2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Transformer Neural Networks for Maximum Friction Coefficient Estimation of Tire-Road Contact using Onboard Vehicle Sensors

Hendrik Schäfke[1], Nicolas Lampe[2], and Karl-Philipp Kortmann[1]

Abstract— **For the optimization of advanced driver assistance systems (ADAS) and the implementation of autonomous driving, the perception of the vehicles environment and in particular the maximum friction coefficient $\mu_{\max}$ is crucial. Since $\mu_{\max}$ cannot be measured directly via existing serial sensors, estimating this coefficient based on available sensors is an area of research. In this paper, $\mu_{\max}$ estimation is presented using transformer neural networks (TNN) based on the input data measured by onboard vehicle sensors. The TNN is applied to both a simulative dataset created with *IPG CarMaker* and an experimental dataset recorded on a test track, each using a sports utility vehicle (SUV) as the test vehicle. Both datasets contain typical longitudinal and lateral driving maneuvers on different road surfaces. On an independent test dataset, the data-based TNN approach shows improved results in estimating $\mu_{\max}$ compared to the model-based approach of an unscented Kalman filter (UKF) and to two other data-based approaches using recurrent artificial neural networks (RANNs) from previous works. In particular, the TNN responds faster and more accurate to jumps of $\mu_{\max}$, especially during lateral driving maneuvers. Moreover, the TNN has both less parameters, and training epochs compared to the RANN.**

I. Introduction

Precise knowledge of vehicle dynamics is important for enhancing the performance of advanced driver assistance systems (ADAS) and will be essential for future autonomous vehicles. Particularly the maximum friction coefficient $\mu_{\max}$ has a significant influence on the vehicle's stability and maneuverability. Since $\mu_{\max}$ cannot be measured directly by existing vehicle sensors in serial production cars, maximum friction coefficient estimation is a challenging field in modern automotive research [2].

Methods for estimating $\mu_{\max}$ can be divided into model-based and data-based. Model-based approaches use a physical model of the vehicle dynamics, which can be integrated into an estimation algorithm such as the Kalman filter [3]. In particular, an unscented Kalman filter (UKF) achieves good results, as shown in [4]–[6]. A more detailed overview about model-based approaches is given in [2] and [3]. Both sources indicate that the estimation accuracy is mainly limited by the level of detail of the physical model. Furthermore, an in-depth knowledge of the system is required for identifying and developing the physical model. As a result, data-based methods are becoming increasingly popular in applications such as vehicle state estimation, e.g. estimating the vehicles sideslip angle [7], or the vehicles roll angle [8].

Data-based methods aim to map inputs to outputs without a priori information [9]–[13]. Disadvantage of many of these methods is the loss of physical interpretability and the resulting susceptibility to errors due to overfitting to the data and lack of extrapolation capabilities. In [9] two feed-forward neural networks (FFNN) are used to estimate $\mu_{\max}$ based on the trucks king pin forces in longitudinal and lateral direction, tie rod forces, steering angle, and suspension inclination angle. In another approach, a FFNN with 12 inputs, the longitudinal and lateral acceleration, slip rates, yaw rate, wheel speeds, and steering wheel angle, and two hidden layers is used [10]. An RANN is applied in [11] for $\mu_{\max}$ estimation on simulative data, based on 18 inputs, by combining long short-term memory (LSTM) cells with deep ensembles. In difference to [10], here the inputs are extended by the throttle position, steering wheel rate, braking pressure, vehicle speed, vertical acceleration, and engine torque. A convolutional neural network (CNN) for friction potential estimation is presented in [13]. The CNN is trained and tested on experimental data with good results, even during low vehicle dynamic excitation. Contrary to the data-based approaches presented in [9]–[11], [13], which all use additional sensors as inputs, e.g. suspension inclination angle, truck kingpin forces, slip rates, or sideslip angle, in [12] only sensors that are available in serial production vehicles are used to train an RANN, with the addition that vehicle dynamic excitation monitoring is introduced.

Moreover, there are several hybrid approaches combining a physical vehicle model and data-based methods [14]–[16] for $\mu_{\max}$ estimation. On the one hand, hybrid approaches can increase accuracy while maintaining physical interpretability. On the other hand, high-level knowledge of the system to find a physical model, as well as suitable combinations of the physical model and data-based methods are still required.

Until now, RANNs based on gated recurrent units (GRUs) have achieved the best results in estimating $\mu_{\max}$ in the majority of cases [5]. This was also the case for natural language processing (NLP) until transformer neural networks (TNNs), introduced by [1], were applied to this area [17]. In recent years, TNNs have been successfully applied to other areas such as computer vision, language generation, time-series forecasting, and speech processing, in which they have achieved state of the art performance [18]. In contrast to RANNs, which are based on recurrence, TNNs rely on a self-attention mechanism, relating different positions of a single sequence in order to compute a representation of the

[1]Both autors are with the Institute of Mechatronic Systems, Leibniz University Hannover, 30823 Garbsen, Germany hendrik.schaefke@stud.uni-hannover.de

[2]Nicolas Lampe is with the Institute of Computer Engineering, Osnabrück University of Applied Sciences, 49076 Osnabrück, Germany n.lampe@hs-osnabrueck.de

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

sequence [1]. There are some examples of applying TNN to time series data [18], but there are only few in which they are used for state or parameter estimation. For instance, a TNN for estimating the state of charge of lithium-ion batteries is presented in [19]. In [20], a TNN for tool wear estimation is presented. A self-attention based approach for remaining useful life (RUL) estimation is given in [21]. Two other examples are the estimation of pan evaporation rates in [22] and the attitude estimation in smartphones based on inertial sensors, presented in [23].

As indicated, TNNs have potential in the area of parameter and state estimation, because self-attention allows for the network inputs to interact with one another and be weighted by their importance to the final estimate. Compared to RANNs, this has the advantage that many time steps can be processed simultaneously. Furthermore, transformers can converge faster than traditional sequence models [17]. Therefore we present a data-based algorithm for $\mu_{\max}$ estimation based on a TNN, which is evaluated on both simulative as well as experimental data. To the best of our knowledge, this work is the first attempt to use transformer- or self-attention-based neural networks in the context of vehicle dynamics. This novel approach is then compared to other state-of-the-art approaches for $\mu_{\max}$ estimation such as the UKF, and two RANNs based on GRUs or LSTM cells, which have been applied on the same datasets.

The paper is organized as follows: In section II, GRUs and TNNs are presented and the network architecture is proposed. The datasets, training procedures, and results for the simulative data from *IPG CarMaker* are presented in section III, and for the experimental data in section IV. Moreover, the novel estimator is compared to other state of the art $\mu_{\max}$ estimators from previous works. Finally, a conclusion and outlook are given in section V.

II. Neural Network Models

In this section, the multi-head attention based neural network for estimating $\mu_{\max}$ is presented as shown in Fig. 1. It mainly consists of transformer encoders and GRU layers, which are explained in the following. Afterwards, the overall architecture is presented.

A. Gated Recurrent Unit (GRU)

RANN have an additional feedback loop compared to FFNN, which allows them to use information from previous inputs to influence the current input. RANNs are therefore able to predict time-varying sequential data by incorporating causal relationships from the past. GRUs are a type of RANN structure, based on the LSTM cell [24]. In comparison, they are simpler in design, as they only use update and reset gates instead of input, forget, and output gates [25]. Given the input $\boldsymbol{x}^t$ and previous output $\boldsymbol{h}^{t-1}$, the reset gate $\boldsymbol{r}^t$, update gate $\boldsymbol{z}^t$, candidate $\tilde{\boldsymbol{h}}^t$, and hidden state $\boldsymbol{h}^t$ can be calculated by

$$\boldsymbol{r}^t = \text{sigmoid}(\boldsymbol{W}_{\text{xr}} x^t + \boldsymbol{W}_{\text{hr}} \boldsymbol{h}^{t-1} + \boldsymbol{b}_{\text{r}}), \tag{1}$$

$$\boldsymbol{z}^t = \sigma(\boldsymbol{W}_{\text{xz}} x^t + \boldsymbol{W}_{\text{hz}} \boldsymbol{h}^{t-1} + \boldsymbol{b}_{\text{z}}), \tag{2}$$

$$\tilde{\boldsymbol{h}}^t = \tanh(\boldsymbol{W}_{\text{x}\tilde{\text{h}}} x^t + \boldsymbol{r}^t \odot (\boldsymbol{W}_{\text{h}\tilde{\text{h}}} \boldsymbol{h}^{t-1}) + \boldsymbol{b}_{\tilde{\text{h}}}), \tag{3}$$

$$\boldsymbol{h}^t = (1 - \boldsymbol{z}^t) \odot \tilde{\boldsymbol{h}}^t + \boldsymbol{z}^t \odot \boldsymbol{h}^{t-1} \tag{4}$$

with trainable input weights $\boldsymbol{W}_{\text{xr}}$, $\boldsymbol{W}_{\text{xz}}$, $\boldsymbol{W}_{\text{x}\tilde{\text{h}}}$, recurrent weights $\boldsymbol{W}_{\text{hr}}$, $\boldsymbol{W}_{\text{hz}}$, $\boldsymbol{W}_{\text{h}\tilde{\text{h}}}$, and biases $\boldsymbol{b}_{\text{r}}$, $\boldsymbol{b}_{\text{z}}$, $\boldsymbol{b}_{\tilde{\text{h}}}$.

B. Transformer Encoder

The TNN is a novel type of neural network for processing sequential data, originally introduced in the field of NLP by [1], which can be applied to time series data as shown in [19]. In contrast to LSTM or GRU networks, a transformer lacks recurrence and can access all points of time of a sequence in parallel by relying on a self-attention

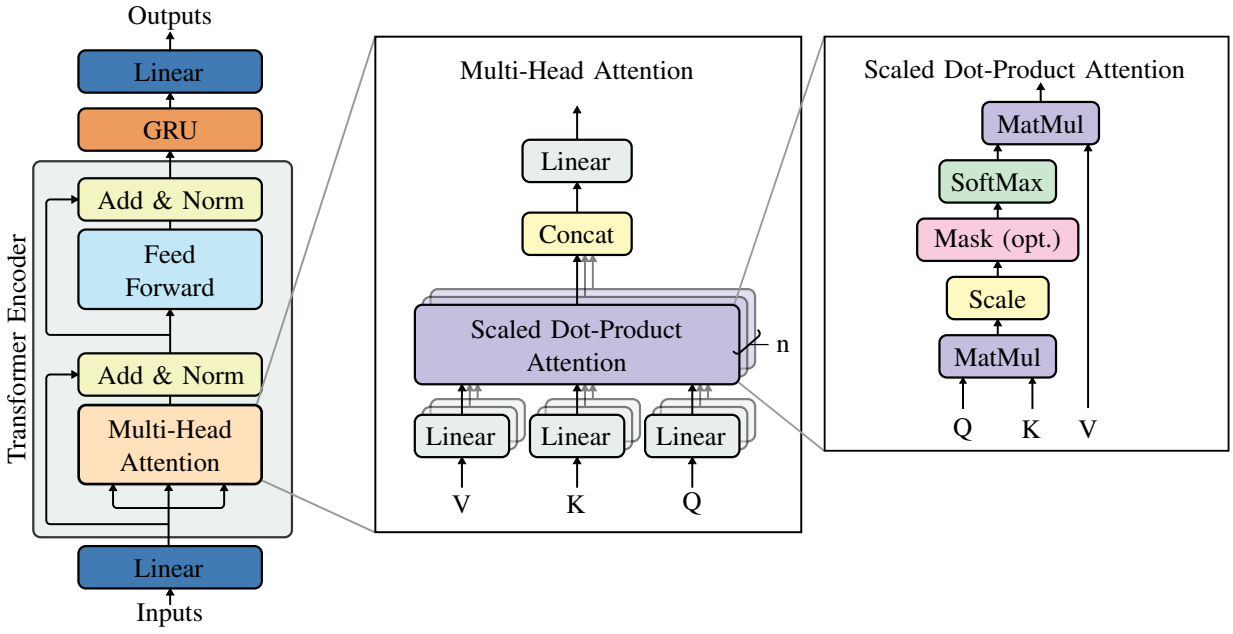


Fig. 1. Overall structure of the proposed transformer neural network model based on transformer encoders and GRUs, cf. [1].

mechanism [18]. The original TNN consists of an encoder-decoder structure. In this work an encoder-only structure is implemented to make use of independent self-attention [26].

An encoder block, which is shown in Fig. 1, consists of a multi-head self-attention module and a position-wise feed-forward network with a residual connection around each of the two sub-layers, followed by layer normalization [1]. The input data from k sensors and a time window length T is defined as $\boldsymbol{n}_T = (t_1, t_2, \dots t_T)^{\mathrm{T}}$ with each vector $\boldsymbol{t}_i$ of dim k. First, three matrices, queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$, and values $\boldsymbol{V}$ are generated in the encoder as follows:

$$\boldsymbol{Q} = \boldsymbol{n}_T \boldsymbol{W}_s^q, \quad \boldsymbol{K} = \boldsymbol{n}_T \boldsymbol{W}_s^k, \quad \boldsymbol{V} = \boldsymbol{n}_T \boldsymbol{W}_s^v, \tag{5}$$

where $\boldsymbol{W}_s^q$, $\boldsymbol{W}_s^k$, $\boldsymbol{W}_s^v$ are the trainable matrices. Using $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, scaled dot-product attention is then applied to determine which elements should be given more attention:

$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\Big(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\Big)\boldsymbol{V} \tag{6}$$

where d_k is the dimension of the keys and queries. Several scaled dot-product attentions are applied in parallel and then linked using multi-head attention to get one output. The multi-head attention is represented as follows:

$$\begin{aligned} &\mathrm{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{Concat}(\mathrm{head}_1, \dots, \mathrm{head}_h)\boldsymbol{W}^O, \\ &\text{with} \quad \mathrm{head}_i = \mathrm{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}_i^V) \end{aligned} \tag{7}$$

where h describes the number of attention heads. Every head is using the scaled dot-product attention, but the weights calculated during training are different, so the multi-head attention summarizes the results of multiple attention mechanisms with different perspectives. Further details can be found in [1].

C. Proposed Neural Network Structure

An overview of the proposed network can be seen in Fig. 1. Several architectures have been tried, including FFNN or LSTM output layers instead of GRU or the multi-dimensional approach from [21], with this architecture showing the best results. The input $\boldsymbol{n}_T \in \mathbb{R}^{(k \times T)}$ of the model is time series data from k onboard vehicle sensors. First, the inputs are fed into a linear layer, which converts the k sensor dimensions into d_{model} dimensions, since the number of inputs for the transformer must be divisible by the number of attention heads. The encoder block weights the time steps according to their relevance using the attention mechanism, while the dimensions remain the same. Afterward, the weighted data is passed into a multi-layer GRU and then broken down to the output $\boldsymbol{o}_T \in \mathbb{R}^{(1 \times T)}$ using two dense layers, representing the predicted $\mu_{\max}$ for T time steps. The network is designed for a sequence length $T = 100$, which corresponds to 1 s using a sampling frequency of 100 Hz. For vehicle dynamics, past causal relationships of $\mu_{\max}$ are not to be expected above this time interval. Thus, from the input $\boldsymbol{n}_T$, a sequence of 100 values is estimated, of which only the last 10, representing 0.1 s, are used. This is done to get a recurrent system, allowing the network to include information from the last 90 time steps for the current estimation. The implementation of all algorithms has been carried out with *PyTorch* 1.11 in *Python* 3.8.10.

III. Results on the Simulative Dataset

The simulative dataset is based on a front-wheel drive Dacia Duster modeled in *IPG CarMaker*, which is identical to the vehicle used in section IV. The dataset was simulated as part of a former contribution [5]. In this chapter, first, the simulative dataset is presented, then the training process is explained and finally, the results are discussed.

A. Dataset

For the simulative dataset, independent training, validation, and test data were simulated in *IPG CarMaker* for various driving maneuvers on different road surfaces. The sensors measure the steering angle δ, steering torque M_{s}, braking pressure $p_{\mathrm{b},i}$, wheel speeds $\omega_{\mathrm{t},i}$ with $i = 1, \dots, 4$ indexing each of the four wheels, the longitudinal a_{x}, lateral a_{y}, and vertical a_{z} acceleration as well as roll $\dot{\kappa}$, pitch $\dot{\varphi}$, and yaw rate $\dot{\psi}$. Furthermore the height of the wheels in relation to the vehicle body $z_{\mathrm{H},i}$ is measured. Moreover, the engine torque M_{e} and gear ratio i_{g} are calculated to the overall driving torque M_{d}. In summary, inputs $\boldsymbol{x}$ and output $\boldsymbol{y}$ of the simulative dataset can be expressed as

$$\boldsymbol{x} = [M_{\mathrm{d}}, p_{\mathrm{b},i}, \delta, M_s, a_{\mathrm{x}}, a_{\mathrm{y}}, a_{\mathrm{z}}, \dot{\kappa}, \dot{\varphi}, \dot{\psi}, \omega_{\mathrm{t},i}, z_{\mathrm{H},i}]^{\mathrm{T}}, \tag{8}$$

$$\boldsymbol{y} = \mu_{\max} \tag{9}$$

assuming identical $\mu_{\max}$ in longitudinal and lateral direction for all four wheels. Gaussian white noise was added to the inputs to account for measurement noise as shown in [5]. As the data in real vehicles are available on the controller area network (CAN) bus with a frequency of 100 Hz, this is also implemented for the simulated data.

Since $\mu_{\max}$ is only observable during vehicle dynamic excitation [2], maneuvers with longitudinal and lateral excitation were simulated with a varying $\mu_{\max}$ from 0.2 (icy road) to 1.0 (dry concrete). For training, a total of 4834 longitudinal maneuvers were simulated including

- accelerations with different trapezoidal engine torques, varying initial velocities, and acceleration durations,
- braking maneuvers with trapezoidal braking pressure for varying initial velocities, braking times, and pressures,
- acceleration and braking from different start to end velocities using the IPG driver in three different modes (defensive, normal, aggressive) on μ jump roads,

and 4048 lateral maneuvers including

- sine steering with varying amplitudes and frequencies,
- slalom with different cone spacings,
- lane changes with varying lateral offsets, and
- left or right turns from different start to end velocities.

Overall, 8882 driving maneuvers were used for training and an additional 20 for validation. A further 961 maneuvers were used for final testing and comparison with model-based approach from [12] and data-based approaches from [5].

B. Training

In this section, the procedure for training the TNN on the simulated dataset is presented. Adaptive Moment Estimation (ADAM), as presented by [27], is used for gradient descent optimization of the TNN weights. The loss function that is minimized during training is the mean square error (MSE) between each actual $\boldsymbol{y} \in \mathbb{R}^{1\times T}$ and estimated $\hat{\boldsymbol{y}}$ value for $\mu_{\max}$.

$$\mathrm{MSE} = \frac{1}{T}\sum_{i=1}^{T}(y_i - \hat{y}_i)^2 \tag{10}$$

Before the driving maneuvers were given into the TNN, each was divided into sequences of $T = 100$ time steps using a rolling window approach with a subsequent step size of 10 to provide continuous $\mu_{\max}$ estimation. As a result, the input tensor of each driving maneuver is extended by one batch dimension b. For example, a driving maneuver with 2000 time steps would be extended from a dimension of $2000 \times k$ to $191 \times T \times k$, corresponding to $b = 191$ batches of the input $\boldsymbol{n}_T$. Each driving maneuver is used for training the network separately. The inputs are min-max normalized to a range from -1 to 1, scaled by training data.

For hyperparameter optimization, the asynchronous successive halving algorithm (ASHA) is used [28], optimizing the number of multi-head attention models, number of encoder layers, number of expected features in the encoder (d_{model}), dimensions of the FFNN in the encoder, sequence length, number of GRU layers, hidden size of GRU layers, learning rate, and dropout rate. For the simulated dataset, a network architecture without GRU layers using only linear layers after the transformer encoder performed best after training for 32 epochs with the parameters shown in Table I. The hyperparameter optimization was performed on four *Nvidia Quadro RTX 6000*. The individual trials of hyperparameter optimization are shown in a parallel coordinate plot in Fig. 2. Each of the lines represents a configuration colored based on its MSE loss over the validation dataset.

C. Results

In this section, the results of the $\mu_{\max}$ estimation for the simulative data using the TNN are presented and compared to the results of a UKF, presented in [12] and two RANNs based on LSTM cells and GRUs, presented in [5].

TABLE I

HYPERPARAMETERS OPTIMIZED BASED ON VALIDATION DATA.

Element	Hyperparameter	Simulative	Experimental
Encoder	Encoder layer	3	4
	Number of heads	4	4
	Dim model	96	64
	Dim feedforward	128	192
GRU	GRU layer	/	2
	Hidden size	/	100
Training	Dropout rate	0.072	0.18
	Learning rate	3.4e−4	2.1e−4

Fig. 4 displays the estimation results of the UKF, LSTM net, GRU net, and TNN on 508 driving maneuvers with longitudinal and 453 maneuvers with lateral excitation. Both the unreduced MSE, which was also used as a loss function during training, and the unreduced mean absolute error (MAE) are shown in the boxplot. Unreduced means that the loss functions were calculated individually for each time step, and then concatenated into one long array. Since no validation dataset has been used in [5] or [12] to date in the collection of UKF, LSTM net, and GRU net results, a TNN (a) is shown, in which the hyperparameters are selected according to the test dataset for better comparability. For TNN (b) a validation dataset is used for choosing the best trial following best practice [29].

In all cases, the model-based approach of the UKF has the highest median MSE and MAE. On the maneuvers with longitudinal excitation, the GRU network performs best with an MSE of 0.0013 and an MAE of 0.036, compared to the TNN (b) with an MSE of 0.013, and an MAE of 0.011. In contrast, on lateral maneuvers, both the TNN (a) and (b) perform best with an MSE of 0.027, and an MAE of 0.16, followed by the RANN based GRUs with an MSE of 0.049 and an MAE of 0.22.

Fig. 3 shows the results of different $\mu_{\max}$ estimations approaches for (a) a longitudinal driving maneuver, and (b) a lateral driving maneuver. The two upper diagrams plot the longitudinal and lateral slip as well as the longitudinal and lateral acceleration. Longitudinal slip is expressed as the ratio of the difference between the wheels velocity and

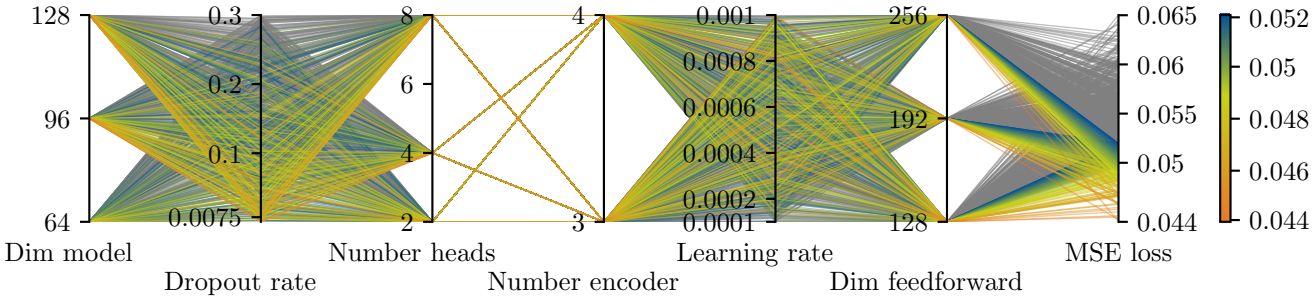


Fig. 2. Results of hyperparameter optimization of the TNN on the simulative dataset with different features, colored by the MSE loss.

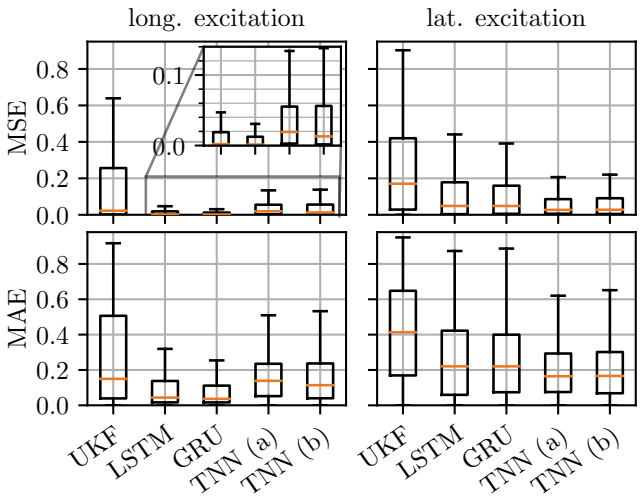


Fig. 4. Comparison of the UKF [12], LSTM net [5], GRU net [5], and the TNN (a) selected based on test dataset, and the TNN (b) selected based on the validation dataset for the estimation of the maximum friction coefficient over a simulative dataset with 508 longitudinal and 453 lateral maneuvers based on the unreduced MSE and MAE.

the vehicles speed to the speed of the vehicle. Lateral slip refers to the angle between the direction the tire is moving in and the direction it is pointing. Finally, the third diagram shows the estimated values for $\mu_{\max}$ using the UKF, LSTM net, GRU net, and TNN compared to the simulated values. The behavior of the TNN on the selected maneuvers can also be observed on other samples of the test dataset and is representative for the dataset.

For longitudinal maneuvers the RANN approaches and the TNN are good at estimating low friction values during sufficient excitation. Moreover, due to their recursion, the LSTM and GRU networks can estimate a previous friction value continuously for a certain time, e.g., from 20 s to 24 s, even without excitation, resulting in less abrupt estimations. The TNN, however, has no recurrence and can only include information of the last 90 time steps in the current estimation, causing difficulties during these situations.

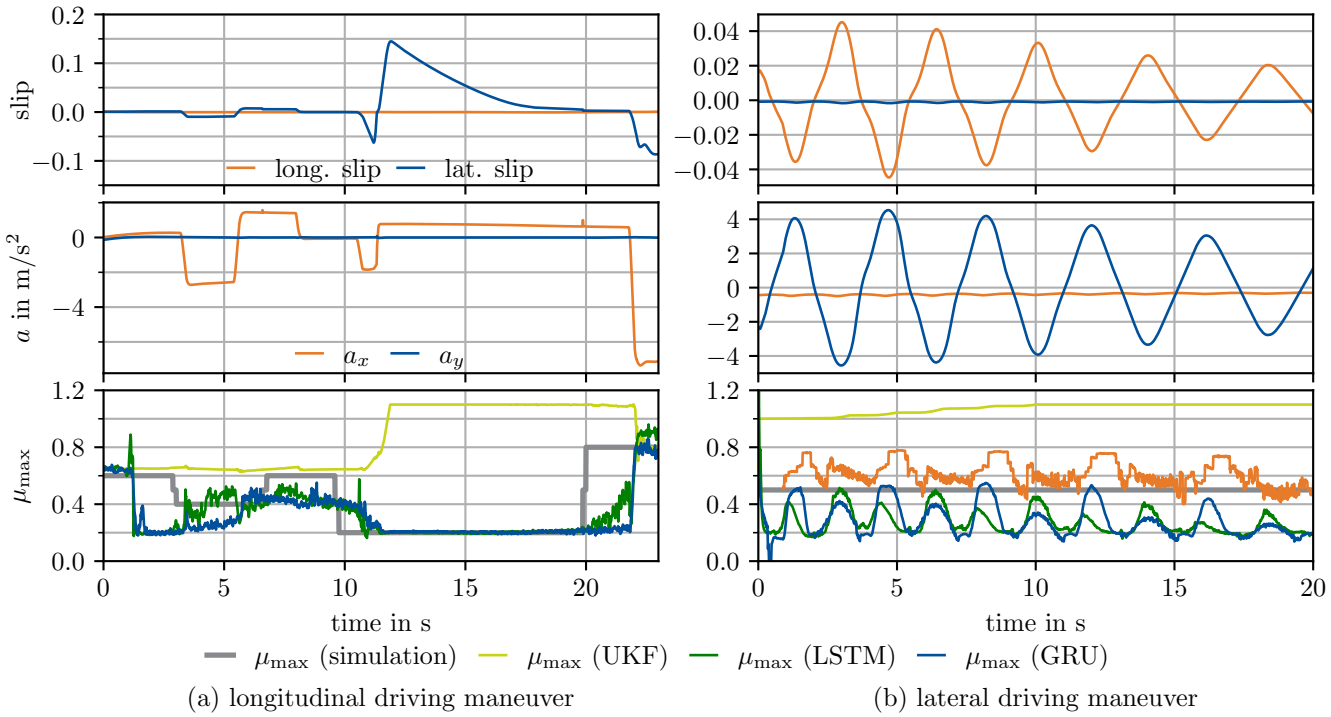
The lateral driving maneuver displayed in Fig. 3 (b) shows a slalom drive on a surface with a constant value for $\mu_{\max}$. It can be seen that the estimation accuracy depends on the excitation, represented by slip and acceleration. Notably, TNN provides the best estimation accuracy with an offset compared to the RANN approaches.

IV. Results on the Experimental Dataset

The experimental dataset was recorded in the context of the work shown in [6] during test drives carried out with a Dacia Duster (Fig. 5, at the ZF WABCO track in Jeversen, Germany. In this chapter, first, the experimental dataset is presented, then the training process is explained, and finally, the results are discussed.

A. Dataset

The test vehicle features an internal combustion engine on the front and an electric motor on the rear axle. Only sensors already installed in production vehicles or under development are used, so the inputs are identical to the simulated dataset with few exceptions. The vehicle is equipped with individual



(a) longitudinal driving maneuver (b) lateral driving maneuver

Fig. 3. Maximum friction coefficient $\mu_{\max}$ estimation using a TNN on simulative data compared to UKF [12] and RANNs based on LSTM cells, or GRU [5] for (a) a maneuver with longitudinal excitation (b) a lateral driving maneuver with curve driving showing the acceleration a and the slip.

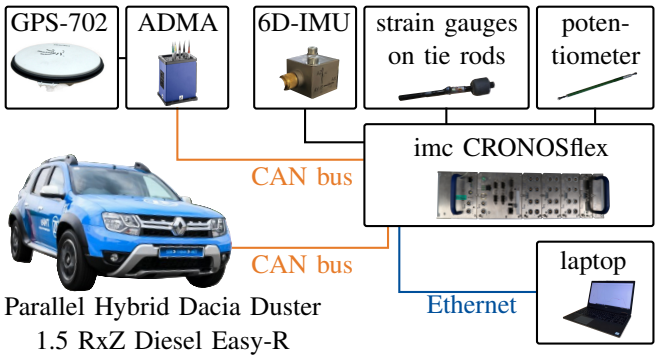


Fig. 5. Test setup: A *Dacia Duster* equipped with a *Genesys ADMA-G-ECO+* inertial platform as a reference sensor system. Measurements of the 6D-IMU, strain gauges, and potentiometers as well as CAN bus data are logged with an *imc CRONOSflex* data acqusition system.

wheel speed sensors, a six degrees of freedom inertial measurement unit (6D-IMU), four potentiometer as height level sensors (HLS), and strain gauges on the tie rods. A low-cost version of the tie rod force sensors is currently under development as shown in [12]. These measure the local strain, which is calculated to the tie rod forces $F_{\mathrm{TR},i}$. Moreover, the engine torques of the front $M_{\mathrm{e,f}}$ and rear axle $M_{\mathrm{e,f}}$, as well as the gear ratio i_{g} are available via the CAN bus, which are converted to the driving torque of the front $M_{\mathrm{d,f}} = i_{\mathrm{g,f}} M_{\mathrm{e,f}}$ and rear axle $M_{\mathrm{d,r}} = i_{\mathrm{g,r}} M_{\mathrm{e,r}}$. In summary, inputs $\boldsymbol{x}$ and output $\boldsymbol{y}$ are expressed by

$$\begin{aligned}\boldsymbol{x} = [&M_{\mathrm{d,f}}, M_{\mathrm{d,r}}, p_{\mathrm{b}}, \delta, a_{\mathrm{x}}, a_{\mathrm{y}}, a_{\mathrm{z}}, \dot{\kappa}, \dot{\phi}, \dot{\psi}, \dots \\ &\omega_{\mathrm{t},i}, z_{\mathrm{H},i}, F_{\mathrm{TR},1}, F_{\mathrm{TR},2}]^{\mathrm{T}},\end{aligned} \tag{11}$$

$$\boldsymbol{y} = \mu_{\max} \tag{12}$$

with all measurements recorded at a sample rate of 100 Hz.

The ZF WABCO test track shown in Fig. 6 provides road surfaces with different friction coefficients, ranging from low $\mu_{\max}$ values, e.g. wet steel, to high $\mu_{\max}$ values, e.g. dry concrete. As $\mu_{\max}$ cannot be measured directly with common sensors, reference values were identified under the assumption that it is constant over the entire road surface for all four wheels. 27 brake maneuvers with anti-lock braking system (ABS) interventions have been executed on each road surface of the test track and an optimization algorithm has been used to identify the reference values for $\mu_{\max}$ by minimizing the error between the estimated and measured

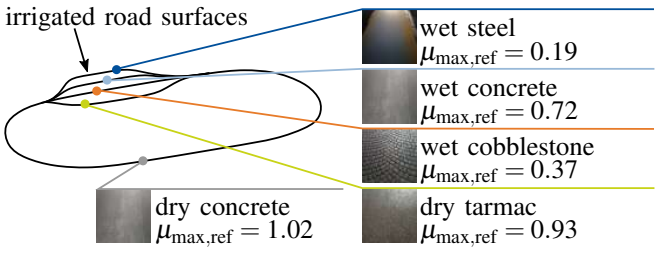


Fig. 6. *ZF WABCO* test track in Jeversen, Germany with the reference maximum friction coefficients $\mu_{\max,\mathrm{ref}}$ for all five road surfaces [12].

longitudinal acceleration a_{x}. Particle swarm optimization in combination with sequential quadratic programming is used as an optimization algorithm. The identification process is carried out 100 times for each surface to ensure convergence and the mean identified values over all iterations are used as ground truth. The determined reference values for $\mu_{\max}$, which are subsequently used, are shown in Fig. 6, mapped on the test track, similar to [6].

For the dataset, a total of 603 driving maneuvers were conducted on the test track at speeds ranging from 0 m/s to 22 m/s with defensive and aggressive driving behavior. The dataset is similar to the simulative dataset and includes 370 longitudinal maneuvers as well as 233 lateral driving maneuvers. Further details on the experimental dataset can be found in [6]. All maneuvers are divided into 509 training, 20 validation, and 74 test maneuvers. The same split has been used for the UKF and RANN results from [6].

B. Training

The training on the experimental dataset is identical to that described in III-B and the optimized hyperparameters are shown in Table I. However, the TNN was trained for 233 epochs instead of 32, mainly due to the size of the datasets. In comparison, the RANNs based on LSTM cells and GRUs presented in [5] were trained for 1000 epochs.

C. Results

In this section the $\mu_{\max}$ estimation results on the experimental dataset using TNN are presented and compared to the results of a UKF, presented in [12], and RANN based on LSTM cells and GRU, presented in [6]. Fig. 8 displays the estimation results on the UKF, LSTM net, GRU net, and TNN on 45 driving maneuvers with longitudinal excitation, and 29 maneuvers with lateral excitation based on the unreduced MSE and MAE.

Similar to the simulated dataset, the model-based approach of a UKF shows the highest median MSE and MAE. Based on the median, the TNN performs best in all cases with a longitudinal MSE of $1{,}5\mathrm{e}-4$, and a lateral MSE of $1{,}7\mathrm{e}-4$, compared to the GRU net with $6{,}8\mathrm{e}-3$ and $5{,}7\mathrm{e}-3$. Noteworthy is that despite the recognizably better median, the TNN still has larger longitudinal maneuver outliers than the GRU net, which can be seen in the upper whisker. In addition, it should be noted that all methods achieve better results on the lateral maneuvers than on the longitudinal ones, probably due to the higher absolute accelerations.

Fig. 7 shows the results of different $\mu_{\max}$ estimations for (a) a longitudinal, and (b) a lateral driving maneuver. The two upper diagrams show the x and y position of the vehicle on the test track, as well as the longitudinal a_{x} and lateral a_{y} acceleration. Finally, the third diagram shows the estimated values for $\mu_{\max}$ using the UKF, LSTM net, GRU net, and TNN compared to the identified reference values.

In the longitudinal driving maneuver, it can be seen that the TNN is able to respond faster to $\mu_{\max}$ jumps as soon as it receives an excitation, which is the case, e.g., at about 50 m, or 160 m. However, this also means that in situations

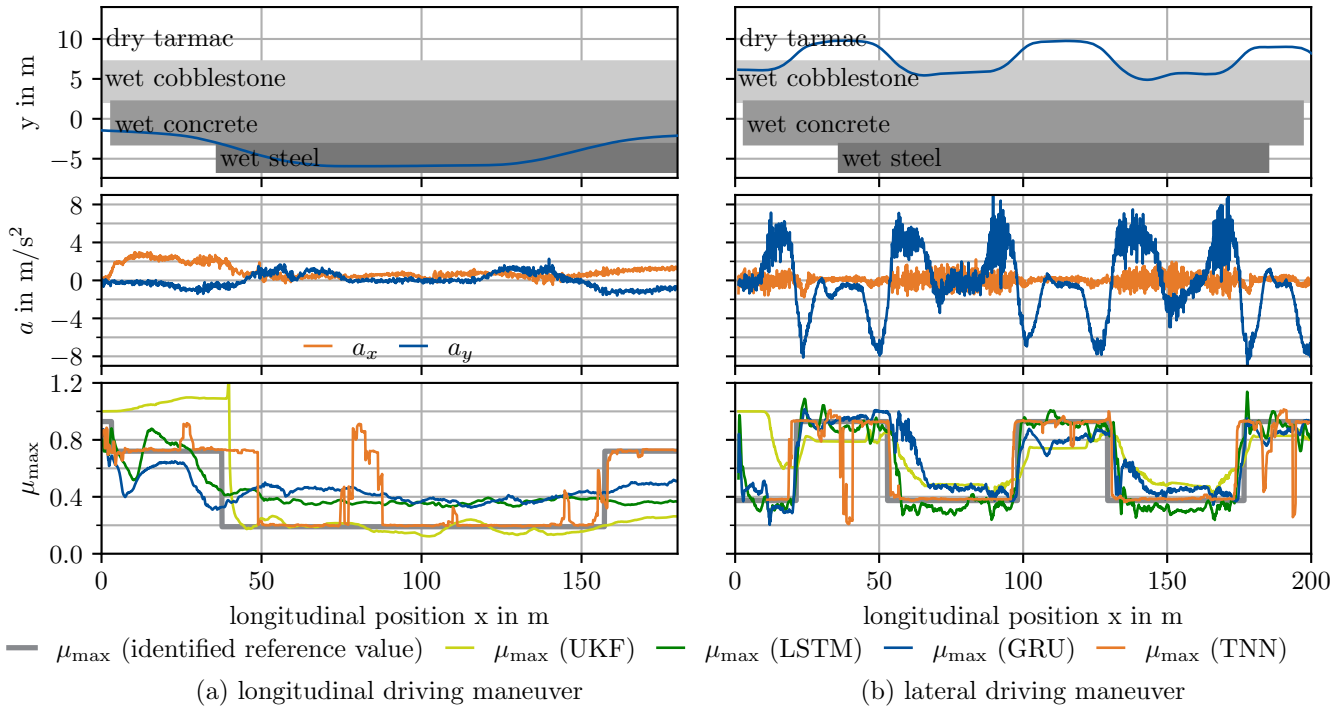


Fig. 7. Maximum friction coefficient μ_{max} estimation using a TNN on experimental data compared to estimations of the UKF [12] and RANNs based on LSTM cells or GRUs [6], showing the vehicles longitudinal x and lateral y position on the test track, and the accelerations a_x and a_y for (a) a maneuver with longitudinal excitation on changing road surfaces, and (b) a lateral maneuver with changing road surfaces.

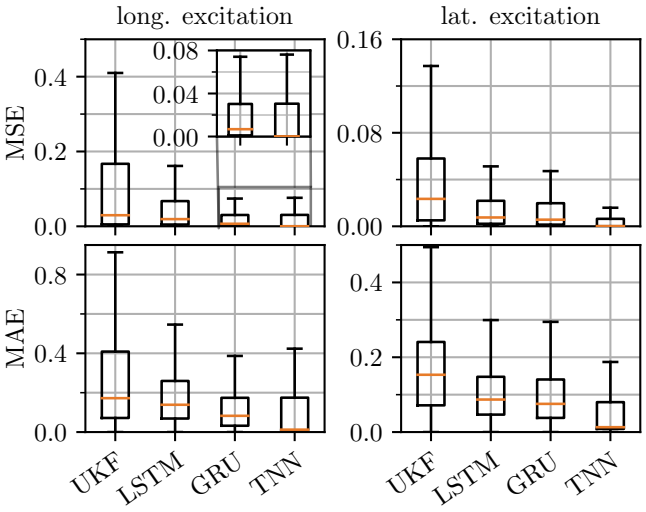


Fig. 8. Comparison of the UKF [12], LSTM net [6], GRU net [6], and TNN for μ_{max} estimation over an experimental test data with 45 longitudinal and 29 lateral driving maneuvers based on the non-reduced MSE and MAE. All driving maneuvers have sufficient driving dynamic excitation $a_x > 2\,\mathrm{m/s}$.

without sufficient excitation, larger outliers in the estimates may occur due to the higher sensitivity, such as at about 80 m. The RANN approaches presented in [6] do not show this problem, since they can continue to estimate the preceding value when there is insufficient excitation over a period of time through recurrence. Overall, the TNN provides the best estimation results for the longitudinal driving maneuver.

During the lateral driving maneuver, it is also the case that the TNN reacts fastest to μ_{max} jumps with some outliers at 40 m and 180 m. It should be noted that for a short period of time at each lane change there will be a mu-split situation which is not reflected in the identified reference value. As a result, friction value jumps may be detected slightly earlier or later. Again, the TNN provides the best, and fastest estimations overall and is particularly accurate in estimating low friction values. In practice, this can help to detect sudden black ice quickly so that ADAS can intervene as soon as possible.

V. Conclusions and Future Work

In this paper, neural maximum friction coefficient μ_{max} estimation based on onboard vehicle sensors is presented using a transformer neural network (TNN). Compared to state of the art approaches from previous works such as a model-based unscented Kalman filter (UKF), or recurrent artificial neural networks (RANN) based on LSTM cells, or GRUs, the TNN achieved improvements on the simulated dataset for lateral, and on the experimental dataset for both lateral and longitudinal driving maneuvers. In particular, the TNN responds faster and better to changes in μ_{max}, although it has fewer parameters and requires fewer training epochs compared to the RANN. Due to the high sensitivity and lack of recurrence, this also means that the TNN can sometimes have larger outliers compared to the other approaches, especially during low vehicle dynamic excitation.

For this reason, and given that the TNN is a black box

model, combining a TNN with excitation monitoring should be considered in the future. In terms of practical application, it should be considered that overestimating $\mu_{\max}$ can have serious consequences. In addition, the behavior of the neural network in case of sensor failure should be tested. Also, the computation times of the TNN in prediction should be compared with other approaches and investigated for real-time capability. In addition, a recurrent transformer structure using the encoder-decoder architecture of [1] could be investigated. In order to avoid overfitting, additional independent experimental test data should be collected on a different test track not included in the training data. Furthermore, other test vehicles, and additional variations such as incline or cross slope maneuvers should be included.

ACKNOWLEDGMENT

The authors would like to thank the Dr. Jürgen and Irmgard Ulderup foundation for funding this project and ZF Friedrichshafen AG for their support during the test drives.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[2] M. Acosta, S. Kanarachos, and M. Blundell, "Road friction virtual sensing: A review of estimation techniques with emphasis on low excitation approaches," *Applied Sciences*, vol. 7, no. 12, 2017. [Online]. Available: https://www.mdpi.com/2076-3417/7/12/1230

[3] S. Khaleghian, A. Emami, and S. Taheri, "A technical survey on tire-road friction estimation," *Friction*, vol. 5, no. 2, pp. 123–146, 2017.

[4] M. Wielitzka, S. Eicke, A. Busch, M. Dagen, and T. Ortmaier, *Unscented Kalman filter for combined longitudinal and lateral vehicle dynamics: Proceedings of the 13th International Symposium on Advanced Vehicle Control (AVEC' 16), Munich, Germany, 13–16 September 2016*, 12 2016, pp. 515–520.

[5] N. Lampe, Z. Ziaukas, C. Westerkamp, and H.-G. Jacob, "Estimation of maximum friction coefficient using recurrent artificial neural networks," in *Proceedings of the 2022 3rd International Conference on Robotics Systems and Vehicle Technology*. New York: ACM, 2022. [Online]. Available: https://doi.org/10.1145/3560453.3560459

[6] N. Lampe, K.-P. Kortmann, and C. Westerkamp, "Neural network based tire-road friction estimation using experimental data," *Accepted for publication in Journal of Modeling, Estimation and Control Conference on 23 January 2023 (MECC 2023)*, 2023.

[7] Z. Ziaukas, A. Busch, and M. Wielitzka, "Estimation of vehicle side-slip angle at varying road friction coefficients using a recurrent artificial neural network," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*, 2021, pp. 986–991.

[8] S. Blume, P. M. Sieberg, N. Maas, and D. Schramm, "Neural roll angle estimation in a model predictive control system," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1625–1630.

[9] W. R. Pasterkamp and H. B. Pacejka, "The tyre as a sensor to estimate friction," *Vehicle System Dynamics*, vol. 27, no. 5-6, pp. 409–422, 1997. [Online]. Available: https://doi.org/10.1080/00423119708969339

[10] T. Song, H. Zhou, and H. Liu, "Road friction coefficient estimation based on bp neural network," in *2017 36th Chinese Control Conference (CCC)*, 2017, pp. 9491–9496.

[11] S. Song, K. Min, J. Park, H. Kim, and K. Huh, "Estimating the maximum road friction coefficient with uncertainty using deep learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3156–3161.

[12] N. Lampe, Z. Ziaukas, C. Westerkamp, and H.-G. Jacob, "Analysis of the potential of onboard vehicle sensors for model-based maximum friction coefficient estimation," in *2023 American Control Conference (ACC)*, 2023, pp. 1622–1628. [Online]. Available: https://doi.org/10.23919/ACC55779.2023.10156574

[13] S. Todorovic, A. Wagner, S. Müller, and J. Neubeck, "Neural network based model for friction potential estimation under longitudinal and lateral excitations," *Journal of Physics: Conference Series*, vol. 2234, no. 1, p. 012005, apr 2022. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2234/1/012005

[14] M. Acosta and S. Kanarachos, "Tire lateral force estimation and grip potential identification using neural networks, extended kalman filter, and recursive least squares," *Neural Comput. Appl.*, vol. 30, no. 11, p. 3445–3465, dec 2018. [Online]. Available: https://doi.org/10.1007/s00521-017-2932-9

[15] A. M. Ribeiro, A. Moutinho, A. R. Fioravanti, and E. C. d. Paiva, "Estimation of tire–road friction for road vehicles: a time delay neural network approach," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 42, no. 1, p. 4, 2019.

[16] X. Zhang and D. Göhlich, "A hierarchical estimator development for estimation of tire-road friction coefficient," *PLOS ONE*, vol. 12, no. 2, pp. 1–21, 02 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0171085

[17] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, G. Rasool, and R. P. Ramachandran, "Transformers in time-series analysis: A tutorial," 2022. [Online]. Available: http://arxiv.org/pdf/2205.01138v1

[18] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," 2023. [Online]. Available: http://arxiv.org/pdf/2202.07125v4

[19] H. Shen, X. Zhou, Z. Wang, and J. Wang, "State of charge estimation for lithium-ion batteries in electric vehicles by transformer neural network and l1 robust observer," in *2022 American Control Conference (ACC)*, 2022, pp. 370–375.

[20] H. Liu, Z. Liu, W. Jia, X. Lin, and S. Zhang, "A novel transformer-based neural network model for tool wear estimation," *Measurement Science and Technology*, vol. 31, no. 6, p. 065106, apr 2020. [Online]. Available: https://dx.doi.org/10.1088/1361-6501/ab7282

[21] Z. Lai, M. Liu, Y. Pan, and D. Chen, "Multi-dimensional self attention based approach for remaining useful life estimation," 2022.

[22] M. Abed, M. A. Imteaz, A. N. Ahmed, and Y. F. Huang, "A novel application of transformer neural network (tnn) for estimating pan evaporation rate," *Applied Water Science*, vol. 13, no. 2, p. 31, 2022.

[23] J. Brotchie, W. Shao, W. Li, and A. Kealy, "Leveraging self-attention mechanism for attitude estimation in smartphones," *Sensors*, vol. 22, no. 22, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/22/9011

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. [Online]. Available: https://aclanthology.org/W14-4012

[26] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey." [Online]. Available: http://arxiv.org/pdf/2009.06732v3

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: http://arxiv.org/pdf/1412.6980v9

[28] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-tzur, M. Hardt, B. Recht, and A. Talwalkar, "A system for massively parallel hyperparameter tuning," in *Third Conference on Systems and Machine Learning*, 2020.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.