Driving Style Analysis Using Recurrent Neural Networks with LSTM Cells

Samuel Würtz and Ulrich Göhner
University of Applied Sciences Kempten, Germany
Email: {samuel.wuertz, ulrich.goehner}@hs-kempten.de

***Abstract*—Many publications work on optimization of driving styles in motor vehicles. Most conclude that they can improve energy efficiency through training. In recent years the tools to address those problems evolved towards machine learning. To get appropriate data for learning algorithms we developed a method to judge a driving style with respect to energy efficiency. This approach leveraged handpicked criteria like acceleration extracted from GPS. Like related works, this method does not scale, since it requires substantial preprocessing. The goal of this evaluation was to reduce the resistance energy of a driven trip, while maintaining a natural traffic flow. This was accomplished by mimicking a low-pass filter on the speed profile. On top excessive speeding gets punished. It was possible to use our data with over 1 million kilometers for training a Recurrent Neural Network. In respect to the RNN the training data was used, to let it map the obtained function. The provided data was adjusted in different stages, until it was only the raw GPS data. The RNN learned to handle most GPS errors, only in initial phases the results are mixed. A RNN Network is well suited to handle GPS data and learn higher level features on its own. The result is a NN which judges the driving style using only raw GPS data.**

***Index Terms*—neural network, deep learning, RNN, LSTM, machine learning, GPS data, sensor data, driving style, driver behavior, intelligent vehicle control, energy efficiency, driving safety**

I. INTRODUKTION

A fundamental challenge of the 21st century, which will affect all of us, is the development of an energy management, which can help to protect the environment [1]. While being energy efficient with personal motor vehicles won't solve the complete problem, the short term consequences can be influenced, because transportation accounts for 30% of the total energy consumption [2], so it is clear that it has a beneficial impact. Therefore it is worthwhile to educate drivers to achieve a more efficient driving style.

The driving style in motor vehicles has a significant effect on two aspects: energy efficiency and road safety. Increased road safety has a big social and economic benefit while energy efficiency has a positive impact on the environment, by reducing CO_2 emissions. There has been previous work which analyzed the effect of different driving styles on fuel consumption [3]. The authors of [4] found that the difference in fuel consumption (and thus energy efficiency) between a normal driving style and a fast aggressive driving style is above 40%. Statistics derived from the electric car fleet of our University suggest that the range of an electric vehicle can be increased by up to 50% only by maintaining an energy efficient driving style. Other studies showed that a direct feedback to car drivers' behavior has a positive impact on the subsequent driving style [5], [6]. Considering these points it is worth while improving the drivers' actions regarding energy efficiency. For this purpose, a master thesis at the University of Kempten developed an algorithm to determine the driving style considering energy efficiency and road safety [7]. This algorithm uses x- and y-accelerations as well as speed to make assertions about the efficiency and safety of the driving pattern. The required values get extracted from car GPS data, but since GPS signals in general are not accurate and reliable enough there is a need to preprocess the accumulated data. The main problem arising from incomplete and inaccurate GPS data will be described in this paper later on.

As a consequence of the required filtering and preprocessing procedure the algorithm could not be used to give the car driver a real-time feedback. In our research group at University of Kempten a huge amount of data with recorded tours from numerous different vehicles and drivers has been collected during the last eight years. This data is very diverse, since there are trips from big cities in Germany and also rural areas in southern Germany. Another reason for the usage of this data is the different nature of scenarios. Many cars belong to local craftsmen, others belong to municipal fleets, transport fleet as well as many privately owned vehicles. In total there a more than one million kilometers of recorded tours. This database was used for testing the algorithm and creating labeled data for the machine learning approach. It is expected to get generalizable results due to the diverse nature of the available data.

In this paper we propose a method to utilize raw GPS data to give live scores for a driving style, using Recurrent Neural Networks. The development and implementation was divided into two different steps. The first step was to apply Deep Learning to the preprocessed data and try to learn which patterns of acceleration and speed correspond to good driving styles. This was used to

Manuscript received September 23, 2019; revised December 23, 2019.

© 2020 J. Adv. Inf. Technol.

doi: 10.12720/jait.11.1.1-9

get a feeling whether the algorithm was able to learn the driving-style-score function. Based on this, the next step was to train the Neural Network with the same function from raw GPS data. This was a far more interesting and difficult problem to solve. Since the data preprocessing is very costly the algorithm gets more flexible and can be applied in a more general way, if preprocessing is becoming obsolete. It could be shown that the neural network was able to extract the necessary features directly from latitude and longitude values. Not only could the neural network learn the features, it was also able to identify flawed GPS data and only base its score on real vehicle movements.

The rest of the paper will be structured as follows. The next chapter explores what is generally understood by the term driving style and describes in detail how it is defined in the context of this paper. It is also explained which parameters were identified as most influential for an educated assertion about the driving style. The chapter 3 will give an overview of some related work. In chapter 4 we will describe the problem formulation and discuss the challenges of our chosen approach. The following chapter number 5 will examine the details of the dataset, how the raw data is composed, the patterns which could be seen and how the processing of this data was implemented. It is also described how the labeled data was obtained. Chapter 6 will illustrate the Deep Learning algorithm including the software architecture, algorithmic improvements and implementation. The section about the architecture is supposed to give an impression how the hyper-parameters where chosen. Then the experiments and evaluations will be shown. To wrap things up we will discuss in chapter 7 our results and their possible use cases in other systems.

II. Driving Style

The topic of driving styles is widely discussed in various research papers [8]. Among many others the most widely discussed topics are energy efficiency and road safety. As a result of the varying objectives, different definitions for the term driving style can be found in literature. This chapter will first explore the definitions found in literature and clarify some terms. Then we will formulate our definition of the term driving style in this paper and also describe in more detail how the new evaluation procedure works.

A. Definitions

Since driving style is a complex concept it is necessary to clarify some terminology to make sure the intended meaning of the terms is clearly understood. According to [8] one can differentiate between three influencing areas. Driving conditions (like traffic, weather, vehicle, road, etc.) are the first part, which mostly cannot be influenced by a driver or any driver assistance. But nonetheless an assistance tool could benefit from being aware of those external conditions.

The second part is the driver level, where there are 3 subcategories: driving skill influences the driving style which in turn can lead to a specific driving behavior. This part can be influenced through training of the driver as well as making the driver aware of sub-optimal behavior [9].

The third part is the vehicle level. This part can be divided in driving events (acceleration, turn, break, etc.) which in sum, form the driving pattern (speed profile, lateral acceleration profile). When optimizing the driving pattern, one has to consider the goals of the optimization to achieve good results. Even though it seems plausible that an energy efficient or anticipating driving style also is beneficial for road safety, there are other parameters which need to be considered. For a safe driving style it is not sufficient to consider the speed profile and with that anticipation of the driver. Whereas speed is a big part, the lateral acceleration also needs to be evaluated considering the behavior in curvy parts of the driven route. According to [10] there is a strong correlation between efficiency, anticipation and safety. They could show that training in fuel-consumption-optimized driving also results in safer more anticipating and in general more defensive driving behavior. This suggests that an improvement in driving style is beneficial for various different objectives.

B. Approach

This section will bring our approach in line with the aforementioned definitions. The aim of this work is to develop a procedure which analyzes the driving behavior through the individual driving events. The aim is to give drivers advice on how to optimize their driving style and, in the best case, to increase their personal driving ability in the long term and thus lead to better driving behavior. The driving style classification here should judge a trip regarding energy efficiency. The concentration on power consumption is just a simplification since it is easier to measure opposed to a safe vs. unsafe driving pattern.

Following are three analyzes which are available for the evaluation of the driving style:

i. Rating for efficient speed
ii. Rating for efficient positive longitudinal acceleration
iii. Rating for efficient negative longitudinal acceleration

In order to come up with a general way to judge a driving event, based on speed and acceleration we analyzed the main forces which oppose the vehicle during driving. Those are acceleration resistance, air drag and rolling resistance. Since the last one of those cannot be influenced by the driver we chose to ignore it in our analyzes. In order to evaluate the influence of the different factors we constructed very basic driving scenarios. The benchmark data were generated as follows: Our research car, a BMW i3 equipped with appropriate measurement and data acquisition hardware for the experiment, was used for different test drives. The driving duration was 60 seconds and the car was accelerated to a given speed. This target speed was maintained until the end of the test drive. The variables for the different experiments were acceleration and target speed. It was found that the acceleration duration didn't influence the energy consumption directly (for simplicity of the experiment the state of charge was used to measure

© 2020 J. Adv. Inf. Technol.

the energy consumption). Because the acceleration resistance is always the same to reach a specific speed, no matter how high the acceleration is. The second finding was, that the acceleration resistance plays a much bigger role in the power demand compared to air drag.

When we compare an example acceleration of 1 s m 2 which results in 1350 N resistance to the air drag of different speeds, it becomes obvious that the air drag is only relevant on motorways. When driving 80 km/h the resistance energy is 82 N, with 100 km/h 330 N still a lot smaller than acceleration resistance. Only with a speed of 200 km/h the air drag values are in the same range as the values from the acceleration resistance. The author of [3] also found that speed in itself does not cause large environmental problems in urban traffic, which agrees with our findings.

Those findings gave rise to the idea of a new definition of an efficient driving pattern. The definition could be expressed in an informal way by saying “the more constant a driving pattern, the lower is the power demand”. If this idea is pursued, the frequency of the speed change is what gets ultimately evaluated. So we decided to approximate an optimal driving style with the help of a low-pass filter. To come up with the right filter criteria we found guidance values in [11]. Here the authors identified acceleration values below 2 s m 2 and speed below 80km/h as efficient. Those values were used to determine specific rules to evaluate the individual driving events. To validate the results two different tours were analyzed with regard to acceleration resistance. First with the actual speed profile and afterwards with the filtered speed profile. The resistance energy for the “defensive” tour almost didn’t change between actual and filtered signal. While the aggressive tour had a similar acceleration resistance as the “defensive” tour after filtering and a much lower acceleration resistance than the original tour. Those findings suggest that a plausible way to judge the driving behavior regarding energy efficiency was found and can be used.

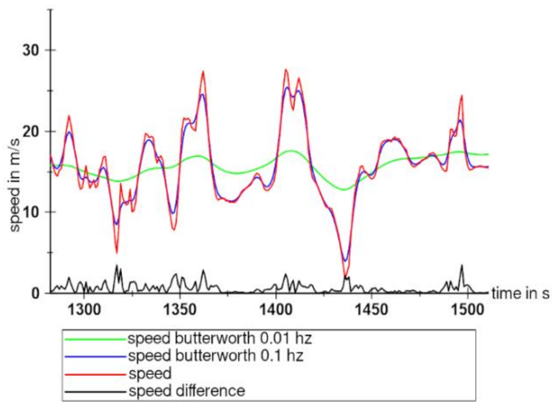

Figure 1. Diagram of a speed profile. Visualizing the criterion to evaluate energy efficiency [7].

In Fig. 1 the speed profile is shown with two different filtered speed signals. With the goal of not punishing a normal driving pattern, 0.1Hz was found as a good fit for the Butterworth filter for our purposes. To evaluate the driving pattern we simply used the difference between the actual speed profile and the optimal speed profile obtained the filtered data. Another counter-intuitive finding was, that an eco-friendly driving doesn’t necessarily take up more time. At least in the way we defined an optimal driving pattern, since the average speed is not changed by the filter. This requires a highly anticipating driving style. To validate that our definition of judging the energy efficiency is a viable option in optimizing the driving style, we compared the optimal driving style with very energy efficient tours in the database. We discovered that they almost didn’t differ, which led us to the conclusion that we found a good way to rate the driving style.

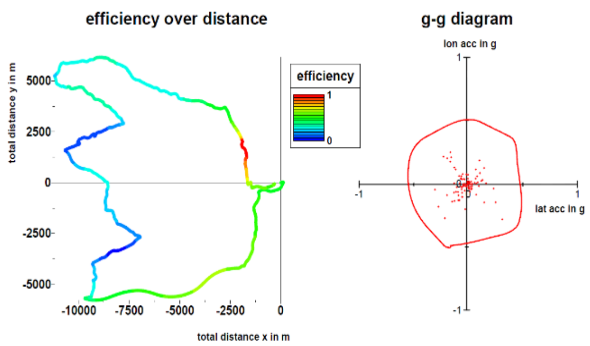

Figure 2. On the left is an evaluated trip with efficiency values between 1 and 0. On the right side are all the accelerations in g visualized, which have the most impact on the evaluation result [7].

In Fig. 2 a graph of an evaluated trip also visualizing the encountered accelerations is shown.

To get those results we had to find a way to interpret those speed deviations in a score between 1 and 0, where 1 is very efficient and 0 worst case driving. In order to prevent people from causing traffic obstructions we decided to define a threshold, before a deviation is punished. If the deviation is below this threshold the driving style scores will not get worse. Additionally a maximum value was defined. The values between the threshold and maximum value got normalized to values between 1 and 0 and inverted, resulting in the desired score. To get an ongoing rating, a sliding window method was employed which gives a rating based on the last 3 minutes, while weighting the most recent values higher than the older ones.

To model also the air drag, we decided to use the threshold of 80km/h to decide how to judge the speed. All speed values below this value were ignored. Because of the quadratic nature of the air resistance we had to adjust the driving evaluation on higher speeds. We modeled a quadratic function between 80 and 200km/h where the lower bound speed gets a score of 1 and the latter gets rated with 0. To get the overall score the two scores get multiplied. This way acceleration has the most important effect on the result, only with very high speeds the overall score gets pulled down and therefore becomes worse.

Those considerations form the basis for the learning algorithm.

© 2020 J. Adv. Inf. Technol.

III. Related Work

Here we want to discuss related work and highlight the differences as well as suggest ideas for follow-up research. There are many researchers working on similar problems like [3], [5], [6], [9], [12]-[15]. They all propose systems which can make sense of vehicle movement. They all had very different goals and just as different were the approaches to achieve those goals. Those goals were:

- To analyze driving styles in order to understand the impact of different driving events on power demand [3]
- Rate driving styles in regards to road safety [14]
- Identify risk level of different driving styles [5] [6]
- Identifying drivers by their driving style [9], [12], [15]
- Categorizing different driving styles [13]

Some of the approaches rely on manual work to find out the necessary features to reach their goals [3], [13]. The other papers rely on some sort of high level feature like speed or acceleration to train a machine learning algorithm. This suggests, that it is hard to extract useful findings from raw GPS data. Thus an interesting application field for deep learning algorithms.

IV. Problem Formulation

This chapter will clarify the purpose of the presented work. The goal is to achieve a driving style classification using raw GPS signals. According to [12] previous proposed solutions relied on preprocessing and feature extraction. All the solutions we have looked at rely on some preprocessed features like acceleration, jerk, speed etc. Because they discovered that the chosen learning algorithms could not handle the raw latitude and longitude values directly. But it is an interesting field to apply deep learning, which in other fields proved to be able to learn different abstraction levels of features from input data [16].

This is a supervised learning scenario in which the algorithm learns to use GPS data in order to rate a driving style with respect to energy efficiency. Before training this model it is necessary to get labeled data. The most obvious way would be to let drivers judge, but in the case of driving style the subjective feeling about the effects may differ significantly from person to person. Considering this, it is a bad basis for a learning algorithm. So we used the evaluation algorithm developed in our research group at the University of Kempten as described above. This procedure is used to label all the available training data. Hereby the learning algorithm cannot achieve more energy efficient results than the developed offline concept, but we can show how well the deep learning model can work with raw and flawed GPS input data, to model the vehicles' trajectories.

Such a neural network, which can model the vehicle movements can be useful for a great variety of scenarios. For example car manufacturers who need to optimize energy consumption, design driver assistance systems or develop car-to-car communication systems which could model an optimal traffic flow based on individual vehicle data.

V. Dataset

Within this chapter the problems with the raw input data are described as well as how the data gets processed. Finally it is shown how the results were validated.

A. Cleaning GPS Data

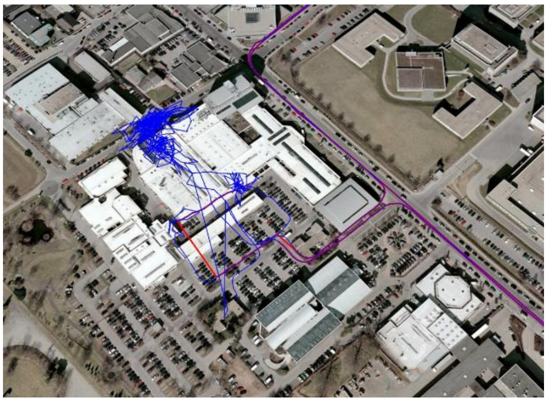

Figure 3. Blue: flawed GPS signal with a typical point cloud. Red: cleaned GPS signal.

One of the most significant things to consider in a machine learning environment are the datasets used for training and evaluating the architectures. In the following part the GPS data will be described in detail, discussing the main challenges arising from working with data from GPS sensors. There are three main errors discovered while working with the GPS data. These are point clouds, zero jumps and time gaps. Point clouds mostly occur at very low speeds or if the vehicle stands still in one place for some time. The result can be seen in Fig. 3. We can see the blue line, coming to and going from the parking lot, where the car went into an underground carpark. This causes the GPS signal to generate extensive point clouds. Those GPS jitters do not represent any car movement and should therefore be removed from the data. The second error source are zero jumps, where the GPS sensor returns a value of (0, 0) or (0, -90) for latitude and longitude, which means that an internal sensor error was encountered. The time gaps can be the result of a sensor error or an error in the logging component which records all vehicle data.

The last two error cases are straightforward to fix. The zero jumps just get removed. For the time gaps we decided to linearly interpolate between the existing points. The point clouds are a lot harder to identify correctly and clean up. Two features which can help to identify those faulty GPS values are very low speed values (mostly < 3 km/h) in combination with extreme changes in angle. For generating the labeled data we found, that we could develop a quite reliable algorithm to remove the point clouds by only considering the speed profile. The rules were simple: remove points, if a higher speed occurs, check whether it belongs to a plausible acceleration then include all points from this acceleration, otherwise keep removing those measured points. Finally all GPS points were filtered with a Butterworth filter, because the smallest of deviations in latitude and longitude can lead to high lateral and longitudinal acceleration. The used filter was a low pass filter with filter degree of 1 and cutoff frequency of 0.1 Hz. To eliminate signal delay in the data, the filter was applied bidirectional.

Those manual corrections are only applied to generate the labels for the driving style and are not the source for the learning algorithm, which is supposed to identify those errors and handle the data accordingly.

B. *Manual Feature Extraction*

This section describes the extraction of the necessary features for the evaluation algorithm. (Fig. 4)

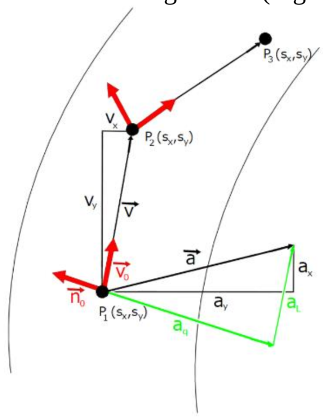

Figure 4. Calculation of lateral and longitudinal acceleration of the vehicle, from the GPS data [7].

With the GPS sensor there are different values which can be obtained. The following values are relevant for the driving style evaluation: timestamp of the value, latitude, longitude, altitude, speed and signal accuracy. Those values are in raw form not enough for the evaluation algorithm. But the speed profile, lateral and longitudinal acceleration as well as deceleration are needed. Those can be obtained by the means of the following equations developed in [7]:

GPS input:

$$\phi_1 = latitude\ at\ the\ start\ time \quad (1)$$

$$\lambda_1 = longitude\ at\ the\ start\ time \quad (2)$$

$$t_1 = start\ time \quad (3)$$

Distance between degree of latitude and degree of longitude:

$$s_{\lambda_1} = 111300\ m \cdot cos(\phi_t) \quad (4)$$

$$s_{\phi_t} = 111300\ m \quad (5)$$

Calculation of the distance traveled in x-direction:

$$s_{x_t} = (\lambda_{t+1} - \lambda_t) \cdot s_{\lambda_t} \quad (6)$$

Calculation of the distance traveled in y-direction:

$$s_{y_t} = (\phi_{t+1} - \phi_t) \cdot s_{\phi_t} \quad (7)$$

Calculation of the distance traveled in driving direction:

$$s_t = \sqrt{s_{x_t}^2 + s_{y_t}^2} \quad (8)$$

Calculation of the speed traveled in x- and y-direction:

$$v_{x_t} = \frac{s_{x_{t+1}} - s_{x_t}}{\Delta t} \quad (9)$$

$$v_{y_t} = \frac{s_{y_{t+1}} - s_{y_t}}{\Delta t} \quad (10)$$

Calculation of the speed traveled in driving direction:

$$v_t = \sqrt{v_{x_t}^2 + v_{y_t}^2} \quad (11)$$

Calculation of the unit vectors for speed:

$$v_{0_x} = -v_{0_y} \quad (12)$$

$$v_{0_y} = +v_{0_x} \quad (13)$$

Calculation of the normal vectors for speed:

$$n_{0_x} = \frac{v_{x_{t+1}}}{v_t} \quad (14)$$

$$n_{0_y} = \frac{v_{y_{t+1}}}{v_t} \quad (15)$$

Calculation of the acceleration share in x- and y-direction:

$$a_{x_t} = \frac{v_{x_{t+1}} - v_{x_t}}{\Delta t} \quad (16)$$

$$a_{y_t} = \frac{v_{y_{t+1}} - v_{y_t}}{\Delta t} \quad (17)$$

Calculation of longitudinal and lateral acceleration:

$$a_{l_t} = a_{x_t} \cdot n_{0_x} + a_{y_t} \cdot n_{0_y} \quad (18)$$

All the described preprocessing and data enriching where implemented according to the formulas in a Python program and validated.

C. *Validation*

In this section it will be described how the results of the extracted features were validated. First, all the encountered values were validated against the values occurring in normal driving situations which are acceleration values up to 2.45 s/m^2 [17]. In order to make a reliable statement about the quality of the calculated values it was necessary to compare them to values of an actual acceleration sensor. In Fig. 5 and 6, the calculated values get compared to the sensor values.

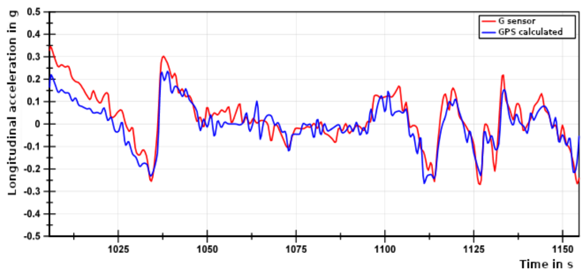

Figure 5. Red: longitudinal acceleration from the sensor. Blue: calculated values [7].

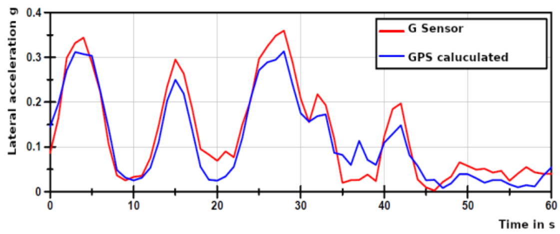

Figure 6. Red: lateral acceleration from the sensor. Blue: calculated values [7].

© 2020 J. Adv. Inf. Technol.

When examining the figures, it is noticeable that the shape is roughly the same. Only single extreme amplitudes do not get modeled by the calculated values. For the purpose of the driving style analysis those peaks should not be relevant. Since those peaks most likely stem from uneven or bumpy road surface or little jerks at the steering wheel which cannot be captured through GPS signals.

VI. The Model

This chapter describes how the used algorithm was built, which improvements were made and how they affect the result. Since Deep Learning has a great number of hyper-parameters, it is a hard task to find the optimal configuration. The search has to be done heuristically, because a brute force approach would take a long time. First, a few scenarios were built up to test the system with different hyper-parameters. After a reasonable configuration was found some further optimizations were included.

A. Intuition

To get to grips with the first idea of how to design the architecture, it is worthwhile to look into the necessary performance requirements of the underlying network. In the area of computer vision there has been done some interesting work to better understand the learning process in a neural network. The authors of [18] proposed an approach where they visualized the weights of the hidden layers to get an insight for what kind of input they are activated. This was the motivation to try and work out an architecture which can represent the features, which we were trying to find with the manual approach, for a Recurrent Neural Network architecture.

If we think about the different requirements of the network, the first one is to iron out the GPS errors. So it seems natural that one layer will learn to identify the GPS errors. The next layer of abstraction is the feature extraction, where the algorithm has to learn to model the accelerations and speed from the GPS input. Based on this there is one more level of abstraction which learns the driving style evaluation based on the output of the previous layer. The thing which is quite hard to estimate is the number of cells in each layer, because it is not obvious how many different features have to be learned from the input data. Some guesses are: fast and slow acceleration/deceleration, speed below the defined threshold and above. To better understand the inner workings of the RNN we tried to analyze the learned weights. Nevertheless it is not easy to find those features for this scenario, analogous to the feature identification in a Convolutional Neural Network. The field of visually understanding the trained model of RNNs has a lot of potential.

Having this in mind we tried to use 2 to 3 layers of RNN-cells and a fully connected output layer. For the number of cells per layer some experiments are required.

B. Hyperparameters

These are the parameters that control the overall behavior of the neural network, but cannot be learned by itself and have to be chosen by the developer. Choosing the right hyper-parameters is no a straightforward task which can be done with some fixed set of rules. LeCun identified a set of heuristics to choose optimizations [19]. Nonetheless there are no scientific results that can guide oneself through the design process of the complete algorithm [20]. Most developers trust their intuition to find the right parameters. Some of the more important factors are:

- Way of initializing the weights
- Kind of architecture
- Number of layers
- Number off cells in each layer
- Regularization parameters
- Learning rate
- Way of backpropagating the error
- Dropout rate
- Weight decay
- Early stopping

The obvious approach would be a grid search, which in the naive approach is a brute force way in which every single combination of hyper-parameters gets tested. The network is trained for every combination, but this is not a viable path for most tasks, since the training time for one model is in most cases really high. To reduce the number of combinations it is good to know the problem domain as deeply as possible, so one can reduce the parameters to test, or at least restrict the range of values.

Weight initialization has a major impact on the training success and also training speed. The researchers in [21]-[24] worked on good ways to do the initial weighting. The basic idea is to give the neural net the best possible starting point to improve fast. The weights should be chosen randomly, but stay in a range where the activation function for the neurons is almost linear. If the weights are too big or too small, the gradient will be small and therefore the training progress will get slow. If the weights are chosen appropriately the network can start with fast learning progress and only later on has to learn the more complex non-linear parts [19].

The above discussed optimal range of values does not only apply to the activation function, but also to the input data. It would be a disadvantage if all the input data would be very big, since it would prevent the gradient based learning. It is therefore a good idea to transform the values in such a way that they are small and in average equal to 0. It can be advantageous to apply this heuristic to every layer of the network [25]. This process is called Batch Normalization. In [26] the problem of Internal Covariate Shift is described. This behavior occurs in deep architectures and can be prevented by means of layer wise batch normalization. For LSTM networks a normalization to values between 0 and 1 is most widely used [27].

The choice of the activation function also has a major impact. This becomes clear when considering the fact that every neural network which uses only linear activation functions can be reduced to a single perceptron [28]. In the LSTM cells there are multiple activation functions required which ideally only return 0 or 1 which can be modeled by sigmoidal functions. To prevent the vanishing gradient problem a function like tanH is a good fit for the LSTM Networks [29].

The works of [30]-[33] did a lot of research for how to adapt the learning rate throughout the training. But it was not until [34] found an easy and efficient way to find a good learning rate. This approach starts with a low learning rate, and after each step it doubles the learning rate. Then it measures the loss and takes the value were the loss function improved the most.

For all Recurrent Neural Networks one needs a special implementation of the backpropagation algorithm, since it has to consider the recurring part of each cell. For this purpose [35] proposed a variant which unfolds the time steps to a "normal" neural network, and adjust all weights for a unfolded recurrent cell at once.

Then there are other parameters which can help the algorithm to prevent from overfitting like dropout, where a chosen rate of the neurons won't get used during a given training step. This way the network needs to be more flexible and learn a more robust representation of its function [36]. The authors of [33] showed that a simple weight decay can improve the generalization capability of a neural network.

Another useful tool is early stopping, which simple stops the learning if the loss function doesn't improve anymore.

C. Implementation

First it was necessary to choose a suitable network architecture. We tried some shallow learning algorithms like, random forest and SVM but those were not able to get any meaningful information extracted from the input data. Afterwards a standard MLP was used to handle the problem, but both approaches were not promising without further data preprocessing. This is what we expected since it is not possible to achieve meaningful results based on single acceleration- or speed values. Only if the history based on the last x time steps are considered, a procedure is able to judge the driving style. The only thing those algorithms were able to learn was the speed dependent part of the driving style score, which is obvious because it is not dependent on previous values. Those tests showed that some sort of Recurrent Neural Network was necessary. Since the LSTM networks are widely used in the domain of timeseries prediction this network architecture was chosen.

The whole software project was implemented in the programming language Python and the Deep Learning framework Keras. This setup made it easy to quickly test any given combination of algorithmic improvements and hyper-parameters.

VII. Experiments & Evaluation

To get an impression what kind of results the network can give, three different scenarios were build up. In each scenario the labels were the preprocessed driving styles, only the input data was varied:

- All GPS- and feature channels
- Acceleration values and speed
- Raw GPS data

The first scenario was designed to see whether the algorithm was able to get useful results based of this big amount of data. But the result was a high bias network, which always returned a near perfect driving style with values close to 1. This is most likely due to the high complexity with the big amount of input data. In this scenario the number of layers and cells was incrementally enlarged until it became unfeasible to run the experiments because of the time and hardware constraints. The validation loss could not be improved. In fact the contrary happened. It became worse. If one would want to pursue this path, bigger networks and really big datasets would be needed.

To improve the results in the second scenario the number of inputs was reduced to the channels which were known to have an effect on the results, namely acceleration/deceleration and speed values. The result is shown in Fig. 7. This result shows, that the algorithm is able to model the driving style evaluation. It is not exactly the same, but the similarity is good and reliable enough to use it for training of drivers. In this case the number of cells and layers was also raised to the point where no significant improvement in validation loss was apparent. In this case there were 3 layers with 10 LSTM cells each.

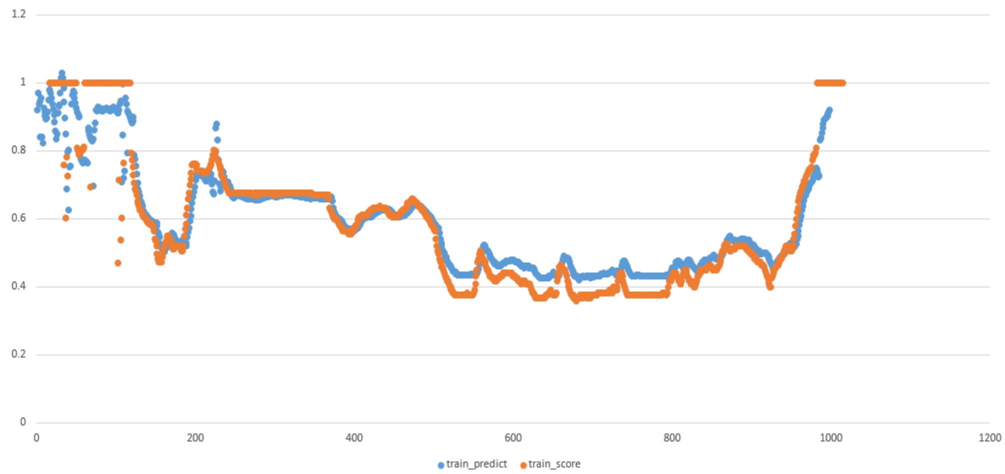

Figure 7. Orange: label for the training data. Blue: the score prediction from the neural net.

Finally the third scenario was applied which should model the same output as the second scenario, only with the raw GPS input. The naive approach was to take the same network architecture as in the previous scenario and just give the latitude and longitude values as input data. All the hyper-parameters and optimizations remained unchanged. The algorithm was able to learn the function without any changed precision compared to the approach with the preprocessed features. This shows the power of Deep Learning methods and in this case of the LSTM cells. Because the developer doesn't need to know the individual properties that are required for the evaluation, but the network learns the different layers of abstraction on itself. It only took about 5 to 7 additional epochs of training to get to the same results. Based on these findings the rest of the hyper-parameters were tried out to improve performance and accuracy. The algorithm showed to perform better with a dropout rate of 0.5 and with reducing the learning rate on plateaus. Additionally early stopping was used to ensure that no unnecessary training steps were performed.

VIII. Conclusion

We presented a system which leverages the possibilities of deep Recurrent Neural Networks, specifically LSTM networks, to make assertions about a driving style based on raw GPS signals. This means it is a system which can handle error-prone and incomplete GPS signals. It can generate a meaningful representation of the vehicle trajectory, in the hidden layers, and make assertions about the energy efficiency of the observed driving patterns. This is a very useful tool for many different applications. A possible application can be a systematic approach to find driving style categories for insurance companies, which want to have different insurance rates depending on the driving style. Another application can be training instructions for people to drive in a safe and economic way. Other applications could be found, where faulty sensor signals, which follow an identifiable pattern need to be handled. It could also help in a great variety of driver assistance systems or autonomous driving scenarios where a model of the vehicle trajectory is needed.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Samuel Würtz collected the data and implemented the described program. He also did the research. All authors wrote the paper and gave their final approval.

References

[1] P. Poizot and F. Dolhem, "Clean energy new deal for a sustainable world: From non-co2 generating energy sources to greener electrochemical storage devices," *Energy Environ Sci.*, vol. 4, pp. 2003-2019, 2011.

[2] R. Scarinci, F. Rast, and M. Bierlaire, "Needed reduction in mobility energy consumption to meet the goal of a 2000-watt society," *Transportation Research Part A: Policy and Practice*, vol. 101, pp. 133-148, 2017.

[3] E. Ericsson, "Independent driving pattern factors and their influence on fuel-use and exhaust emission factors," *Transportation Research Part D: Transport and Environment*, vol. 6, no. 5, pp. 325-345, 2001.

[4] R. Vaiana, *et al.*, "Driving behavior and traffic safety: An acceleration-based safety evaluation procedure for smartphones," *Modern Applied Science*, vol. 8, pp. 88-96, 2014.

[5] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and M. C. González, "Safe driving using mobile phones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1462-1468, 2012.

[6] J. S. Wijnands, J. Thompson, G. D. Aschwanden, and M. Stevenson, "Identifying behavioural change among drivers using long short-term memory recurrent neural networks," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 53, pp. 34-49, 2018.

[7] S. Keidler, "Konzeption und entwicklung einer fahrerklassifizierung zur einstufung von einzelfahrten nach ausgewählten kriterien," master's thesis, 2016.

[8] C. M. Martinez, M. Heucke, F. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666-676, 2018.

[9] J. Guo, Y. Liu, L. Zhang, and Y. Wang, "Driving behaviour style study with a hybrid deep learning framework based on gps data," *Sustainability*, vol. 10, no. 7, p. 2351, 2018.

[10] M. Geiler and H. Kerwien, "Evaluation of fuel-saving driving courses on driver behaviors," *Zeitschrift für Verkehrssicherheit*, vol. 54, no. 3, 2008.

[11] R. Krause, "Anfahrbeschleunigungen im alltäglichen straßenverkehr," *Verkehrsunfall und Fahrzeugtechnik*, 2002.

[12] W. Dong, J. Li, R. Yao, C. Li, T. Yuan, and L. Wang, "Characterizing driving styles with deep learning," arXiv:1607.03611v2, 2016.

[13] J. E. M. Anastasio, *et al.*, "Characterizing the driving style behavior using artificial intelligence techniques," 2013.

[14] Z. F. Quek and E. Ng, "Driver identification by driving style," Technical report, technical report in CS 229 Project, Stanford University, 2013.

[15] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85-117, 2015.

[16] M. Nickel, *et al.*, "Längs- und querbeschleunigungen im alltagsverkehr," 12 EVU-Jahrestagung, Zürich, 2003.

[17] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional net-works: Visualising image classification models and saliency maps," preprint, 2013.

[18] Y. LeCun, L. Bottou, P. Haffner, and P. Howard, "Djvu: A compression method for distributing scanned documents in color over the internet," in *Proc. Color and Imaging Conference*, 1998.

[19] F. Shaikh. (2016). Optimizing neural networks using keras. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/10/tutorial-optimizing-neural-networks-using-keras-with-image-recognition-case-study/

[20] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feed-forward neural networks," in *Proc. of the International Conference on Artificial Intelligence and Statistics*, 2010.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. of the IEEE International Conference on Computer Vision*, 2015, pp. 1026-1034.

[22] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," technical report arXiv:1207.0580, 2012.

[23] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. the 25th International Conference on Machine Learning*, 2008, pp. 1096-1103.

[24] Y. LeCun, L. Bottou, G. Orr, and K. Muller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, G. Orr and K. Muller, Eds., Springer, 1998.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," ArXiv e-prints 1502.03167, 2015.
[26] J. Brownlee. (2016). Time series prediction with lstm recurrent neural networks in python with keras. [Online]. Available: http://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
[27] J. Brauer, "Deep learning lecture slides," 2017.
[28] S. Hochreiter and J. Schmidhuber, "Lococode versus PCA and ICA," in *Proc. the International Con-0ference on Artificial Neural Networks*, Sk övde, Sweden, 1998, pp. 669-674.
[29] C. Darken and J. E. Moody, "Note on learning rate schedules for stochastic optimization," *Advances in Neural Information Processing Systems*, pp. 832-838, 1991.
[30] N. Murata, K. R. M üller, A. Ziehe, and S. Amari, "Adaptive on-line learning in changing environments," *Advances in Neural Information Processing Systems*, pp. 599-605, 1997.
[31] R. S. Sutton, "Adapting bias by gradient descent: An incremental version of delta-bar-delta," in *Proc. of the Tenth National Conference on Artificial Intelligence*, 1992, pp. 171-176.
[32] L. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conference on Applications of Computer Vision*, 2017, pp. 464-472.
[33] H. Sompolinsky, N. Barkai, and H. Seung, "On-line learning of dichotomies: Algorithms and learning curves," *Neural Networks: The Statistical Mechanics Perspective*, pp. 105-130, 1995.
[34] P. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, pp. 1550-1560, 1990.
[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.
[36] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," *Adv. Neural Inform Process Systems*, vol. 4, 1992.

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.

Samuel Würtz is a researcher at the German University of Applied Sciences (UAS) Kempten in the research department "connected mobility". He received his Bachelor degree in Computer Science and Master degree in Applied Computer Science from the UAS Kempten. His field of research moved from Data Science in the context of electro mobility to connected mobility in general. Focusing on applications and adaptations of machine learning and specifically deep learning algorithms to mobility and transportation data, in the recent years.

Ulrich Göhner is professor at the University of Applied Sciences Kempten. He received his Diploma degree at University of Stuttgart and got a doctoral degree in the field of Computational Fluid Dynamics. His research interests include algorithms for high performance computing, parallel computing and machine learning. He is working on multiple research projects in the field of "connected mobility" and "optimization of production systems". He is cofounder of the company DYNAmore GmbH and is involved in a number of scientific projects especially in the field of car crash simulation, occupant safety and optimization of production processes.