Bike trajectory prediction with onboard sensors

ĄŽUOLAS ARLAUSKAS, University of Twente, The Netherlands

As the number of cyclist casualties rises, it is essential to make the bike a safer option of transport. This can be done by making the bike a "smarter" device. However, bicycles are severely behind in the smart vehicle industry. Whilst the automobile industry is already creating autonomous cars, research on smart bikes is few and far between. An intelligent bicycle solution would allow for a safer environment for every smart vehicle in the system. One of the key features of an intelligent bike is trajectory prediction as it allows for safer navigation, lane keeping, obstacle detection and more. The current state of the art solutions are not made for bicycles as they are limited in space and processing power. The study compares different approaches to path prediction with the goal of finding an effective LSTM architecture for the purpose of a bike's trajectory prediction using on-board sensors. The described model lays the ground work for further research in the field of smart bicycles by offering an effective LSTM model and plenty of ways to improve upon it with the goal of bridging the gap between the high-tech autonomous cars and comparably low-tech everyday bikes. The paper deals with limitations of time, variety in the dataset, lack of attention to external factors, edge computing and limited testing of prediction history and horizon.

Additional Key Words and Phrases: bike, bicycle, trajectory prediction, path prediction, RNN, recurrent neural network, LSTM, long short-term memory, time sequence prediction

1 INTRODUCTION

According to the government of the Netherlands, the 17 million people residing in the country share about 22.8 million bicycles [5]. The relatively inexpensive and small vehicle is a staple in Holland. Even though back in 2022 the dutch government set aside a budget of 780 million euro and just last year they allocated another 18 million euro to be used over the next 3 years to improve bicycle infrastructures [2], a statistic from Statista claims that in just 2022 there were 291 fatal cyclist road incidents [6]. That is the most cyclist casualties since 1996 [6]. There are many possible reasons for this including weather, visibility conditions. Since infrastructural upgrades do not seem to affect the number of accidents positively, one could take a different approach to road safety. For example, insuring more safety for a cyclist by enhancing the bicycle's intelligence

However, compared to other forms of transportation, the bicycle is quite uninventive today. Apart from electric bicycles, not much innovation has become widespread in the field. As the world moves on to new smart technologies, the bike must keep up. This research is an effort to bridge the gap between the low-tech bicycle and the high-tech autonomous vehicles. Although autonomous vehicles have been around since the 1980s [7], certain means of transport are much more researched than others. To be more specific, bicycle research is few and far between. As we aim to make cycling safer and more convenient for future generations, we must implement intelligent assistance to the everyday bicycle. Although many sophisticated systems exist for other autonomous vehicles, they are not as feasible for bicycles as the space and resources (computing power, memory, etc.) they have are much more limited.

Currently, the main safety mechanism on most bikes is the very simple bell. However, it is only useful in simpler situations as it can only be properly used when the driver can foresee an obstacle and wants to communicate the danger to it. This does not entail proactive safety though, i.e. forecasting possible future events. That could be done using data driven approaches. More specifically, one of the essential features of an autonomous vehicle is trajectory prediction. This enables the possibility of safer navigation, lane keeping, obstacle detection and more. However, the problem of implementing such a system on a bicycle in a lightweight manner is far beyond a simple one. As infrastructure gets smarter and big cities connect their traffic systems with other devices like smart cars to maintain road safety – adding bikes to this system would make it even more detailed and safer.

The main gaps in research come in four forms. Firstly, there is a lack of research on bikes specifically. Most route prediction research is made for smart cars [1, 3, 9, 10]. This leaves a wide gap in research as a bike is fundamentally different from a car. A bike can be far more reactive as it relies on human input more and is generally less stable. Also bikes are often ridden on different types of paths whilst cars mostly stick to paved roads. Moreover, most research does not account for complex situations, often choosing to predict a general direction (east, west, south, north) [9, 10] whilst pointing out the need for a more adaptive model. Furthermore, most papers on the topic do not mention what they run their models on [1, 10]. This is another big gap in research as real-time trajectory prediction on a bike is heavily constrained by the aforementioned constraints of space and resources. Finally, there's a gap in multimodal sensor implementations for trajectory prediction.

This study will attempt to construct an algorithm that will use differently placed IMU sensors and a phone to measure GPS for the task of trajectory prediction. An inertial measurement unit (IMU) is a sensor that produces accelerometer and gyroscope measurements whilst GPS is global positioning data collected on a phone. The research aims to study current architectures used for route prediction to answer the following research questions:

RQ1: What algorithm is best suited for a bike's trajectory prediction using GPS and IMU inputs?

RQ2: What effect would adding an additional IMU have to the quality of a bike's trajectory prediction?

2 RELATED WORK

2.1 INITIAL MODEL SELECTION

The first point of research was a scientific paper sent in by the study's supervisor. The paper was a Survey On Trajectory-Prediction Methods for Autonomous Driving [3]. This paper covers many different

TScIT 42, January 31, 2025, Enschede, The Netherlands

methods which allowed to weigh the pros and cons of each to decide on which system is best fit for a bike.

2.2 PHYSICS BASED MODELS

The main issues of physics based models come down to two things. Firstly, they are very limited in complex situations, this is especially the case since the bike is quite maneuverable and humans can act unpredictably at times. Another key issue here is the fact that every time the system needs to make a prediction it has to run its calculations anew. This means that it is not very feasible to use a system like this in a real-time setting on a bike.

2.3 MACHINE LEARNING MODELS

Machine learning models are much better than physics based models in predicting complex situations however worse than deep learning models. Moreover, feature engineering is a crucial part of the success of ML models. If the person running a machine learning model does not have much experience extracting the correct features for the model - the quality of the model will be low.

2.4 DEEP LEARNING MODELS

This is the category this study will explore. More specifically, this study works with RNNs (recurrent neural networks) or LSTMs (long short-term memory). A recurrent neural network (RNN) is a deep neural network trained on sequential data which can make future predictions depending on the given sequential input [4]. This is due to a few reasons, namely these models having a good balance between complexity and performance and being good at modelling dependencies over time. These traits are perfect for a real-time trajectory prediction on edge devices.

2.5 FURTHER RESEARCH

After choosing to focus on LSTMs, a type of RNN, in order to find more related literature, works that the study supervisor sent in and papers on Google Scholar were studied. Since bikes are not widely covered in this field, the main search terms were ‘trajectory’, ‘prediction’, ‘path’, ‘detection’, ‘time sequence prediction’, ‘RNN’, ‘LSTM’, ‘recurrent neural network’, and ‘long short-term memory’.

Research has already been conducted about using RNNs in trajectory prediction [1, 3, 8–10]. These papers allowed me to outline the 5 gaps mentioned in the introduction whilst also offering some models to try out.

3 METHODOLOGIES

3.1 DATASET

The dataset used for this research was previously compiled by the University of Twente. It is made up of 53 people’s data of riding a bike with 5 sensors. Firstly there’s GPS collected from a phone. Moreover, there are 4 separate IMUs – on the handlebar, on the pedal, on the helmet and on the frame. Due to time constraints and there being a lot of data, only one person’s data was used. That is around 160000 data points. The columns of the dataset include: a timestamp, the angular acceleration and gyro data of all the IMUs, the latitude, longitude, the velocity and the angular acceleration from the GPS (phone).

3.2 GROUND TRUTH

For this project, the GPS data will work as the ground truth. The GPS, whilst having 5 meter deviation, is still good for recording the general path and low-speed movements.

3.3 TRAINING, VALIDATION AND TESTING

The dataset was split into 3 parts – 70% training, 15% validation, 15% testing. For each 100 data points the next 10 data points are predicted. The models were implemented in TensorFlow Keras and were trained for 30 epochs each using mean squared error (MSE) as the loss function:

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

where:

- n is the number of samples
- $\hat{y}$ is the predicted value
- y is the true value

and the Adam optimizer.

The input data used was the angular acceleration and gyro data of the frame IMU with the hypothesis that it is the most stable, relative coordinates calculated from the latitude and longitude, the velocity and the angular acceleration from the GPS (phone). The targets were relative coordinates.

3.4 MODEL ARCHITECTURE

3.4.1 “RoNIN”. Before constructing a model from other papers, the plan was to use the “Robust Neural Inertial Navigation” model [8]. This model offers multiple different neural inertial navigation architectures which use IMU and the now discontinued Google application Tango data. Due to the similarity in the used data, the plan was to take away the extra features that came from Tango and train the model with the aforementioned dataset. However, this plan did not work out as very quickly after removing some of the inputs the performance became terrible even with their own dataset which meant that getting it to work well on the restricted type of data available would have been unfeasible.

3.4.2 MODEL 1. After spending a lot of time on the failure of using RoNIN, it was clear that a more stripped down model was necessary due to the lack of time and lack of features in the used dataset. Because of this, other papers were investigated for a simpler model. It was important that the paper either detailed the architecture of the model clearly or offered a code repository to investigate for the model, to be able to examine and compare the models efficiently. The first two papers [9, 10] offered a simple structure that can be seen in figure 1.

It is also important to note that this particular model was originally used for a classification task.

To allow for the output to be 10 predictions in the future, a small adjustment to the architecture was necessary. The dense output layer would collect the 10 predictions and the reshape layer put them into the correct shape to compare against the validation set. The final architecture for model 1 can be seen in figure 2.

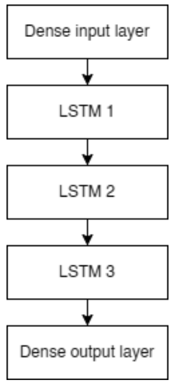

Fig. 1. The architecture of the first model

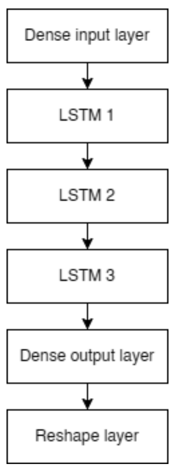

Fig. 2. The adapted architecture of the first model

3.4.3 MODEL 2. The next model [1] offered a different structure which can be seen in figure 3. The big differences between the two models are the number of LSTM layers and the second model using time distributed dense layers. A time distributed wrapper applies the selected layer (in this case a dense layer) to each timestamp independently. Without this wrapper, the time sequence input gets flattened and loses the temporal structure, failing to extract the important features of each step. Instead, a non-time distributed layer would try to extract the features from an amalgamation of the different time steps.

Once again, the model needed to be slightly adapted to only keep the 10 last predictions per 100 datapoints, thus the revised architecture can be found in figure 4.

4 RESULTS

4.1 RESEARCH QUESTION 1

4.1.1 MODEL 1. The training lead to a root mean square error of approximately 166.32923 meters. This was calculated by first getting the distance between two coordinates using the Haversine formula:

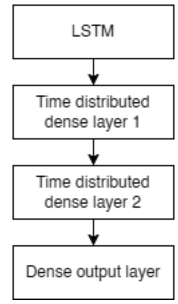

Fig. 3. The architecture of the second model

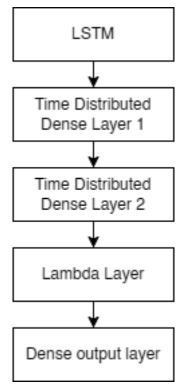

Fig. 4. The adapted architecture of the second model

$$d = 2R \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

where:

- d: Distance between the two points (in meters)
- R: Radius of the Earth (in meters, typically $6,371,000$ meters)
- ϕ_1, ϕ_2: Latitudes of points 1 and 2 (in radians)
- λ_1, λ_2: Longitudes of points 1 and 2 (in radians)

and then turned into RMSE using the following formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2}$$

Here:

- n: Total number of data points
- d_i: The Haversine distance (in meters) for the i-th pair of actual and predicted coordinates

The real vs. predicted trajectory from the testing can be found in figure 5 and the graph of the loss function from the training can be found in figure 6. The graph shows the entire trajectory for context but only the true trajectory (in green) is predicted (prediction is in red).

4.1.2 MODEL 2. This model performed way better than the previous. The root mean squared error was approximately 22.46106

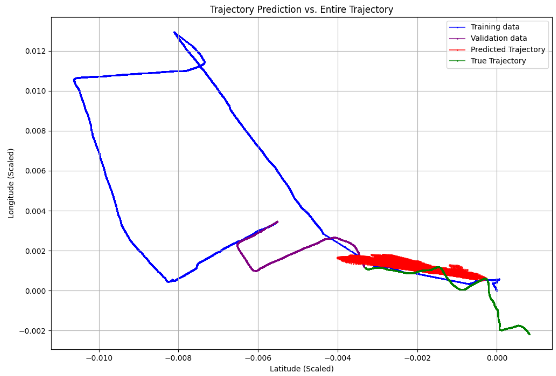

Fig. 5. Model 1 testing performance

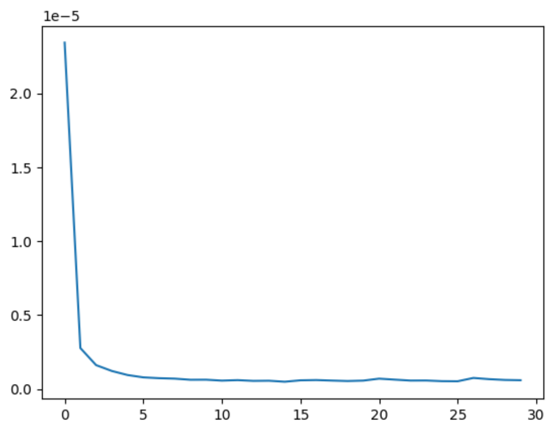

Fig. 6. Model 1 training loss graph

meters. The real vs. predicted trajectory from the testing can be found in figure 7 and the training loss graph can be found in figure 8.

4.2 RESEARCH QUESTION 2

After the success of the second model, a second IMU was added. The handlebar IMU was chosen to be added specifically due to the hypothesis that the handlebar movements could contain meaningful information for the prediction of trajectory due to steering being directly connected to the path that the bike will take. This resulted in a root mean squared error of 19.69739 meters. The predicted vs. real trajectory can be found in figure 9 and the loss graph is available in figure 10.

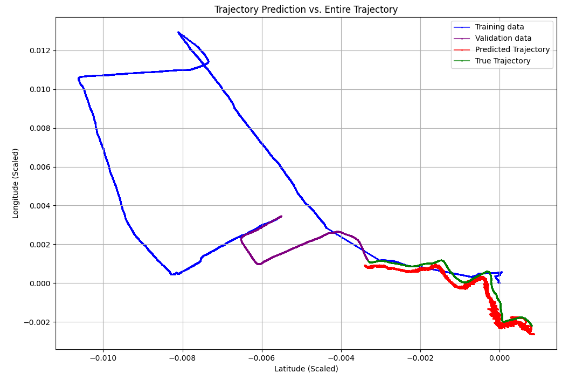

Fig. 7. Model 2 testing performance

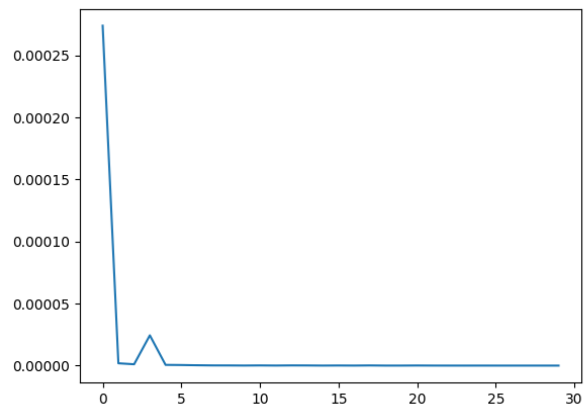

Fig. 8. Model 2 training loss graph

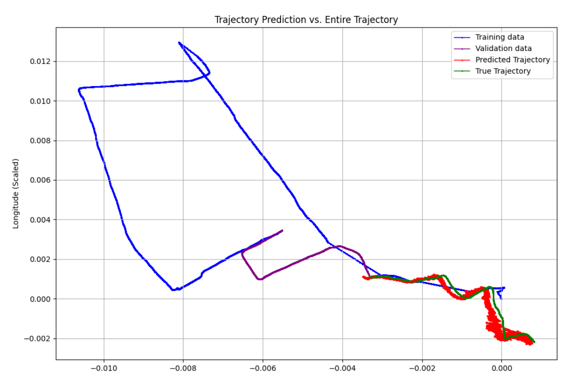

Fig. 9. Model 2 with an extra IMU testing performance

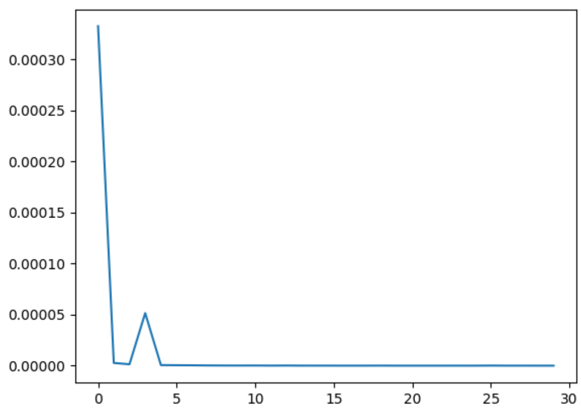

Fig. 10. Model 2 with an extra IMU loss graph

5 DISCUSSION

5.1 RESEARCH QUESTION 1

After comparing the two models from the 3 different papers, the second one outperformed the first one. This could be due to a number of factors.

5.1.1 LSTM UNIT SIZE. Even though the first model had more LSTM layers – they were all 128 units each (compared to the second model's 256 units in one layer). Even though more layers should result in more complex pattern recognition and better generalization – the results tell a different story.

5.1.2 TIME DISTRIBUTED DENSE LAYERS. A big part of the second model's success can be attributed to the two time distributed dense layers. This type of dense layer is especially useful in RNNs as it applies the dense layer to each step of the time sequence input independently. This means that the model optimizes the prediction with the temporal dependencies in mind. It is likely that this is the key to this model performing better than the first one. This allows the model to better learn from the entire sequence of the input.

5.1.3 APPLICATION OF THE MODEL . As mentioned before, the first model was originally a classification model and thus might be more optimal for that purpose. Since the task this paper talks about is a regression task, the second model is much more optimal.

5.2 RESEARCH QUESTION 2

The additional handlebar IMU information ended up performing a little better than the same model without the additional IMU. There could be a few potential reasons for this.

5.2.1 PLACEMENT. As described previously, the handlebar IMU was chosen with the hypothesis that since its movements directly correspond with the way the bike goes, the performance should be better. However, this cannot be confirmed nor denied until tested with an extra IMU in a different placement.

5.2.2 BETTER UNDERSTANDING OF COMPLEX SITUATIONS. As mentioned in the introduction, one of the gaps in research is the complexity of the prediction. The model with an extra IMU may have performed better due to the fact that it was able to better learn more complex patterns.

5.3 LIMITATIONS

5.3.1 TIME. This study was very limited in time thus a lot of corners were cut. This is the main factor that held back this research from being more expansive as the dataset that was made available was of significant size and plenty other models are offered in other papers. This lack of time affected both of the research questions as more different structures could have been explored in the first research question for a more complete understanding of what works well in bike trajectory prediction and more different IMU placements could have been explored in the second research question. Future research should be conducted with more time set aside for more experiments.

5.3.2 DATASET. Even though the dataset is of reasonable size, it could have featured more sensors. This could have lead to the use of the RoNIN model or an even better result for a different model. Furthermore, the dataset was collected in a rather controlled environment as the path was almost the same each time. This may cause the model to not be able to predict more erratic behavior. A further study could feature a more expansive dataset and more multimodal sensor fusion research.

5.3.3 EXTERNAL FACTORS. This study did not take into account many external factors that may affect the quality of the prediction. Different road conditions, weather conditions, urban environments were not considered during this study. A future analysis could take these factors into account.

5.3.4 EDGE COMPUTING. The training and predictions were all done on quite powerful computers. In a real use case, the prediction should be done on the bike itself which would require the model to be ran on something like a single board computer. Future researchers should test out the predicting process in a live situation on an edge device.

5.3.5 PREDICTION HISTORY AND HORIZON. Through out the entire research, 100 data points were used to predict the following 10 steps. The extent to which the model could predict was never tested due to lack of time assigned for the research. This could be further tested to see how far the model can predict with a good RMSE and also with how many data points can the prediction be done whilst having the RMSE remain small.

6 CONCLUSION

This study looked into the possibility of path prediction for bicycles using multimodal sensor data, more specifically using IMU and GPS data. After comparing two different LSTM architectures it was found that for the regression task of predicting a bike's trajectory, time distributed dense layers on top of LSTM layers is the key to a quality model. The second additional handlebar IMU proved to make the model predictions higher quality but more research is needed to find out if that is the case for other IMU placements.

This research leads the way in connecting the gap between the old school bicycle and the modern autonomous vehicles that grow in popularity each day. The study offers a quality trajectory prediction model which can be easily understood and expanded in the covered gaps of this research.

Many limitations stopped this analysis from being more in-depth. Future studies on the subject could make sure there is enough time to cover a wider variety of models, use a dataset with more various sensors, take into account external factors, test the predictions on an edge computing device to make sure it is fit for real-time use and test different prediction history and horizon lengths.

In conclusion, this paper builds on the everyday bicycle by offering trajectory prediction with the use of simple sensors that can be found in a smartphone. The success of this short study means that the future of a smarter and safer bicycle is promising and feasible.

REFERENCES

[1] Florent Altché and Arnaud de La Fortelle. 2018. An LSTM network for highway trajectory prediction. https://arxiv.org/abs/1801.07962

[2] European Commission. 2024. *€18 million funding for cycling highways in the Netherlands.* Retrieved November 16, 2024 from https://urban-mobility-observatory.transport.ec.europa.eu/news-events/news/eu18-million-funding-cycling-highways-netherlands-2024-06-06_en

[3] Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen. 2022. A survey on trajectory-prediction methods for autonomous driving. *IEEE Transactions on Intelligent Vehicles* 7, 3 (Sep 2022), 652–674. https://doi.org/10.1109/tiv.2022.3167103

[4] IBM. [n.d.]. *Recurrent Neural Networks.* Retrieved November 16, 2024 from https://www.ibm.com/topics/recurrent-neural-networks

[5] Government of the Netherlands. 2024. *Bicycles.* Retrieved November 16, 2024 from https://www.government.nl/topics/bicycles

[6] Statista. 2024. *Number of Cyclist Road Fatalities in the Netherlands 1996–2022.* Retrieved November 16, 2024 from https://www.statista.com/statistics/523310/netherlands-number-of-cyclist-road-fatalities/.

[7] Carnegie Mellon University. [n.d.]. *Autonomous Land Vehicle in a Neural Network (ALVINN).* Retrieved November 16, 2024 from https://www.cs.cmu.edu/afs/cs/project/alv/www/index.html

[8] Hang Yan, Sachini Herath, and Yasutaka Furukawa. 2019. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods. https://arxiv.org/abs/1905.12853

[9] Alex Zyner, Stewart Worrall, and Eduardo Nebot. 2018. A recurrent neural network solution for predicting driver intention at unsignalized intersections. *IEEE Robotics and Automation Letters* 3, 3 (Jul 2018), 1759–1764. https://doi.org/10.1109/lra.2018.2805314

[10] Alex Zyner, Stewart Worrall, James Ward, and Eduardo Nebot. 2017. Long short term memory for driver intent prediction. *2017 IEEE Intelligent Vehicles Symposium (IV)* (Jun 2017). https://doi.org/10.1109/ivs.2017.7995919