2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Neural Network-Based Prediction of Vehicle Energy Consumption on Highways

Dennis Bank[1], Daniel Fink[1], Simon F.G. Ehlers[1] and Thomas Seel[1]

Abstract—**The use of predictive energy management systems can improve the efficiency of multi-energy storage vehicles. However, current systems have limitations, such as short prediction horizons, the requirement for input data that is not publicly available, or the training of the Neural Networks on the routes on which the prediction is made. To overcome these challenges, this paper introduces a novel method for long-horizon energy prediction, utilizing readily available data such as route geometry and traffic information. Our study compares Convolutional Neural Networks (CNNs), Gated Recurrent Units (GRUs), and Transformer Networks optimized using the Asynchronous Successive Halving Algorithm (ASHA). The models were evaluated in a simulated environment using the Simulation of Urban MObility (SUMO) and further tested on real-world driving data, demonstrating that we are able to predict the consumed energy over a 45km stretch of highway with a median RMSE of 0.018 kWh/km for practical application. The energy prediction developed in this study has the potential to enhance predictive energy management systems, thereby optimizing energy usage and contributing to $\mathbf{CO_2}$ emission reduction.**

I. Introduction

Climate change, a pressing global issue, is significantly influenced by the transportation sector, which stands as the third largest CO_2 emitter, following the energy and industry sectors. Specifically, road transport is responsible for over 70% [1] of the sector's CO_2 emissions. This highlights the urgent need for drastic reductions in these emissions as part of our collective efforts to combat climate change. In response to this challenge, multi-energy storage vehicles such as Hybrid Electric Vehicles (HEVs), Fuel-Cell Electric Vehicles (FCEVs), and Supercapacitor Electric Vehicles (SCBEVs) have emerged as a focal point of considerable attention and research interest. These vehicles necessitate a control system to manage power distribution among different energy storages and, in certain cases like HEVs, propulsion types.

The control algorithms currently in use are predominantly reactive. However, there has been a notable surge in research efforts aimed at developing predictive energy management systems (PEMS). These offer the potential for further optimization of the power distribution, thereby reducing CO_2 emissions or battery degradation in the case of SCBEVs [4]. While it has been demonstrated that long prediction horizons are beneficial for PEMS [5], state-of-the-art energy management systems such as [2] typically have short prediction horizons, often between 5–15 seconds (60 seconds at most [3]). This underscores the need for innovative approaches to long-horizon energy prediction.

In this paper, we present a novel approach to vehicle energy prediction in traffic that focuses on full drive cycle segment-wise predictions. The most common approach for PEMS is to predict the vehicle's speed. This highly detailed prediction usually only allows for a prediction of a few seconds. In their paper [4] Wu et. al. have proposed a promising approach for long-horizon predictions of vehicle speed, identifying this as a relevant open research topic. Their method employs comprehensive route knowledge, identifying potential stop points such as traffic signals and classifying typical stop durations for each. While this could be a feasible approach in a future scenario where intelligent road infrastructure communicates with vehicles, the current infrastructure generally lacks this capability. Contrarily, our proposed technique uses readily accessible data, encompassing route geometries and traffic information, to make predictions for routes that have not been previously encountered. To ensure the accuracy of our predictions, we concentrate on forecasting energies over the entire road segment, which results in less detailed data in comparison to the usual speed prediction but enables us to predict over longer horizons. This methodology increases the applicability and versatility of long-horizon PEMS, potentially rendering it a practical tool for real-world situations.

To the best of our knowledge, this is the first study to provide a detailed energy prediction over an extended horizon working on unseen routes. To achieve this, we trained and optimized Convolutional Neural Networks (CNNs), Gated Recurrent Units (GRUs), and Transformer networks. We optimized their topology and hyperparameters using the ASHA algorithm [6].

After training the models in simulated environments using SUMO, we retrained them on approximately 300 km of real-world driving data collected from German country roads and highways. We then tested the models on a continuous 45 km section of a German highway previously unseen by the Neural Networks. This resulted in a network that is able to predict the consumed energy over a continuous 45km highway stretch, which it has not seen before, with a median accuracy of 0.018 kWh/km.

*The authors would like to thank the project manager "Ger- man Aerospace Center" (DLR) for managing the project "RekuTrAx" (project number 01MV22018) and the "Fed- eral Ministry for Economic Affairs and Climate Action of Germany" (BMWK) for the funding.

[1]Institute of Mechatronics Systems, Leibniz University Hannover, 30823 Garbsen, Germany, dennis.bank@imes.uni-hannover.de

II. Dataset Generation and Vehicle Model

In the following section, we explain the process of generating the dataset, employing both the Simulation of Urban MObility (SUMO) and real-world driving data. Additionally, we provide an overview of the architectures of the deployed Neural Networks.

A. Dataset Generation

1) Simulation: SUMO [7], an open-source microscopic traffic simulation model, was used to generate both training and testing data for the artificial Neural Networks. SUMO's ability to simulate each vehicle individually, with distinct driving characteristics and driver profiles, makes it an ideal tool for realistic simulations not only of traffic flow but also of individual vehicles.

The road network in SUMO is represented through a series of interconnected edges and lanes. An edge symbolizes a section of road within the network, defined from one node to another, thereby indicating its direction. Vehicles traverse these edges during a simulation. Each edge can comprise one or more lanes, each of which can accommodate traffic in the direction of the edge. The actual movement of vehicles is simulated within the lanes. SUMO operates as a time-discrete simulation, calculating the state of each vehicle at each step. The stepsize is variable and user-defined. The Traffic Control Interface (TraCI) allows external scripts to control the SUMO simulation and access all vehicle states.

To aggregate vehicle data segment-wise, the consumed energy of each vehicle during each time step is summed up as long as the vehicle remains on the same edge. Once the vehicle leaves that edge, the total energy spent on that edge is stored, and the value for the new edge is reset to zero. A change of lanes on a given edge does not trigger this process. Traffic in SUMO can be simulated by generating random start and endpoints for vehicles. The traffic generation can be influenced by setting a minimum drive distance. After these have been generated, the vehicle routes are calculated; any unreachable routes are discarded from SUMO.

During the simulation, SUMO calculates lane and edge-based traffic measures such as average speed and vehicle density. These measures are always calculated for a predefined time, e.g., every five minutes. To generate the dataset, the vehicle-specific data calculated during the simulation is expanded by the edge-based traffic measures at the beginning of each vehicle's drive after the simulation has concluded.

While SUMO is a state-of-the-art traffic simulation, it can not model all real-world scenarios. For example, illegal driving maneuvers will not be simulated, and neither will weather conditions or the impact of different road surfaces. Additionally, it is important to note that SUMO does not inherently contain a dynamic vehicle model. For this reason, we augment the SUMO simulation with the longitudinal vehicle model introduced in Section II-A2.

2) Longitudinal Vehicle Model: This study utilizes a longitudinal vehicle model, which is based on the hybrid-electric Golf GTE as it is also the test vehicle used for the real-world drives, described in Section II-A3. Several key equations define the model. The overall moment of the vehicle's tires is represented by

$$M_{\text{tire}} = r_{\text{dyn}}(F_{\text{Fr}} + F_{\text{Air}} + F_{\text{Slope}} + F_{\text{Accel}}), \tag{1}$$

where M_{tire} is the moment the tires experience, r_{dyn} is the dynamic tire radius, F_{Fr} is the force caused by the tire friction, F_{Air} the force caused by wind resistance, F_{Slope} the force caused by a slope of the road and F_{Accel} is the force caused by accelerating or decelerating the vehicle.

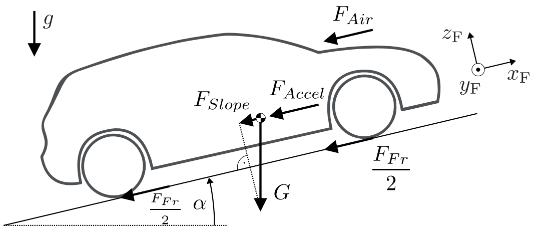


Fig. 1: Free-body diagram of longitudinal vehicle model

The force caused by tire friction, F_{Fr} is calculated as

$$F_{\text{Fr}} = c_r \cdot m \cdot g \cdot cos(\alpha), \tag{2}$$

with c_r the coefficient of rolling resistance, m the mass of the vehicle, g the gravitation constant and α the slope angle.

F_{Air} is the force caused by wind resistance.

$$F_{\text{Air}} = c_W \cdot A_{XF} \cdot \frac{\rho}{2} \cdot v_{xF}^2 \tag{3}$$

where c_W is the drag coefficient, A_{XF} the frontal area of the vehicle, ρ the air density and v_{xF} the longitudinal vehicle speed. The slope angle of the road is accounted for by

$$F_{\text{Slope}} = m \cdot g \cdot sin(\alpha) \tag{4}$$

and the acceleration or deceleration of the vehicle with the formula

$$F_{\text{Accel}} = m \cdot a_{xF} \cdot e, \tag{5}$$

with a_{xF} the longitudinal acceleration and e a factor for rotating masses. The parameter values for the Golf GTE were either taken from the datasheet of the vehicle or experimentally determined as in [8]. The different values are shown in Table I.

Parameter	Description	Value	Unit
ρ	Air Density	1.225	kg/m³
A_{XF}	Frontal Area	2.19	m²
c_W	Drag Coefficient	0.27	—
m	Vehicle Mass	1814.23	kg
e	Rotational Mass Factor	1.109	—
c_r	Rolling Resistance Coefficient	0.013	—

TABLE I: Longitudinal vehicle model parameter of the test vehicle Golf GTE

3) Real World Data: Real-world test drives were conducted to validate the findings derived from the simulation data, and the corresponding vehicle data was stored. The test vehicle utilized for this purpose was a 7th generation hybrid electric Golf GTE, as depicted in Figure 2. The vehicle's current speed v_{xF} and acceleration a_x were recorded at a frequency of 50 Hz. Additionally, the vehicle's GPS position was captured every 1.5 seconds. To synchronize all signals to the same frequency, the GPS data was resampled to match the 50 Hz signals. All signals were taken from the vehicle's internal sensors by accessing the CAN-Bus, reading the corresponding signals, and storing them using an ETAS.

Fig. 2: Test Vehicle and measuring hardware

Post-drive, the vehicle states were cross-referenced with route information using the HereMaps API, which provides both route and traffic data. Upon completion of the drive, the start and end points of the route were used with HereMaps to compute the route between them. If discrepancies were found between the calculated route and the actual test drive, waypoints were iteratively added until full alignment was achieved. Given that the HereMaps API has a maximum byte limit for its requests, it is crucial to minimize the number of necessary waypoints.

Once the GPS trace and the HereMaps route were aligned, the data from each corresponding route segment was combined with the vehicle drive dataset. This was achieved by associating each GPS point with the nearest route segment from the HereMaps route. Lastly, the energies on each route segment were calculated using the measured speed and acceleration as input for the dynamic longitudinal model described in Section II-A2. The calculated torque is used to calculate the power in each timestep. The power will then be integrated over the time the vehicle spends on each route segment, resulting in the energy consumption for each segment, which will later be used as the labels for the neural networks. Now that the generation of simulation real-world data has been covered, all the necessary inputs and outputs for the Neural Networks have been described. In the next section, we'll go over the fundamental structure of these networks in order to explain the differences in performance later on.

B. Neural Network Architectures

A variety of neural network architectures have been proposed and developed to solve the problem of sequence-to-sequence predictions. This paper will concentrate on examining the prediction accuracy among three primary classes: Convolutional Neural Networks, Gated Recurrent Units, and Transformer-based Networks.

1) Convolutional Neural Networks: CNNs have seen widespread adoption in the field of image processing but have also seen usage in sequence-to-sequence predictions. CNNs operate by applying a series of filters to the input sequence. Each of these is optimized during training. The CNN applies these filters across the entire input sequence, generating feature maps. These feature maps are then passed through a non-linear activation function, such as ReLU, to introduce non-linearity into the model. This process is repeated for multiple layers, extracting more abstract features from the input data. After several convolutional layers, the output is passed through one fully connected layer to produce the final one-dimensional output sequence. As the filters are convoluted with the input, the output dimension changes. In order to keep the same output length as the input data, the input needs to be padded with the size

$$P = \frac{K-1}{2} \tag{6}$$

with P being the padding size and K the kernel size, assuming that the stride is 1. Our proposed CNN structure is shown in Figure 3. It is comprised of multiple parallel paths with different kernel sizes in order to aid the learning of short and long-term importance of features.

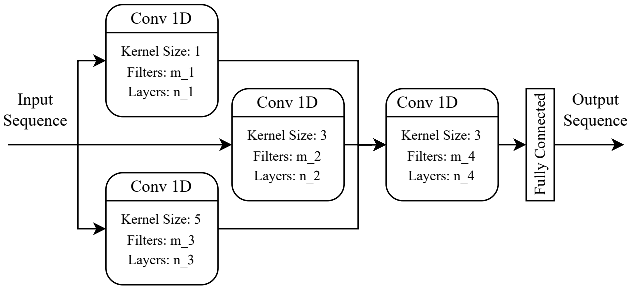


Fig. 3: Structure of the used CNN with multiple paths of different kernel sizes to encourage the learning of features of individual segments as well as larger scale features that influence the network over multiple segments

2) Gated Recurrent Units: GRUs [9] are a type of Recurrent Neural Network (RNN) architecture that has shown significant effectiveness in sequence-to-sequence prediction tasks. Unlike traditional RNNs, GRUs incorporate gating mechanisms that modulate the flow of information through the network. These gates allow GRUs to mitigate the vanishing gradient problem, a common issue in training traditional RNNs, thereby enabling the network to capture long-term dependencies in sequence data. In a sequence-to-sequence prediction task, a GRU model takes a sequence of input data and generates a corresponding output sequence. The gating mechanisms in the GRU control the information flow between consecutive time steps. This makes GRUs particularly suited for tasks where the prediction at a given time step depends on the context from both past and future steps. [9] The GRU cell has two gates: an update gate and a reset gate.

The update gate helps the model determine how much of the past information needs to be passed along to the future, while the reset gate decides how much of the past information to discard.

The update gate $\mathbf{z_t}$ is computed as follows:

$$\mathbf{z_t} = \sigma(\mathbf{W_z} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_z}), \quad (7)$$

where σ denotes the sigmoid function. The update gate determines the extent of information from the previous hidden state $\mathbf{h_{t-1}}$ that should be used for computing the candidate hidden state.

The reset gate ($\mathbf{r_t}$) is given by

$$\mathbf{r_t} = \sigma(\mathbf{W_r} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_r}). \quad (8)$$

This gate decides the amount of information from $\mathbf{h_{t-1}}$ that should be discarded.

The candidate hidden state $\tilde{\mathbf{h}}_t$ contains potential values for the new hidden state at time t, and is calculated as

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W} \cdot [\mathbf{r_t} * \mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b}), \quad (9)$$

where $*$ denotes element-wise multiplication.

Finally, the new hidden state ($\mathbf{h_t}$) at time t is computed as

$$\mathbf{h_t} = (1 - \mathbf{z_t}) * \mathbf{h_{t-1}} + \mathbf{z_t} * \tilde{\mathbf{h}}_t. \quad (10)$$

Here, $\mathbf{x_t}$ represents the input at time t, while $\mathbf{W_z}$, $\mathbf{W_r}$, and $\mathbf{W}$ are weight matrices to be learned for the update, reset, and candidate hidden state respectively. Similarly, $\mathbf{b_z}$, $\mathbf{b_r}$, and $\mathbf{b}$ are bias terms to be learned for these states.

3) Transformer: Transformer networks are model architecture that relies entirely on an attention mechanism to draw global dependencies between input and output. They have been firmly established as state-of-the-art approaches in sequence modeling and transduction problems [10].

The Transformer model is based on a sequence-to-sequence framework, where the input and output are sequences. The model consists of an encoder and a decoder, both of which are composed of multiple identical layers containing self-attention and feed-forward Neural Networks, as can be seen in Figure 4.

The self-attention mechanism in the Transformer model allows it to weigh the importance of tokens within the sequence, thereby capturing the dependencies between tokens regardless of their distance from each other. [11]

The input matrix $\mathbf{X}$ is first transformed into query, key, and value matrices using learned weight matrices $\mathbf{W_Q}$, $\mathbf{W_K}$, and $\mathbf{W_V}$ respectively, as shown below:

$$\mathbf{Q} = \mathbf{X}\mathbf{W_Q},$$
$$\mathbf{K} = \mathbf{X}\mathbf{W_K},$$
$$\mathbf{V} = \mathbf{X}\mathbf{W_V}.$$

The scaled dot-product attention mechanism then computes the attention scores as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

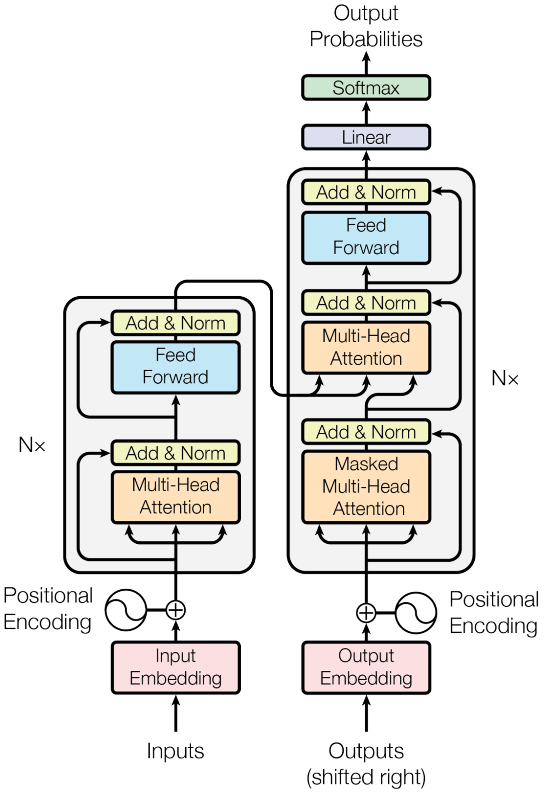


Fig. 4: The Transformer Network processes all inputs as one, which allows it to use information from future parts of the data, in comparison to sequential models, that can only rely on past information [11]

In the Multi-Head Attention mechanism, multiple attention heads are concatenated and linearly transformed:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, ..., \text{head}_h)\mathbf{W_O}$$

Each attention head is computed as:

$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W_{Qi}}, \mathbf{K}\mathbf{W_{Ki}}, \mathbf{V}\mathbf{W_{Vi}})$$

Finally, the Position-wise Feed-Forward Networks are applied to the output of the multi-head attention layer:

$$\text{FFN}(x) = max(0, x\text{W}_1 + b_1)\text{W}_2 + b_2$$

Compared to GRUs, Transformer networks offer several advantages. They process all tokens in the sequence simultaneously, which allows for efficient handling of long-range dependencies and parallelization of computations. Furthermore, the self-attention mechanism in Transformers assigns weights to tokens within the sequence based on their importance, capturing dependencies between tokens irrespective of their distance from each other. To compensate for their lack of recurrence, Transformers incorporate positional encoding to inject order information into the input [11].

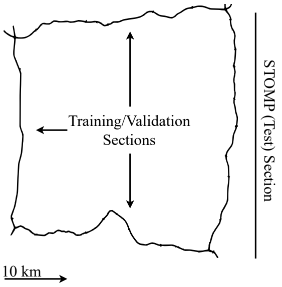
III. Experiments and Results

A. Simulation

1) Dataset: The initial phase of this study involved the generation of simulation training data for the Neural Networks, utilizing SUMO as outlined in Section II-A1. The road network employed for data generation was derived from the open-source Stuttgart Open Motorway Project (STOMP) [12]. This project accurately models traffic over a 24-hour period on a 45 km section based on traffic data from German traffic authorities. The vehicle population in the simulation is categorized into four classes: passenger cars, motorbikes, trucks, and buses, each with their respective acceleration profiles.

The STOMP road network, depicted in Figure 5, consists of four large segments of German highways. However, traffic is only modeled on one of these four segments. To augment the training data, traffic was synthetically generated by creating random routes through the remaining three segments. In order to simulate varying traffic densities, the number of departing vehicles in the SUMO simulation was progressively increased over time for the synthetically generated data. The original four vehicle classes from the STOMP data set were retained, and the distribution of vehicle types was aligned with the data from STOMP.

The data generated on the three segments of the road network served as the training set for the models, while the vehicle data from the STOMP dataset, grounded in real-world traffic, was utilized as the validation set for the simulation. Given that the networks were not exposed to any portion of the highway used for validation during their training, it can be ensured that there was no information leakage. The split between the training, validation, and test dataset in SUMO, as well as an overview of the input features, is shown in Figure 5.



Features	Unit
Edge Length	m
Avg. Speed	m/s
Allowed Speed	m/s
Lane Curvature	rad
Lane Slope	rad
$\frac{\text{Avg.Speed}}{\text{AllowedSpeed}}$	/

Fig. 5: Topology of the SUMO road network together with the input-data used for each segment as input for the Neural Networks

2) Training: Three neural network architectures are compared in this study: CNNs, GRUs, and Transformer networks. The Networks were built using PyTorch, and to improve their prediction quality, their hyper-parameter were tuned using the Asynchronous Successive Halving Algorithm (ASHA), which is an optimization technique that operates by sampling a large number of configurations from the hyper-parameter space and allocating each a small amount of resources. Configurations that perform poorly are discarded early, freeing resources for more promising configurations. The top-performing configurations are then given more resources for further training. [6]

The structure of the input data is shown in Figure 5; the label data consisted of the log of the energy used on the specific road segment; this helped with the convergence during training as the energy consumed on one road segment can vary widely making standard normalization unfeasible. To train the models, a domain-specific loss function was created to further aid with the convergence of the models. It is calculated as

$$loss = \text{mse}(\mathbf{y}, \hat{\mathbf{y}}) + \text{mse}(\sum_{i=1}^{n} 10^{y_i}, \sum_{i=1}^{n} 10^{\min(\hat{y}_i, 5)}). \quad (11)$$

For the predictions, the cumulative sum is calculated with $10^{\min(\hat{y}_i,5)}$ and not $10^{\hat{y}_i}$ because the loss during early training could cause an overflow.

With this loss function, the networks are also incentivized to predict potential energy when it is not clear on which of the route's segments exactly that energy will be needed.

The CNNs that were optimized had the structure shown in Figure 3, the optimizable features were the decision of which paths to activate, the number of layers for each convolution block, the kernel size for the final convolution block, the number of filters and the learning rate. An overview of the parameters is given in Table II.

Hyperparameter	**Tuning Range**
Learning Rate	Distribution between 1×10^{-4} and 1×10^{-1}
Filters	Choice among 2, **4**, 8, 16, 32
Kernel Sizes	Choice among **1**, **3**, 5
Number of Layers	Choice among **1**, 2, **4**, 8
Path Activation	Choice among True, False

TABLE II: CNN Hyperparameters Searchspace

The optimal CNN structure on the simulation data used the path with kernel sizes one and three, with four filters. The path with kernel size one consisted of one layer; the other path, as well as the final convolution block, used four layers.

For the GRU networks, the learning rate, number of layers, and hidden dimensions were optimized as shown in Table III. The best-found model consisted of two layers with sixty-four hidden dimensions each.

Hyperparameter	**Tuning Range**
Learning Rate	Distribution between 1×10^{-4} and 1×10^{-1}
Hidden Dimensions	Choice among 2, 4, 8, 16, 32, **64**
Number of Layers	Choice among **2**, 4, 8, 16, 32

TABLE III: GRU Hyperparameters Searchspace

Finally, the transformer networks were optimized. As before, the learning rate and number of layers were optimized, as well as the number of attention heads, as shown in Table IV. The chosen network used two attention heads and layers. After finding an optimized structure for all network types, these networks were trained on the SUMO dataset and evaluated using the STOMP dataset.

The results are shown in Figure 6. GRUs and CNNs perform comparably, but the Transformer performs best on the simulation data.

Hyperparameter	Tuning Range
Learning Rate	Distribution between 1×10^{-4} and 1×10^{-1}
Number of Heads	Choice among 1, **2**, 3
Number of Layers	Choice among 1, **2**, 4, 8, 16, 32

TABLE IV: Transformer Hyperparameters Searchspace

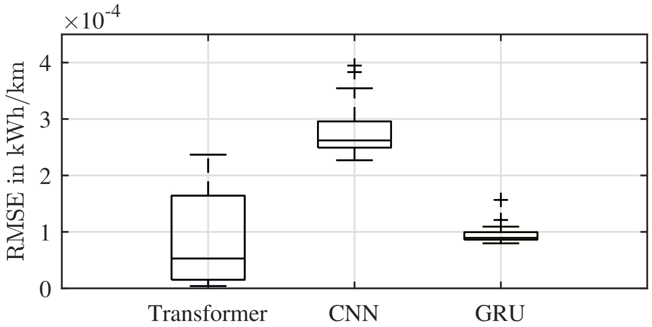


Fig. 6: RMSE in kWh/km of the three Networks for the 50 longest routes over the test section in SUMO. The Transformer Networks perform best, followed by the GRUs and the CNNs.

B. Real World Data Validation

1) Dataset: To ensure the accuracy and reliability of the simulation results, a validation process was undertaken using real-world driving data. This data was collected as outlined in Section II-A3, providing a dataset for model testing.

The dataset comprises five individual test drives, which collectively account for 345 kilometers of driving data. These test drives were conducted across various days and times, ensuring a diverse range of conditions and scenarios were captured. Furthermore, different routes were driven by different drivers. This diversity is crucial in testing the model's robustness under different circumstances.

Driver	Road Type	Length
Driver A	Highway	28.2 km
Driver A	Highway	28.9 km
Driver B	Highway and Country Road	190.6 km
Driver C	Highway and Country Road	51.6 km
Driver D	Highway	45.9 km

TABLE V: Overview of the driver, length, and road types of the five taken real-world drives.

Four of the five test drives, consisting of 300 km, were split up into training and validation data. The last drive was chosen as test data, as no part of it overlapped with any of the other drives. The training drives were utilized to retrain the pre-existing models that were initially trained as described in Section III-A. This retraining process allowed the models to learn and adapt based on real-world driving data, thereby enhancing their prediction accuracy.

The final evaluation of the model was carried out using the test drive. It consists of a 45-kilometer segment of the German highway, a route not included in any part of the training process. This drive was chosen as the driver did none of the drives in the training or validation data set, and no part of the test route was in them.

2) Results: For the evaluation of the network, we split the 45km into four parts with a length of 10km each. The results can be seen in Figure 7.

The Transformer network had better predictions for sections one and three, while the CNN outperformed it in Section 2. In Section 4, both networks had comparable results. The GRU, however, was, in all cases, the worst network. This may be due to the sequential structure of the model, as it does not allow the GRU to take into account future segments, while the CNN, due to its kernel size and the Transformer network, can.

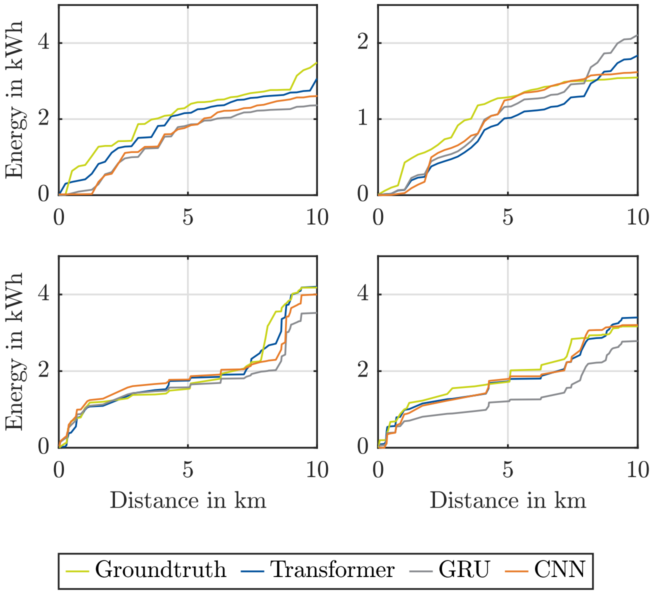


Fig. 7: Results for 10 km predictions, while Transformer Networks and CNNs perform comparably, GRUs are consistently worse

To further validate that the networks are able to make long-term predictions, we additionally made a prediction for the whole 45km route. The resulting predictions can be seen in Figure 8.

During this test, the limitations of the CNNs and GRU Networks become visible. They are not able to make accurate predictions for such long horizons and suffer from drifts. They are still able to predict the relative movement of the energy but consistently under-predict the total amount of drive energy. The Transformer network, on the other hand, is still able to accurately predict the energies because it processes all inputs at once.

To further illustrate the difference in performance, we show the root mean square error of the cumulative predicted energy divided by the total number of driven kilometers so far in Figure 9.

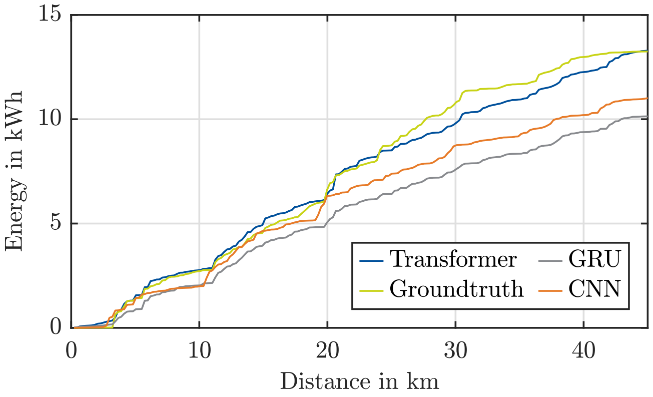


Fig. 8: Results for a 45 km long test. While both Transformer Networks and CNNs performed comparably on 10 km segments, Transformers outperform the CNNs on longer routes

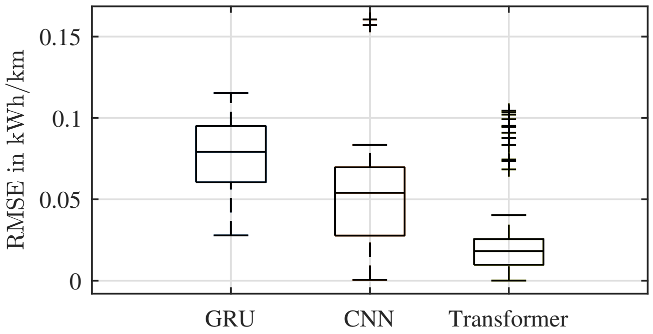


Fig. 9: RMSE in kWh/km of the three Networks over the full 45 km drive consisting of n = 200 individual road segments. With a median RMSE of 0.018 kWh/km or 6.43%, the Transformer model predicts the energies most accurately.

With a median RMSE of 0.018 kWh/km, the Transformer network is the best network for long-horizon energy prediction, showing promising results for future usage in predictive energy management systems. While the GRU network outperformed the CNN on the simulated data, the CNNs performed slightly better on the real-world data. In general, the performance of the networks is better on simulated data than on real-world data. This is to be expected because not all real-world effects can be simulated. Nevertheless, the real-world results shown in this paper are still encouraging.

IV. Conclusions

In conclusion, this paper has successfully demonstrated the feasibility of long-horizon prediction of driving energies. We have trained and evaluated GRUs, CNNs, and transformer-based networks. To train them, we created a domain-specific loss function to improve the prediction accuracy. We have shown that it is possible to predict energy consumption in a simulated environment and, promisingly, real-world conditions. Our approach is novel as it works with only publicly available data from the HereMaps API. This means that our method can be widely applied without requiring specific knowledge of the route, making it a practical and accessible tool for energy prediction.

The energy prediction developed in this paper has the potential to enhance predictive energy management systems, as the limited prediction currently hinders their progress. By improving these systems, energy usage in the transportation sector can be further optimized. This, in turn, would lead to a reduction in CO_2 emissions, aiding global efforts to combat climate change.

While our findings are promising, they necessitate further validation through extensive real-world driving tests. In the future, this method can be expanded using driving data from inner-city traffic to increase its coverage of driving scenarios, and the current prediction in the spatial domain can be transformed into a temporal domain prediction.

References

[1] Hannah Ritchie (2020) - ”Cars, planes, trains: where do CO2 emissions from transport come from?”. Published online at OurWorldInData.org. Retrieved from: ’https://ourworldindata.org/co2-emissions-from-transport’ [Online Resource]

[2] Jie Han, Hong Shu, Xiaolin Tang, Xianke Lin, Changpeng Liu, Xiaosong Hu, Predictive energy management for plug-in hybrid electric vehicles considering electric motor thermal dynamics, Energy Conversion and Management, Volume 251, 2022, 115022, ISSN 0196-8904, https://doi.org/10.1016/j.enconman.2021.115022.

[3] 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,” in IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp. 14657-14673, Dec. 2020, doi: 10.1109/TVT.2020.3032989.

[4] Yue Wu, Zhiwu Huang, Yusheng Zheng, Yongjie Liu, Heng Li, Yunhong Che, Jun Peng, Remus Teodorescu, Spatial–temporal data-driven full driving cycle prediction for optimal energy management of battery/supercapacitor electric vehicles, Energy Conversion and Management, Volume 277, 2023, 116619,

[5] Yang Zhou, Alexandre Ravey, Marie-Cécile Péra, A survey on driving prediction techniques for predictive energy management of plug-in hybrid electric vehicles, Journal of Power Sources, Volume 412, 2019, Pages 480-495, ISSN 0378-7753, https://doi.org/10.1016/j.jpowsour.2018.11.085. (https://www.sciencedirect.com/science/article/pii/S0378775318313405)

[6] Li, Liam and Jamieson, Kevin and Rostamizadeh, Afshin and Gonina, Ekaterina and Hardt, Moritz and Recht, Benjamin and Talwalkar, Ameet ”A System for Massively Parallel Hyperparameter Tuning” (2018) arXiv preprint arXiv:1810.05934

[7] Michael Behrisch, Laura Bieker, Jakob Erdmann, Daniel Krajzewicz, SUMO – Simulation of Urban MObility: An Overview, Institute of Transportation Systems German Aerospace Center, 2011, https://doi.org/10.48550/arXiv.1810.05934.

[8] Fink, D.; Shugar, S.; Ziaukas, Z.; Schweers, C.; Trabelsi, A. and Jacob, H. (2022). Energy Demand Prediction in Hybrid Electrical Vehicles for Speed Optimization. In Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS; ISBN 978-989-758-573-9; ISSN 2184-495X, SciTePress, pages 116-123. DOI: 10.5220/0011075600003191

[9] Chung, Junyoung; Gulcehre, Caglar; Cho, KyungHyun; Bengio, Yoshua (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv preprint arXiv:1412.3555. Available at: https://arxiv.org/abs/1412.3555

[10] Tianyang Lin, Yuxin Wang, Xiangyang Liu, Xipeng Qiu, A Survey of Transformers, arXiv preprint arXiv:2106.04554, 2021, https://doi.org/10.48550/arXiv.2106.04554. (https://arxiv.org/abs/2106.04554)

[11] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. and Polosukhin, I. (2017). Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems - NIPS’17; arXiv:1706.03762

[12] D. Förster, H. Löhr, A. Grätz, J. Petit and F. Kargl, ”An Evaluation of Pseudonym Changes for Vehicular Networks in Large-Scale, Realistic Traffic Scenarios,” in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 10, pp. 3400-3405, Oct. 2018, doi: 10.1109/TITS.2017.2775519.