,cool,start}$ and a lower threshold (2 °C below $T_{b,cool,start}$, triggers end of cooling). A list of all features is provided in Appendix B. The history horizon size is determined to 5 km and the sizes of the foresight input horizon and the prediction horizon are 20 km. As a result, the history input shape is (20, 43) and the foresight input shape is (80, 17). Dependent on the number of quantiles n_q (in this work 7), the output shape is (80, n_q).

D. MODEL BUILDING AND TRAINING

The architecture described in Section II-D is built and trained for varying hyperparameters. A list of fixed parameters is included in Appendix C. In this work, the variation space is defined as follows:

- Base number of layer nodes = [128, 192, 256]
- Dropout rate = [0.2, 0.3, 0.4]
- Loss weight α_{MSE} = [0.01, 0.03, 0.1]
- Learning rate = [0.001, 0.0001, 0.00001]
- Mini-batch size = [32, 64, 128]

Due to training time, not all 243 possible combinations are considered in this work. Instead, models are built and trained for randomly chosen sets of these parameters. For each random choice, building and training are repeated three times which results in three different models with the same set. This takes the randomization of the training process into account. In total, 150 models are trained with 50 different sets of hyperparameters, which covers 20.58% of all possible combinations. All resulting models are evaluated and compared in the next step.

E. MODEL TESTING AND EVALUATION

After training, all models are tested with the collected test data set. Results of the evaluation metrics are shown in Appendix D. The worst performance is related to model training stuck in a local minimum with a high mini-batch size, small learning rate and a stop of training at an early epoch (due to early stop callback). These models also have a high sharpness value ψ, which reduces the practical value of their predictions due to the large range between upper and lower quantiles.

The set of hyperparameters and the resulting metrics of the model with best performance (model number 149) is included in Table 5 in Appendix D. It has a total loss on the test data of 4.09 × 10^{-4}. The performance plot of model number 149 is given in Fig. 9. Training is stopped after 57 epochs when the early stop callback identifies no further improvement of the validation error. The second best model (model number 148) consists of the same hyperparameters, which confirms their suitability. The third model trained with the same set of hyperparameters is on seventh place, with a total loss of 4.34 × 10^{-4}. Hence, it does not exhibit the best possible performance of this set (compared to models 149 and 148), which shows the importance of repeating building and training of models.

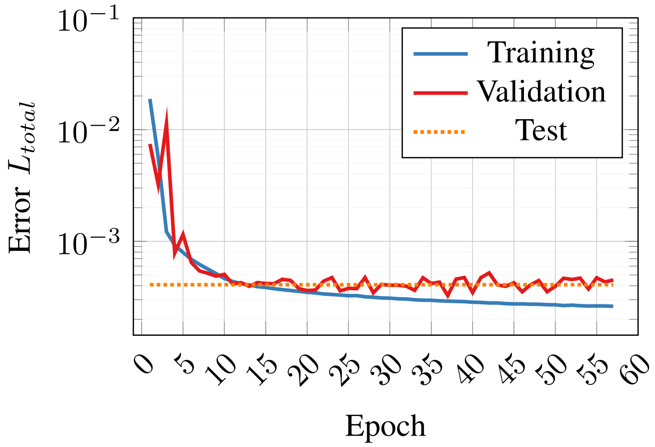


FIGURE 9. Training and validation error during training of best performing model 149. The error of the model on test data after training is finished is included as a dotted line.

TABLE 3. Metrics per quantile of the best performing model with units if applicable. The first group relates to point-forecast metrics, the second group to quantile-related metrics.

quantile	0.01	0.10	0.25	0.50	0.75	0.90	0.99
MSE ((°C)^2)	0.55	0.26	0.19	0.16	0.17	0.23	0.47
MAE (°C)	0.59	0.36	0.30	0.27	0.29	0.37	0.57
RMSE (°C)	0.74	0.51	0.44	0.40	0.41	0.48	0.68
R^2	-0.02	0.51	0.64	0.71	0.69	0.57	0.14
σ_{q_j} (°C/km)	0.12	0.07	0.07	0.07	0.07	0.08	0.13
$p(q_j)$	0.04	0.18	0.29	0.47	0.68	0.81	0.93
$p(q_j, 1-q_j)$	0.89	0.64	0.39	n/a	0.39	0.64	0.89
WS	3.01	1.08	0.79	n/a	0.79	1.08	3.01
CORS in 10^{-3}	0.05	11.15	6.35	0.47	0.22	0.25	0.25
ψ_{avg} (°C)	1.11	0.56	0.29	n/a	0.29	0.56	1.11
ψ_{max} (°C)	7.53	4.08	2.25	n/a	2.25	4.08	7.53

Besides the total loss, a comparison of other metrics provides a deeper understanding of the performance. For instance, model 9 (fourth place) shows a high CORS value of 51.44 × 10^{-3}, compared to 2.68 × 10^{-3} for model 149. Thus, its predictions exhibit unfavorable high quantile-crossing despite the low total loss. Model 121 (third place) shows better quantile-related metrics, but point-forecast metrics and sharpness are worse than for the other four models. Choosing a model for application therefore requires an evaluation based on all given metrics.

The best performing model 149 is further analyzed concerning its point and quantile forecasts. Table 3 shows the metrics for each quantile. The point-forecast results (MSE, MAE and RMSE) are higher for outer quantiles (due to L_{pb} and different weights in L_{cwq}). Upper quantiles (0.75, 0.90, 0.99) show better point-forecast results than their lower counterparts. The table includes the achieved shares of true values below the according quantile $p(q_j)$ or within a quantile

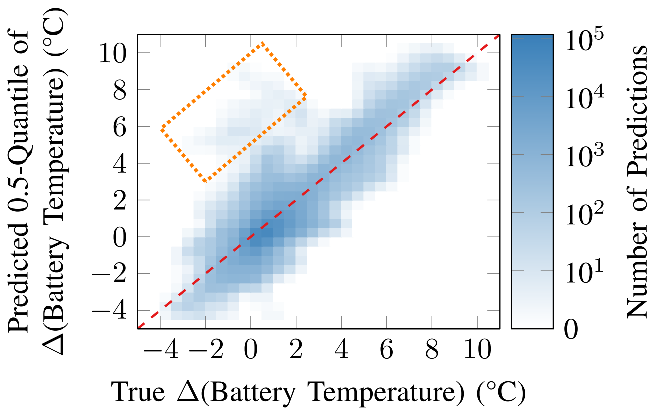


FIGURE 10. Regression plot with the logarithmic number of median (0.50-quantile) predictions of the best performing model, in comparison with the true change in battery temperature for all test data and over all segments of the prediction horizon. Outliers are marked with a rectangle and the identity line (dashed) indicates where predictions equal the true value.

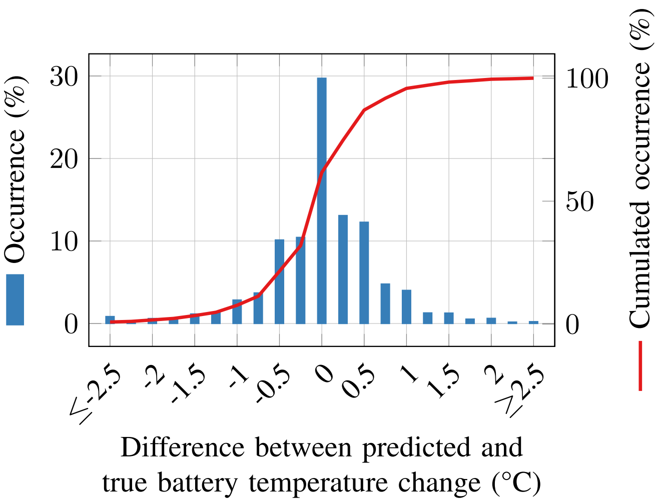
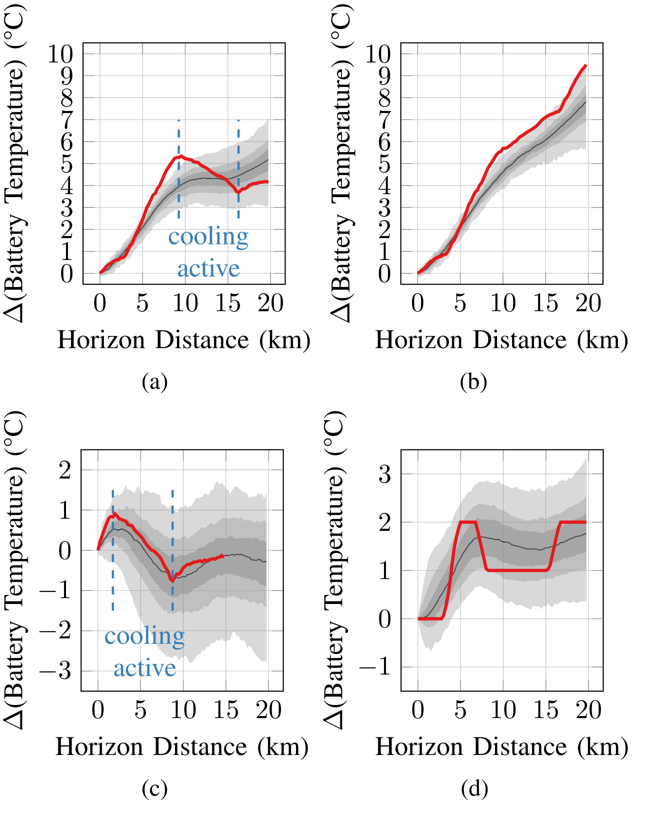


FIGURE 11. Histogram and cumulated histogram of the occurrence of differences between the quantile 0.5 (median) predictions and true values of T_b of the test data.

interval $p(q_j, 1 - q_j)$. Good performance is achieved for values close to the quantile definition, for example 47% of all true values are lower than the predicted quantile 0.50 and 89% of all true values are in the quantile interval 0.98 (between quantile 0.99 and 0.01). Metrics related to quantile intervals ($p(q_j, 1-q_j)$, Winkler Score and Sharpness ψ) cannot be calculated for quantile 0.50 (median). They are symmetric around the median, since they describe properties of quantile intervals. Sharpness is lower for inner quantile pairs according to their definition (range between upper and lower quantile).

A heatmap of the differences between median (point) predictions of model 149 and true values is shown in Fig. 10 as a regression plot. It shows a general fit between the predictions and true values for any observed change in battery temperature. Outliers of predictions occur in the area marked with a rectangle, where a temperature change is overestimated to higher values. This is more acceptable than an underestimation, because it leads to a more conservative control (i.e., lower cooling thresholds), considering higher aging at high battery temperatures. A histogram of the differences between median (point) predictions and true values is shown in Fig. 11. In more than 90% of the segments the absolute difference between median and true value is smaller than 1.1 °C, in more than 95% smaller than 1.5 °C. The detailed analysis shows a general capability of the model to predict the change in battery temperature.

In a further investigation, predictions of simulations with different battery cooling thresholds are compared. Fig. 12 (a) and (b) depict an example of two predictions with the same drive profile and boundary conditions but a shifted cooling threshold. It shows the difference in T_b in relation to T_b(0 km) at the position of the vehicle. The effect of the cooling threshold is represented both in the actual results and the predictions: In Fig. 12 (a) the activation of cooling by the thermal management system after 9 km leads to a decrease in T_b. In Fig. 12 (b) a higher cooling threshold prevents battery cooling and T_b continues to increase instead.



(a)

(b)

(c)

(d)

FIGURE 12. Prediction result of model 149 for a simulated profile with cooling threshold 35 °C (a) and 40 °C (b), both with a battery temperature of 30 °C at the current segment (0 km). A prediction after 7.5 km of the same profile with threshold 35 °C is shown in (c), where the end of drive is reached at 15 km. An examplary prediction on vehicle fleet data is given by (d), with a battery temperature of 12 °C at the current segment (0 km). The predictions are shown as shaded areas for quantile intervals (0.01/0.99, 0.1/0.9 and 0.25/0.75 from outer to inner), with quantiles as their boundaries from bottom (0.01) to top (0.99). The median is shown as a (black) thin line, the correct value as a (red) thick line. Battery cooling is active between the (blue) dashed lines.

This difference in prediction is an important ability in the context of predictive thermal management, since it needs to evaluate if T_b reaches unwanted values with different cooling thresholds.

The prediction in Fig. 12 (a) shows a greater discrepancy after 6 km of the prediction horizon but follows the general trend. Another prediction 7.5 km later is closer to the true values, as shown in Fig. 12(c). Fig. 12(d) depicts a prediction of a vehicle drive. In contrast to the discrete behavior of the true, sensor-measured values, the predicted quantiles are smoother but show the same trend.

The results in this section indicate the capability of the method to develop models that can be used in a predictive controller of the thermal management of the battery. Despite that and considering the observed limitations, a further optimization of the model needs to be investigated to achieve better performance. Examples for further optimization are given in the following section.

IV. DISCUSSION

The best performing model shows the method's suitability to develop a prediction model for a predictive BTMS. The performance is achieved based on rather small data sets from different domains, that are analyzed for a better understanding of limitations and preprocessed for the usage of building, training and evaluating a prediction model. Besides the potential of the method, there are several aspects related to each method step that can be optimized.

The sizes of the collected data sets allow fast prototyping, but limit data balancing and the usage of more complex model architectures. For example, deeper neural networks require more training data [37]. This can be achieved by conducting more simulations, collecting more vehicle fleet data or by data augmentation. In this work, the effect of different battery cooling thresholds is solely present in simulation data. Therefore, predictions with the considered variation of thresholds cannot be evaluated using vehicle fleet data. Furthermore, when using more vehicle fleet data in future work, more simulation data are needed to preserve the share of training data with different thresholds.

The steps data understanding and preprocessing provided insights in the suitability of the collected data as input data for model training and testing. The impact of additional features, such as joined weather data or engineered features, needs to be further investigated. In deeper architectures, feature engineering might become less important [38], [68]. In this work, foresight input data consist of actual, measured values, such that the results shown represent a best case. In an application, the model needs to handle inaccuracies introduced by deviations from foresight input data. The sizes of history horizon, input foresight horizon and prediction horizon need to be analyzed according to the application. For example, a shorter prediction horizon might lead to a higher accuracy but a smaller scope for the usage in a predictive control.

In future work, the model can be trained on larger data sets and extended by multivariate convolutions, recurrent layers and a deeper network. Optimizers can be used for a more thorough optimization of hyperparameters, not only based on a random choice of given sets. Continuous training in the car after deployment can adapt the prediction model to individual drivers and driving profiles.

Evaluation and ranking of the trained models was addressed by metrics for different performance criteria. Absolute differences between median predictions of the best performing model and the true values smaller than 1.1 °C in 90% of the predictions are acceptable due to sensor discreteness (1 °C resolution). Differences between prediction and true values can be caused by simplifications of the simulation model and by sensor inaccuracy of vehicle fleet data. In future work, an optimized model needs to be compared with alternative methods of time-series forecasting and tested in a predictive control of a BTMS.

V. CONCLUSION

This work proposed a method for developing a prediction model of the battery temperature for the later usage in predictive control of battery thermal management systems (BTMS). Its output was a convolutional neural network (CNN) that predicted quantiles of the change in battery temperature for different cooling thresholds as sequences over a prediction horizon. The contribution of this work's method included an understanding and preprocessing of cross-domain data from simulation and vehicle fleet for the development of CNN. The CNN architecture took into account both data from previous segments and data from predictions for following segments (using route and weather forecast) of the same drive. The practical value of the model was increased by the prediction of quantiles which add the model's uncertainty as input for predictive control.

The best performing model out of 150 trained models showed promising results. A mean absolute error (MAE) of the median prediction of 0.27 °C and an absolute difference between prediction and true value smaller than 1.1 °C in more than 90% of the segments showed the ability of the model to produce acceptable point-forecasts. The difference between the theoretical probabilities of quantiles and the achieved occurrence of true values within the quantiles showed the capability of predictions including the model's uncertainty, with potential for improvement. For example, 47% of the true values were below the predicted 0.5 quantile (median), and 93% of the true values were below the 0.99 quantile. A plausible effect of different cooling thresholds on predictions pointed out the general suitability of the model for a predictive BTMS, for instance to reduce the energy consumption by avoiding battery cooling when a lower power demand is expected.

The contributions of this work could be translated to the development of models for other applications in which cross-domain data for both a history and foresight channel would be available. Examples could be a predictive thermal or energy management of the battery of other vehicle types, of other vehicle components (e.g., electric machines), of buildings or for vehicle motion planning.

In future work, further optimization could target the size of training data by collecting more fleet data and by conducting data augmentation. Based on the increased size of training data, the architecture could become more complex and consist of more layers, including recurrent layers. The horizon sizes of prediction and both input channels could be optimized with regards to the application of predictive control. An optimized model with better performance should be applied in predictive control of BTMS and compared to other methods.

APPENDIX A

The following equations describe heat generation and heat transfer quantities used in (1). Heat generation of the battery (10) includes its inner resistance R_b, current I_b, voltage U_b and open circuit voltage U_{ocv} [69]. Heat transfer from or to the surroundings depends on the ambient temperature T_a and a factor of combined heat transfer coefficient h_s and surface A_s (11). The heat transfer coefficient combines convection and radiation which depends on further quantities [70, pp. 25–29]. Heat transfer caused by the thermal management system depends on the coolant temperature T_c, heat transfer coefficient h_{tm} and surface A_{tm} (12). The coolant temperature is controlled by cooling or heating demand from the battery and cooling or heating power supply from the thermal management system.

$$\dot{Q}_{b,gen} = R_b I_b^2 + I_b(U_b - U_{ocv}) + I_b T_b \frac{\partial U_b}{\partial T_b} \tag{10}$$

$$\dot{Q}_s = h_s A_s (T_b - T_a) \tag{11}$$

$$\dot{Q}_{tm} = h_{tm} A_{tm} (T_b - T_c) \tag{12}$$

The battery current I_b can be described by

$$I_b = \frac{P_b}{U_b} = \frac{P_{em} + P_{cons}}{U_b}, \tag{13}$$

with the battery power P_b, the electric power of the electric machine P_{em} (14) and subsidiary consumer power consumption P_{cons}. P_{em} depends on driving resistance F_{res} and efficiency η_{em} of the electric machine. The driving resistance depends on velocity v, slope angle θ (or road height profile) and further quantities [61].

$$P_{em} \approx \frac{F_{res}(v^2, \dot{v}, \theta, \ldots,) \cdot v}{\eta_{em}}. \tag{14}$$

APPENDIX B

All input features that result from the step of understanding and preprocessing are shown in Table 4, including their qualifier QL if needed. It is marked with 1 if the feature is used in the history or foresight channel and 0 otherwise.

TABLE 4. Input features of history and foresight channel.

Parameter	History Channel	Foresight Channel
Acceleration pedal	1	0
Ambient temperature	1	1
Battery cooling threshold end	1	1
Battery cooling threshold start	1	1
Battery SOC rate	1	0
Battery SOC squared	1	0
Battery State of Charge (SOC)	1	0
Battery temperature	1	0
Battery temperature - Ambient temperature	1	0
Battery temperature rate	1	0
Cabin target temperature	1	0
Cabin target temperature QL	1	0
Cabin temperature	1	0
Cabin temperature QL	1	0
DPMA acceleration propulsion	1	0
DPMA acceleration propulsion QL	1	0
DPMA acceleration recuperation	1	0
DPMA acceleration recuperation QL	1	0
DPMA speed propulsion	1	0
DPMA speed propulsion QL	1	0
DPMA speed recuperation	1	0
DPMA speed recuperation QL	1	0
Electric machine stator temperature	1	0
Electric machine stator temperature QL	1	0
Global radiation	1	1
Global radiation QL	1	1
Inverter temperature	1	0
Inverter temperature QL	1	0
Relative humidity	1	1
Relative humidity QL	1	1
Road height difference negative sum	1	1
Road height difference negative sum QL	1	1
Road height difference positive sum	1	1
Road height difference positive sum QL	1	1
Speed	1	1
Speed difference	1	1
Speed difference squared	1	1
Speed inverted	1	1
Speed to the power of five	1	1
Temperature after heat pump	1	0
Temperature after heat pump QL	1	0
Vehicle torque	1	0
Within horizon QL	1	1

APPENDIX C

Additional settings of parameters of the neural network in this work:

- Optimizer: Adaptive Moment Estimation (ADAM)
- Maximum number of epochs: 200
- Early stopping patience: 20
- Quantiles: [0.01, 0.1, 0.25, 0.5, 0.75, 0.9, 0.99]
- Initial quantile weights (after constrains and softmax): [0.024, 0.064, 0.175, 0.475, 0.175, 0.064, 0.024]
- Horizon lengths: 20 (history), 80 (foresight and prediction)
- Kernel and bias constraint: maxnorm with weight 2
- Activation function: “relu” (“linear” in final dense layers)
- Conv1D kernel size: 3 (history), 15 (foresight), 25 and 15 (combined)
- MaxPooling poolsize: 2 (history), 3 (foresight and combined)
- Scaling factor for number of nodes: 2 (Conv1D) and 0.5 (Dense) for history and foresight, 1.75 (Dense in

TABLE 5. Hyperparameters and quantile-averaged metrics of the best five models with units if applicable. The best (lowest) value of each metric is marked in bold.

Model nbr	149	148	121	9	13
Batchsize	32	32	32	64	32
Base node number	192	192	128	256	128
Dropout Rate	0.3	0.3	0.4	0.3	0.2
α_{MSE}	0.01	0.01	0.03	0.01	0.03
Learning Rate	10^{-4}	10^{-4}	10^{-5}	10^{-3}	10^{-4}
Epochs	57	119	200	77	88
Total Loss in 10^{-4}	**4.09**	4.11	4.12	4.22	4.28
MSE ((°C)^2)	**0.29**	**0.29**	0.40	0.35	**0.29**
MAE (°C)	**0.39**	0.40	0.46	0.44	**0.39**
RMSE (°C)	**0.52**	0.53	0.60	0.57	**0.52**
$R^2_{0.5}$	**0.71**	**0.71**	0.69	0.67	0.69
σ_{q_j} (°C/km)	0.09	0.13	0.11	0.33	**0.08**
$p(q_j)_{d,abs}$	0.06	0.06	**0.05**	0.09	0.07
$p(q_j, 1-q_j)_{d,abs}$	0.12	0.13	**0.05**	0.10	0.12
WS	1.62	1.52	**1.01**	1.18	1.77
CORS in 10^{-3}	2.68	**1.31**	1.88	51.44	1.98
ψ_{avg} (°C)	0.65	0.66	0.85	0.76	**0.64**
ψ_{max} (°C)	**4.62**	4.93	7.63	7.39	6.49

TABLE 6. Hyperparameters and quantile-averaged metrics of the worst five models with units if applicable. The worst (highest) value of each metric is marked in bold.

Model nbr	68	91	36	46	128
Batchsize	128	64	128	128	128
Base node number	192	256	192	128	128
Dropout Rate	0.4	0.2	0.4	0.4	0.3
α_{MSE}	0.1	0.03	0.1	0.03	0.01
Learning Rate	10^{-4}	10^{-5}	10^{-4}	10^{-5}	10^{-5}
Epochs	61	32	46	52	60
Total Loss in 10^{-4}	13.43	14.89	45.84	51.64	**69.74**
MSE ((°C)^2)	1.28	146.75	5.50	440.24	**721.22**
MAE (°C)	0.72	1.38	1.26	**2.34**	2.20
RMSE (°C)	1.09	9.85	2.23	17.11	**22.00**
$R^2_{0.5}$	-0.42	-4.30	-6.63	-34.13	**-159.07**
σ_{q_j} (°C/km)	0.79	11.67	1.99	31.88	**54.65**
$p(q_j)_{d,abs}$	0.15	0.03	**0.22**	0.04	0.03
$p(q_j, 1-q_j)_{d,abs}$	0.25	0.04	**0.37**	0.05	0.03
WS	8.98	1.27	**30.90**	0.76	0.99
CORS in 10^{-3}	122.93	5.46	**296.34**	7.57	3.39
ψ_{avg} (°C)	0.95	2.90	1.01	**5.05**	4.59
ψ_{max} (°C)	5.76	515.49	14.88	1003.15	**1805.81**

Concatenate) and 0.25 (Dense in ChannelOut) for combined. The number of nodes of the quantile output layers equals the fixed horizon length of 80. All other layers have a scaling factor of 1.

APPENDIX D

All models resulting from the loop with random choice of hyperparameters are evaluated using the metrics described in Section II-E and sorted by the total loss on the test data. The set of hyperparameters and metrics of the best five and worst five models are presented in Tables 5 and 6. All metrics are calculated as the average of the metrics per quantile.

REFERENCES

[1] A. Enthaler, T. Weustenfeld, F. Gauterin, and J. Koehler, "Thermal management consumption and its effect on remaining range estimation of electric vehicles," in *Proc. Int. Conf. Connected Veh. Expo (ICCVE)*, Vienna, Austria, 2014, pp. 170–177.

[2] Y. Xie, C. Wang, X. Hu, X. Lin, Y. Zhang, and W. Li, "An MPC-based control strategy for electric vehicle battery cooling considering energy saving and battery lifespan," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14657–14673, Dec. 2020.

[3] S. Park and C. Ahn, "Model predictive control with stochastically approximated cost-to-go for battery cooling system of electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 4312–4323, May 2021.

[4] L. Spitthoff, P. R. Shearing, and O. S. Burheim, "Temperature, ageing and thermal management of lithium-ion batteries," *Energies*, vol. 14, no. 5, p. 1248, 2021.

[5] A. Lajunen, Y. Yang, and A. Emadi, "Recent developments in thermal management of electrified powertrains," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 11486–11499, Dec. 2018.

[6] X. Kuang *et al.*, "Research on control strategy for a battery thermal management system for electric vehicles based on secondary loop cooling," *IEEE Access*, vol. 8, pp. 73475–73493, 2020.

[7] S. Park and C. Ahn, "Stochastic model predictive controller for battery thermal management of electric vehicles," in *Proc. IEEE Veh. Power Propulsion Conf. (VPPC)*, Hanoi, Vietnam, 2019, pp. 1–5.

[8] T. J. Shelly, J. A. Weibel, D. Ziviani, and E. A. Groll, "A dynamic simulation framework for the analysis of battery electric vehicle thermal management systems," in *Proc. 19th InterSociety Conf. Thermal Thermomech. Phenomena Electron. Syst.*, Orlando, FL, USA, 2020, pp. 538–547.

[9] S. Park and C. Ahn, "Computationally efficient stochastic model predictive controller for battery thermal management of electric vehicle," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8407–8419, Aug. 2020.

[10] T. Fischer, F. Götz, L. Berg, H.-P. Kollmeier, and F. Gauterin, "Model-based development of a holistic thermal management system for an electric car with a high temperature fuel cell range extender," in *Proc. 11th Int. Modelica Conf.*, Palais des Congrès de Versailles, France, 2015, pp. 127–133.

[11] T. Fischer, T. Kraus, C. Kirches, and F. Gauterin, "Nonlinear model predictive control of a thermal management system for electrified vehicles using FMI," in *Proc. 12th Int. Modelica Conf.*, 2017, pp. 255–264.

[12] J. Lopez-Sanz *et al.*, "Thermal management in plug-in hybrid electric vehicles: A real-time nonlinear model predictive control implementation," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 7751–7760, Sep. 2017.

[13] H. Zomorodi, D. Yoon, and B. Ayalew, "Use of predictive information for battery pack thermal management," in *Proc. Amer. Control Conf. (ACC)*, Seattle, WA, USA, 2017, pp. 5020–5025.

[14] C. Zhu, F. Lu, H. Zhang, and C. C. Mi, "Robust predictive battery thermal management strategy for connected and automated hybrid electric vehicles based on thermoelectric parameter uncertainty," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 6, no. 4, pp. 1796–1805, Dec. 2018.

[15] K. Purohit *et al.*, "Soft sensors for state of charge, state of energy, and power loss in formula student electric vehicle," *Appl. Syst. Innov.*, vol. 4, no. 4, p. 78, 2021.

[16] C. Akkaldevi, S. D. Chitta, J. Jaidi, S. Panchal, M. Fowler, and R. Fraser, "Coupled electrochemical-thermal simulations and validation of minichannel cold-plate water-cooled prismatic 20 ah LiFePO$_4$ battery," *Electrochem*, vol. 2, no. 4, pp. 643–663, 2021.

[17] M.-K. Tran, C. Cunanan, S. Panchal, R. Fraser, and M. Fowler, "Investigation of individual cells replacement concept in lithium-ion battery packs with analysis on economic feasibility and pack design requirements," *Processes*, vol. 9, no. 12, p. 2263, 2021.

[18] E. Chemali, P. J. Kollmeyer, M. Preindl, Y. Fahmy, and A. Emadi, "A convolutional neural network approach for estimation of li-ion battery state of health from charge profiles," *Energies*, vol. 15, no. 3, p. 1185, 2022.

[19] A. Afzal, J. K. Bhutto, A. Alrobaian, A. R. Kaladgi, and S. A. Khan, "Modelling and computational experiment to obtain Optimized neural network for battery thermal management data," *Energies*, vol. 14, no. 21, p. 7370, 2021.

[20] Y. Liu and J. Zhang, "Self-adapting intelligent battery thermal management system via artificial neural network based model predictive control," in *Proc. Vol. 2A 45th Des. Autom. Conf.*, Anaheim, CA, USA, 2019, p. 10.

[21] Y. Liu and J. Zhang, "Electric vehicle battery thermal and cabin climate management based on model predictive control," *J. Mech. Des.*, vol. 143, no. 3, p. 8, 2020.

[22] J. Park and Y. Kim, "Supervised-learning-based optimal thermal management in an electric vehicle," *IEEE Access*, vol. 8, pp. 1290–1302, 2020.

[23] T. J. Kim, B. D. Youn, and H. J. Kim, "Battery pack temperature estimation model for EVS and its semi-transient case study," *Chem. Eng. Trans.*, vol. 33, pp. 955–960, Jul. 2013.

[24] S. Arora, W. Shen, and A. Kapoor, "Neural network based computational model for estimation of heat generation in $\mathrm{LiFePO_4}$ pouch cells of different nominal capacities," *Comput. Chem. Eng.*, vol. 101, pp. 81–94, Jun. 2017.

[25] Y. Jiang, Y. Yu, J. Huang, W. Cai, and J. Marco, "Li-ion battery temperature estimation based on recurrent neural networks," *Sci. China Technol. Sci.*, vol. 64, pp. 1335–1344, Apr. 2021.

[26] J. Park *et al.*, "Intelligent vehicle power control based on machine learning of optimal control parameters and prediction of road type and traffic congestion," *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 4741–4756, Nov. 2009.

[27] Y. Jeong, S. Kim, and K. Yi, "Surround vehicle motion prediction using LSTM-RNN for motion planning of autonomous vehicles at multi-lane turn intersections," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 2–14, 2020.

[28] J. W. Moon and J.-J. Kim, "ANN-based thermal control models for residential buildings," *Build. Environ.*, vol. 45, no. 7, pp. 1612–1625, 2010.

[29] J. W. Moon, S.-H. Yoon, and S. Kim, "Development of an artificial neural network model based thermal control logic for double skin envelopes in winter," *Build. Environ.*, vol. 61, no. 1, pp. 149–159, 2013.

[30] G. Gao, J. Li, and Y. Wen, "DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8472–8484, Sep. 2020.

[31] K. Mason and S. Grijalva, "A review of reinforcement learning for autonomous building energy management," *Comput. Elect. Eng.*, vol. 78, pp. 300–312, Sep. 2019.

[32] M.-K. Tran, S. Panchal, T. D. Khang, K. Panchal, R. Fraser, and M. Fowler, "Concept review of a cloud-based smart battery management system for lithium-ion batteries: Feasibility, logistics, and functionality," *Batteries*, vol. 8, no. 2, p. 19, 2022.

[33] K. Grolinger, M. Hayes, W. A. Higashino, A. L'Heureux, D. S. Allison, and M. A. M. Capretz, "Challenges for MapReduce in big data," in *Proc. IEEE World Congr. Services*, Anchorage, AK, USA, 2014, pp. 182–189.

[34] J. Djolonga *et al.*, "On robustness and transferability of convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16458–16468.

[35] A. F. Kuri-Morales, "The best neural network architecture," in *Proc. 13th Mexican Int. Conf. Artif. Intell.*, Tuxtla Gutiérrez, Mexico, 2014, pp. 72–84.

[36] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, p. 1, 2015.

[37] D. Jha *et al.*, "Enabling deeper learning on big data for materials informatics applications," *Sci. Rep.*, vol. 11, no. 1, p. 4244, 2021.

[38] A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, 2017.

[39] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine learning on big data: Opportunities and challenges," *Neurocomputing*, vol. 237, pp. 350–361, May 2017.

[40] Y. Zheng, "Methodologies for cross-domain data fusion: An overview," *IEEE Trans. Big Data*, vol. 1, no. 1, pp. 16–34, Mar. 2015.

[41] L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, and J. Garcke, "Combining machine learning and simulation to a hybrid modelling approach: Current and future directions," in *Proc. 18th Int. Symp. Intell. Data Anal.*, Konstanz, Germany, 2020, pp. 548–560.

[42] L. von Rueden *et al.*, "Informed machine learning—A taxonomy and survey of integrating prior knowledge into learning systems," *IEEE Trans. Knowl. Data Eng.*, early access, May 12, 2021, doi: 10.1109/TKDE.2021.3079836.

[43] C. F. Minett, A. M. Salomons, W. Daamen, B. van Arem, and S. Kuijpers, "Eco-routing: Comparing the fuel consumption of different routes between an origin and destination using field test speed profiles and synthetic speed profiles," in *Proc. IEEE Forum Integr. Sustain. Transp. Syst.*, Vienna, Austria, 2011, pp. 32–39.

[44] P. Engel, S. Lempp, A. Rausch, and W. Tegethoff, "Improving thermal management of electric vehicles by prediction of thermal disturbance variables," in *Proc. ADAPTIVE*, Barcelona, Spain, 2018.

[45] P. T. Tsilingiris, "Thermophysical and transport properties of humid air at temperature range between 0 and 100°C," *Energy Convers. Manag.*, vol. 49, no. 5, pp. 1098–1110, 2008.

[46] G. J. Marshall, C. P. Mahony, M. J. Rhodes, S. R. Daniewicz, N. Tsolas, and S. M. Thompson, "Thermal management of vehicle cabins, external surfaces, and onboard electronics: An overview," *Engineering*, vol. 5, no. 5, pp. 954–969, 2019.

[47] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.

[48] J.-S. Ang, K.-W. Ng, and F.-F. Chua, "Modeling time series data with deep learning: A review, analysis, evaluation and future trend," in *Proc. 8th Int. Conf. Inf. Technol. Multimedia (ICIMU)*, Selangor, Malaysia, 2020, pp. 32–37.

[49] Z. Liu, Z. Zhu, J. Gao, and C. Xu, "Forecast methods for time series data: A survey," *IEEE Access*, vol. 9, pp. 91896–91912, 2021.

[50] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philos. Trans. Roy. Soc. A*, vol. 379, no. 2194, 2021, Art. no. 20200209.

[51] N. K. Ahmed, A. F. Atiya, N. El Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric Rev.*, vol. 29, nos. 5–6, pp. 594–621, 2010.

[52] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[53] S. B. Taieb and A. F. Atiya, "A bias and variance analysis for multistep-ahead time series forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 1, pp. 62–76, Jan. 2016.

[54] C.-L. Liu, W.-H. Hsaio, and Y.-C. Tu, "Time series classification with multivariate convolutional neural network," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4788–4797, Jun. 2019.

[55] B. Qian *et al.*, "Dynamic multi-scale convolutional neural network for time series classification," *IEEE Access*, vol. 8, pp. 109732–109746, 2020.

[56] M. Carney, P. Cunningham, J. Dowling, and C. Lee, "Predicting probability distributions for surf height using an ensemble of mixture density networks," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 113–120.

[57] D. Gan, Y. Wang, S. Yang, and C. Kang, "Embedding based quantile regression neural network for probabilistic load forecasting," *J. Mod. Power Syst. Clean Energy*, vol. 6, no. 2, pp. 244–254, 2018.

[58] M. Lopez-Martin, A. Sanchez-Esguevillas, L. Hernandez-Callejo, J. I. Arribas, and B. Carro, "Additive ensemble neural network with constrained weighted quantile loss for probabilistic electric-load forecasting," *Sensors*, vol. 21, no. 9, p. 2979, 2021.

[59] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, "A multi-horizon quantile recurrent forecaster," in *Proc. 31st Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 1–8.

[60] M. Cussigh, T. Straub, M. Frey, T. Hamacher, and F. Gauterin, "An all-electric alpine crossing: Time-optimal strategy calculation via fleet-based vehicle data," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 134–146, 2020.

[61] T. Straub, M. Frey, and F. Gauterin, "Learning from the fleet: Map attributes for energetic representation of driving profiles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 471–482, Jan. 2022.

[62] M. Saripuddin, A. Suliman, S. S. Sameon, and B. N. Jorgensen, "Random undersampling on imbalance time series data for anomaly detection," in *Proc. 4th Int. Conf. Mach. Learn. Mach. Intell. (MIMI)*, Hangzhou, China, 2021, pp. 151–156.

[63] X. Li, S. Chen, X. Hu, and J. Yang, “Understanding the disharmony between dropout and batch normalization by variance shift,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 2677–2685.

[64] P. Luo, X. Wang, W. Shao, and Z. Peng, “Towards understanding regularization in batch normalization,” in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019, pp. 1–23.

[65] C. Garbin, X. Zhu, and O. Marques, “Dropout vs. batch normalization: An empirical study of their impact to deep learning,” *Multimedia Tools Appl.*, vol. 79, no. 19, pp. 12777–12815, 2020.

[66] S. Zhang, Y. Bao, P. Zhou, H. Jiang, and L. Dai, “Improving deep neural networks for LVCSR using dropout and shrinking structure,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 6849–6853.

[67] Deutscher Wetterdienst. “Climate Data Center (CDC).” Offenbach. 2021. [Online]. Available: https://opendata.dwd.de/climate_environment/CDC/ (Accessed: Nov. 27, 2021).

[68] K. Ong, S.-C. Haw, and K.-W. Ng, “Deep learning based-recommendation system: An overview on models, Datasets, evaluation metrics, and future trends,” in *Proc. 2nd Int. Conf. Comput. Intell. Intell. Syst.*, Bangkok, Thailand, 2019, pp. 6–11.

[69] B. Wu, V. Yufit, M. Marinescu, G. J. Offer, R. F. Martinez-Botas, and N. P. Brandon, “Coupled thermal–electrochemical modelling of uneven heat generation in lithium-ion battery packs,” *J. Power Sour.*, vol. 243, pp. 544–554, Dec. 2013.

[70] Y. A. Cengel, *Heat Transfer: A Practical Approach*, 2nd ed. New York, NY, USA: McGraw-Hill, 2002.

ANDREAS M. BILLERT received the B.Sc. degree in engineering science and the M.Sc. degree in automotive technology from Technical University of Munich in 2017 and 2020, respectively, with visiting studies at University of California Berkeley and Shanghai Jiao Tong University. He is currently pursuing the Ph.D. cooperatively with BMW AG and Karlsruhe Institute of Technology by supervision of F. Gauterin. His research interests include high-voltage batteries, predictive thermal management, machine learning, and neural networks.

MICHAEL FREY received the Diploma and Doctoral degrees in mechanical engineering from the University of Karlsruhe in 1993 and 2004, respectively. He is the Manager of the Research Group Automation and the Research Group Suspension and Propulsion Systems, Institute of Vehicle System Technology. His research interests are autonomous driving, driver assistance systems, operational strategies, suspension systems, vehicle dynamics, as well as vehicle modeling and optimization.

FRANK GAUTERIN received the diploma degree in physics from the University of Münster, Germany, in 1989, and the Dr.rer.nat. (Ph.D.) degree from the University of Oldenburg, Germany, in 1994. From 1989 to 2000, he worked as an Acoustics Engineer, and from 2000 to 2006 as the Director of the Noise Vibration & Harshness (NVH) Engineering Profit Centre, Continental AG, Hanover, Germany, with locations in Germany, USA, and Malaysia. Since 2006, he has been a Full Professor of Vehicle Technology with the Karlsruhe Institute of Technology (KIT), Germany, the Head of the Institute of Vehicle System Technology, and Scientific Spokesperson, KIT Centre Mobility Systems. His research interests are vehicle conception, control, suspension and drive systems, tire road interaction, and NVH.