Weather-Enhanced Multi-Input LSTM for Trajectory Prediction

Po-Chin Chang*
Sofia Ozhogina*
Universiteit Leiden
Leiden, South Holland, The Netherlands

Abstract

This study builds upon the work of Deo, Nachiket, and Mohan M. Trivedi [8], which introduced convolutional social pooling (CSP) to learn interdependencies between neighboring vehicles for trajectory prediction. CSP is a novel approach that captures spatial relationships by pooling LSTM states of surrounding vehicles into a grid-like social tensor, enabling convolutional layers to efficiently model interactions between vehicles, similar to how convolutional neural networks extract features from images. Accurate vehicle trajectory prediction is critical for improving road safety and advancing autonomous driving systems. While their approach performed well, it was limited to vehicle-related data. This study enhances the accuracy of trajectory prediction by incorporating weather data. Features like visibility, precipitation, wind speed, temperature, and humidity are included alongside vehicle-related data. Including weather features strengthens the model, resulting in lower RMSE and NLL values compared to the original study. These findings underscore the importance of integrating environmental factors into vehicle trajectory prediction models for improved performance.

Keywords

Trajectory Prediction, Convolutional Social Pooling, LSTM, Temporal Data, Spatial Data
.

1 Introduction

Self-driving cars are a technology that improves the human lifestyle. Therefore, ensuring safe and efficient navigation for self-driving cars is critical. Convolutional Social Pooling (CSP) combined with Long Short-Term Memory (LSTM) has been proposed as a method to learn spatial and temporal relationships for car trajectory prediction [8]. This framework efficiently extracts features of vehicle interactions and suggests better-predicted trajectories compared to approaches that consider only a single vehicle.

In addition to vehicle interactions, weather conditions are essential for diver behavior. For instance, people usually slow down their cars on a rainy or foggy day to stay safe. Hence, we assume that using weather information in the model can further improve the trajectory prediction performance. A similar idea of incorporating weather data was used in trajectory predictions for aircrafts [12].

Our experiment considers the same trajectory data as the original paper. For the weather data, the weather in California, Los Angeles, and San Francisco corresponds to the vehicle data in freeway US-101 and I-80. Both datasets are aligned by the global time. We built a new model branch for the model structure, including an LSTM encoder and a fully connected layer for introducing the weather data. Our new model is concatenated with the original model, and the output of the whole model is the same, so it is convenient to benchmark. Both root mean square error (RMSE) and negative log-likelihood (NLL) are regarded as metrics.

Overall, the data preprocessing and new model are successfully implemented. The experiment results show that introducing the weather data can indeed advance the model predictions in both metrics. Additionally, the improvement is better with time.

Our contribution is included below:

- Preprocessing the dataset for both trajectory and weather.
- Extend the model structure to introduce weather conditions.
- Achieved improvement in both RMSE and NLL compared to the initial study [8].

2 Related Work

This section explains the core principles of the previous work [8], covering its methodologies, strengths, and limitations. Lefevre et al. [10] conducted a comprehensive survey on vehicle motion prediction models, classifying them into physics-based, maneuver-based, and interaction-aware models. The approach proposed in the foundational study aligns most closely with the maneuver-based and interaction-aware models. The maneuver-based aspect is reflected in splitting trajectory into multiple maneuver classes, while the interaction-aware component is represented by leveraging the cars' coordinates in relation to neighboring cars. This section further discusses these concepts, along with a detailed description of the model introduced in the initial study.

2.1 Maneuver Based Approach

The prior work [8] uses distinct maneuvers to improve trajectory predictions. The maneuvers have two dimensions: lateral and longitudinal. Three lateral maneuvers refer to changing lanes to left or right or staying in the same lane. Since lane changes involve preparation and stabilization, the authors consider the vehicle to be in the changing lane state for $\pm 4s$ w.r.t the actual crossover. Two longitudinal maneuvers imply either keeping the speed or slowing down. The vehicle is defined as performing a braking maneuver if its average speed over the prediction horizon is less than 0.8 of its speed at the prediction horizon. The visual representation of maneuver classes is depicted in Figure 1 [8].

*Both authors contributed equally to this research.

Urban Computing'24, Dec 2024, Leiden, South Holland, The Netherland

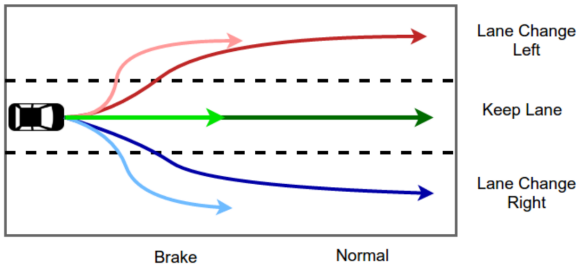


Figure 1: Lateral and longitudinal maneuver classes.

2.2 Interaction Based Approach

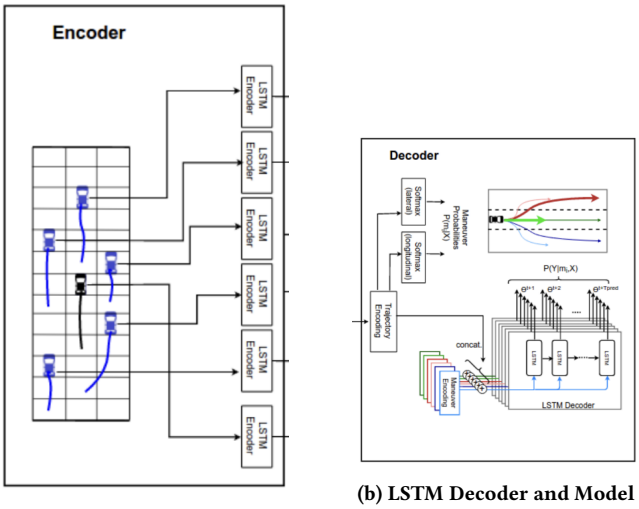
Interaction-aware models leverage the effect of inter-vehicle interactions to predict vehicles' motion. There are two approaches to incorporating inter-vehicle interactions. Firstly, hand-crafted cost functions based on relative configurations can be implemented, and the prediction of future motion is made with respect to these functions [4]. Designing hand-crafted cost functions has the main advantage of not relying on training data. However, creating a well-functioning cost function is tedious and unclear. The alternative approach used by the authors of the foundational study is data-driven, avoiding a tailored cost function. The common downside of this approach is the lack of variable data. However, this is not applicable in this case due to the properties of the used dataset.

2.3 LSTM for Vehicle Trajectory Prediction

Long Short-Term Memory networks are a widely used framework for handling sequential data. In trajectory prediction problems, capturing temporal dependencies is crucial. A previous study [3] introduced a model architecture comprising an LSTM encoder, a fully connected layer, and an LSTM decoder to predict human trajectories in crowded spaces. The LSTM encoder, as in Figure 2a [8], is responsible for learning the motion patterns from historical trajectories. It processes input in sequential time frames, using the most recent track history of a vehicle's velocity and all surrounding cars as input data. As the sequence progresses, the LSTM updates its states frame by frame, capturing temporal dependencies in the motion patterns. By the end of the sequence, the final LSTM state is expected to encode a comprehensive representation of the state of motion for the target vehicle, effectively summarizing its dynamic behavior over the observed period. The LSTM decoder, as in Figure 2b [8], generates predictions for future movements. It can either predict the maneuver probability or predict the probability distribution of vehicle motion.

2.4 Convolutional Social Pooling for Vehicle Trajectory Prediction

Convolutional Social Pooling (CSP) is a novel contribution introduced in the prior paper [8], designed to capture the interdependencies between the motion of vehicles efficiently. While LSTM networks are adept at modeling temporal information, CSP focuses on extracting spatial relationships. The authors of the previous study replaced the fully connected layer between the LSTM encoder and decoder with the CSP layer. The CSP structure relies



(a) LSTM Encoder. **(b) LSTM Decoder and Model Output.**

Figure 2: LSTM Encoder-Decoder architecture.

on a social tensor, as illustrated in Figure 3 [8], which is conceptually similar to how convolutional neural networks (CNNs) extract features from images [11]. Social tensor is a grid of a size $13x3$, where each of the 13 rows corresponds to a distance of 15 feet - approximately one car length. Three columns represent lanes: the current one, one to the right, and one to the left. The visual example of a social tensor can be found in Figure 4 [8]. The "Social Pooling" part of the acronym CSP indicates that LSTM states around the predicted vehicle are pooled into one social tensor. Then, multiple convolutional layers are applied to the social tensor, as shown in Figure 3. Lastly, a max-pooling layer is applied to it to achieve translational invariance. The last step of the model concatenates the fully connected neural network, which is applied to the LSTM state of the car, which trajectory we intend to predict with the max pooling layer.

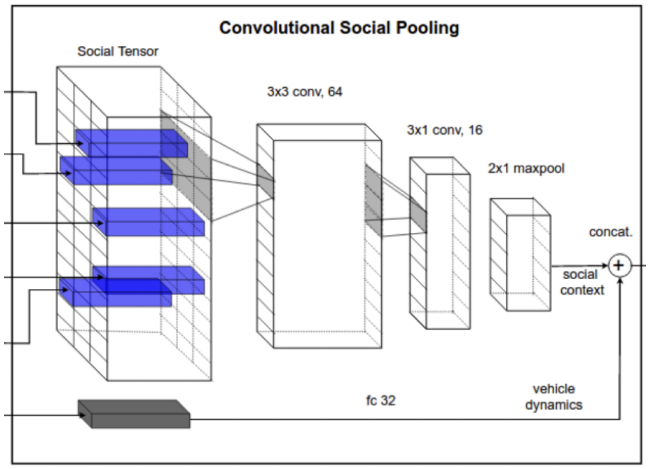


Figure 3: Convolutional social pooling layer.

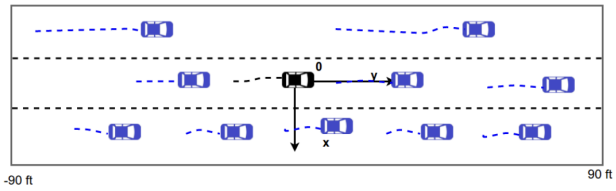


Figure 4: X and Y coordinates on the freeway.

3 Problem Statement

In previous research, capturing vehicle interactions has been a significant challenge for trajectory prediction. Furthermore, designing hand-crafted loss functions to account for vehicle interactions has proven complex and difficult to implement. The study by Deo, Nachiket, and Mohan M. Trivedi [8] addresses this issue by proposing a data-driven approach to model vehicle interactions, thereby improving overall model performance. The relationship between input and output is defined as shown in Equation 1.

$$\mathbf{Y} = \mathbf{M}_{csp\_lstm}(\mathbf{X}_{target}, \mathbf{X}_{neighbors}) \tag{1}$$

Here, $\mathbf{M}_{csp\_lstm}$ is the model described in Section 2, $\mathbf{X}$ represents a vector of $x^{(t)}$ - x and y coordinates of the vehicle recorded consequently for t time frames. The Equation 2 provides the formula for $\mathbf{X}$, and the Equation 3 shows the formula for $x^{(t)}$. $\mathbf{X}_{target}$ and $\mathbf{X}_{neighbors}$ correspond to the car whose trajectory is predicted and the neighboring cars, respectively.

$$\mathbf{X} = [x^{(t-t_0)}, ..., x^{(t-1)}, x^{(t)}] \tag{2}$$

$$\mathbf{x}^{(t)} = [x_0^{(t)}, y_0^{(t)}, x_1^{(t)}, y_1^{(t)}, ...x_n^{(t)}, y_n^{(t)}] \tag{3}$$

The model's output is a probability distribution over $\mathbf{Y}$, as shown in Equation 4, where each component $\mathbf{y}^{(t)}$ represents the future coordinates of the predicted vehicle, as illustrated in Equation 5. The motion is predicted over the next t_f frames in the future.

$$\mathbf{Y} = [\mathbf{y}^{(t+1)}, ..., \mathbf{y}^{(t+t_f)}] \tag{4}$$

$$\mathbf{y}^{(t)} = [x_0^{(t)}, y_0^{(t)}] \tag{5}$$

Intuitively, weather conditions can significantly influence driver behavior. For example, drivers are likely to reduce speed on rainy or foggy days. To account for this, we propose reconstructing the model to incorporate weather information as an additional input for trajectory prediction. The revised input-output relationship is illustrated in Equation 6. Here, the meaning of the variables is preserved from above. $\mathbf{X}_{weather}$ contains the values of visibility, precipitation, and wind speed. Units of measurement are miles, inches, and miles per hour, respectively. Importantly, since the modification only affects the model's input, the output dimension remains unchanged, ensuring ease of comparison and benchmarking.

$$\mathbf{Y} = \mathbf{M}_{csp\_lstm\_weather}(\mathbf{X}_{target}, \mathbf{X}_{neighbors}, \mathbf{X}_{weather}) \tag{6}$$

4 Methods

4.1 Data Preprocessing

We followed the methodology outlined in the original paper [8] to preprocess the trajectory data. Key features, including vehicle ID, frame ID, local X, local Y, and lane ID, are selected. Local X and local Y are the vehicles' coordinates with respect to the frame in which they are present. It was decided to use the local coordinates instead of the global ones for ease of computation. To simplify the data, lane IDs 7 and 8, which represent the US-101 freeway's up-ramp and down-ramp, are merged into lane ID 6, as they correspond to the same lane. This statement can be clearly verified in Figure 6. Maneuvers are defined based on changes in lane ID within an 8-second window, while the "slow down" action is determined when the vehicle's velocity drops to 80% of its historical velocity. Additionally, the relative positions of neighboring vehicles are represented using a 13 x 3 grid structure, as explained in detail in Section 2.4. Finally, the dataset is split into training, validation, and testing subsets.

We aligned the weather data with the trajectory data using global timestamps, ensuring that each vehicle was assigned the corresponding weather information. Key weather features, including precipitation, wind speed, and visibility, were chosen, as they can signal rainy and foggy conditions. The main issue was the disparity in data frequency, with weather measurements recorded hourly, while vehicle coordinates were recorded every 0.1 seconds. To resolve this, we aligned the data by extrapolating the weather measurements to match the higher frequency of the vehicle recordings.

4.2 Model Architecture Expansion

To introduce weather features, we reconstructed the model architecture. The new model is presented in Figure 5, suggesting a new input branch was added with the LSTM encoder and a fully connected layer of size 32. This new branch is parallel with the trajectory model. Before the model produces an output, both structures are embedded together to make the final decision. Furthermore, this framework shows a high flexibility for the model extension. If new features need to be considered, we can introduce them to this branch or create a new branch for them.

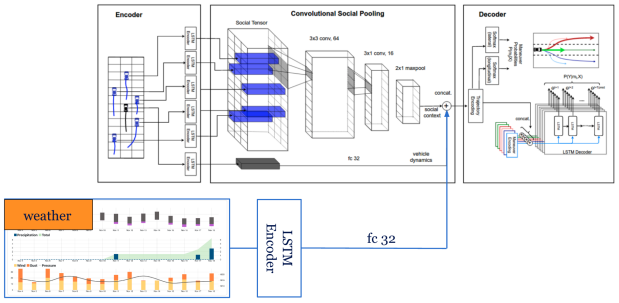


Figure 5: Convolutional social pooling LSTM model embedded with weather data.

4.3 Inputs and Outputs

The input of the expanded model is history trajectories combined with weather information. In Equation 7, the x and y coordinates at time t correspond to the vertical and horizontal directions of the freeway.

$$\mathbf{x}^{(t)} = [x_0^{(t)}, y_0^{(t)}, w_0^{(t)} ... x_n^{(t)}, y_n^{(t)}, w_n^{(t)}] \tag{7}$$

The w at time t suggests the weather features considered at a certain time. The output as in Equation 4 signifies the future location of the vehicle, kept in the same form as in the original paper [8]. Overall, the $\mathbf{x}^{(t)}$ from Equation 7 is exactly the ($\mathbf{X}_{target}$, $\mathbf{X}_{neighbors}$) as shown in Equation 1 in Section 3, when explaining the problem statement.

5 Experimental Setup

5.1 Data

5.1.1 Trajectory Data. This paper considers two trajectory datasets sourced from the Federal Highway Administration's (FHWA) Next Generation SIMulation (NGSIM) program. These datasets include freeway segments from US-101 [5] and I-80 [6], located in Los Angeles and San Francisco, California. As illustrated in Figure 6 [5], the study area focuses on US-101. Data was collected using eight video cameras, and the NGSIM-developed software transcribed the video recordings into vehicle trajectory data. Each dataset consists of trajectories captured at 10 Hz over a period of 45 minutes. Trajectory datasets include features such as X and Y coordinates, time frames, and lane IDs.

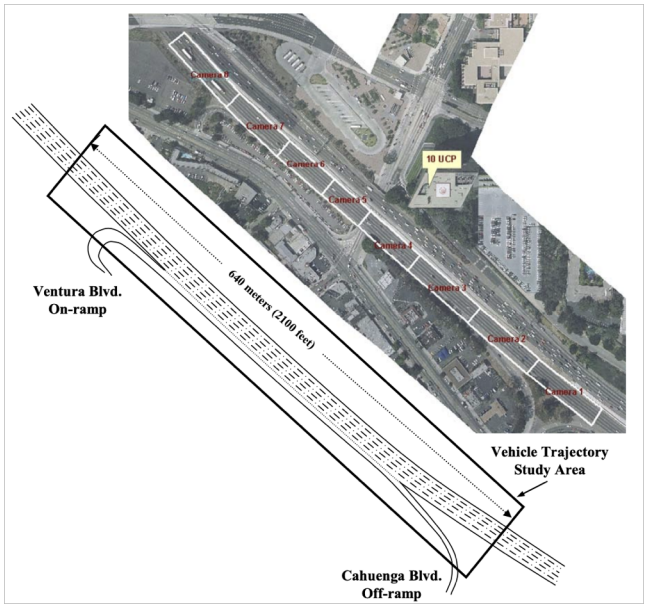


Figure 6: Study area for US-101 freeway.

Each dataset comprises 15-minute segments representing mild, moderate, and congested traffic conditions. Following the setup of the initial paper, the complete dataset is divided into training and testing sets, with the test set containing one-fourth of the trajectories from each of the three subsets of the US-101 and I-80 datasets. These trajectories are segmented into 8-second intervals, using 3 seconds of track history and a 5-second prediction horizon. The prediction horizon refers to the future frames for which the vehicle's coordinates are predicted. For example, a prediction horizon of 5 seconds corresponds to 50 frames, as the dataset is originally sampled at a rate of 10 Hz (10 frames per second). However, to reduce model complexity, the data is downsampled by a factor of 2, resulting in 25 frames for the 5-second prediction horizon. Thus, the performance of the model for the i^{th} second in the horizon is the average of its performance for each of the relevant 5 frames. This downsampling reduces the complexity of the model while maintaining sufficient temporal information for accurate trajectory predictions.

5.1.2 Weather Data. The weather data was obtained from the Visual Crossing website [7]. This website provides weather data for different regions, including Los Angeles, California. The hourly recorded data contains many features: temperature, humidity, precipitation, wind speed, visibility, etc. This study considered precipitation, wind speed, and visibility as the default weather aspects influencing drivers' behavior. The justification comes from common sense since heavy rain, strong wind, and poor visibility usually slow down responsible drivers. To check if a more holistic approach to weather can improve the model's performance even more, the default set of weather features is compared to the extended one, where temperature and humidity are added.

5.2 Evaluation Metrics

For the results to be comparable with the foundational paper, we adopted the metrics used by the authors, namely root mean square error (RMSE) and negative log-likelihood (NLL), as shown in Equations 8 and 9. While RMSE offers a concrete metric for assessing the predictive accuracy of models, it has limitations when evaluating multi-modal predictions. RMSE tends to favor models that produce averaged predictions across modes, which may not necessarily reflect accurate outcomes. For instance, a driver intending to overtake another vehicle might choose to either switch to the left or right lane while accelerating. An average of these two actions would suggest accelerating while remaining in the same lane, which is not a realistic prediction. To overcome this drawback, we also evaluate the NLL of the actual trajectories within the predictive distributions generated by the models. Although NLL values do not correspond to a directly interpretable physical quantity, they enable a meaningful comparison between uni-modal and multi-modal predictive distributions. A multi-modal algorithm gives a predictive distribution over all maneuver classes separately, while a uni-modal algorithm only provides the predicted coordinates. NLL was more relevant for the prior work [8], as they explicitly compared uni-modal and multi-modal versions, and in this study all the models compared are multi-modal. However, we still decided to preserve this metric for more detailed understanding.

In Equation 8 $\hat{y_i}$ is the predicted vehicle's coordinates and y_i is the true coordinate regarding the time frame i in the prediction horizon.

$$RMSE = \sqrt{\sum_{i}^{n} \frac{(\hat{y_i} - y_i)^2}{n}} \tag{8}$$

The Equation 9 aims to minimize NLL by adjusting the model's parameters ϕ so that model's output $f[x_i, \phi]$ is most likely to match the true values of y_i.

$$NLL = argmin_{\phi}(-\sum_{i}^{n} log(Pr(y_i|f[x_i, \phi]))) \tag{9}$$

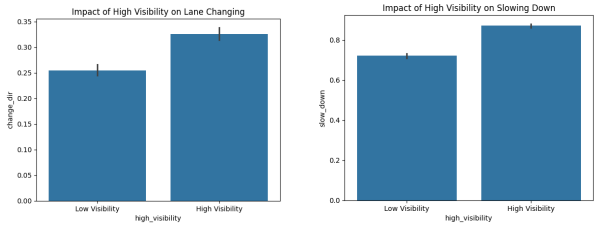
5.3 Model Training

The model training consists of two phases: pretraining with RMSE, and training with NLL. The first five epochs are pretraining, and the other three epochs are training. The training-validation-testing ratio is 70:10:20 in our experiment. The training batch size is 128, and the optimizer is Adam [9] with the default settings.

6 Results

6.1 Analysis of Weather Impact

To prove our assumption that weather conditions influence driving behavior, we compare the frequency of changing lanes and slowing down in different weather conditions as presented in Figure 7. The caption for the figure indicates that we only compare only in terms of visibility. Our analysis revealed that, within our dataset, high visibility, low humidity, and strong winds frequently always occur together, while low visibility, high humidity, and light winds are also always grouped. The higher rate of lane change and slowing down corresponds to the former weather conditions. While gathering more diverse data from other geographical locations beyond Los Angeles would strengthen this comparison, our findings highlight the importance of adopting a holistic approach to weather conditions. Rather than analyzing the impact of individual components in isolation, this shows the value of considering weather factors collectively for a more robust understanding.



(a) Comparison of lane-changing frequency under conditions of high visibility, low humidity, and stronger wind versus low visibility, high humidity, and weaker wind.

(b) Comparison of slowing-down behavior under the same conditions, with a slightly increased likelihood of slowing down during high visibility, low humidity, and stronger wind.

Figure 7: Impact of Visibility on Driving Behaviors.

6.2 Benchmark

Table 1 and Figure 8 present a comparison of the baseline model from the previous study [8], referred to as CS-LSTM(M), with two enhanced versions. The first advanced model, CS-LSTM-Weather(M)(3), incorporates additional weather features, including visibility, wind speed, and precipitation. The second enhanced model, CS-LSTM-Weather(M)(5), further expands the feature set by including temperature and humidity.

In terms of RMSE, the CS-LSTM-Weather(M)(3) model demonstrates better performance compared to CS-LSTM(M). At the same time, CS-LSTM-Weather(M)(5) achieves the best results, outperforming both throughout the whole prediction horizon of 5 seconds, as shown in the right-most column in Table 1. When comparing the baseline model with the model incorporating the default weather features in terms of NLL, the advanced model shows slightly worse performance for predicting coordinates 1 second into the future. However, it recovers and performs better for the rest of the prediction horizon. This could indicate that weather is more significant for longer-term horizons rather than for near-future predictions. The other explanation would be that this results from generalizing the hourly weather to all the recordings within this hour. The model with more weather characteristics outperforms both models in terms of NLL throughout the prediction horizon, as illustrated in Figure 8. This demonstrates that a more systematic approach to including weather conditions brings more accurate trajectory predictions. In the future, collaborating with meteorologists could explain this phenomenon in depth.

Evaluation Metric	Prediction Horizon (s)	CS-LSTM(M)	CS-LSTM -Weather(M)(3)	CS-LSTM -Weather(M)(5)
RMSE	1	0.7115	0.6954	**0.6904**
	2	1.5052	1.4869	**1.4672**
	3	2.4360	2.4022	**2.3686**
	4	3.5530	3.4955	**3.4654**
	5	4.9117	4.8337	**4.7684**
NLL	1	1.6924	1.6962	**1.6611**
	2	3.1566	3.1504	**3.1117**
	3	3.9732	3.9723	**3.9496**
	4	4.5956	4.5815	**4.5683**
	5	5.1193	5.0880	**5.0853**

Table 1: Metric comparison for CS-LSTM(M) and CS-LSTM-Weather(M).

6.3 Statistical test

The enhanced models outperform the baseline; however, statistical tests should be conducted to validate this improvement. Here, only RMSE is considered, as NLL has no physical interpretation. A paired sample t-test [1] is a great fit here, as the data distribution is naturally assumed to be normally distributed. The observations for all models are present for the same prediction horizon, facilitating the paired comparison. The null hypothesis assumes no significant difference between the models, and the alternative hypothesis is that a significant difference exists. Python's `scipy.stats` [2] library was used for the statistical tests, as it conveniently computes the t-statistic and p-value based on the differences between the

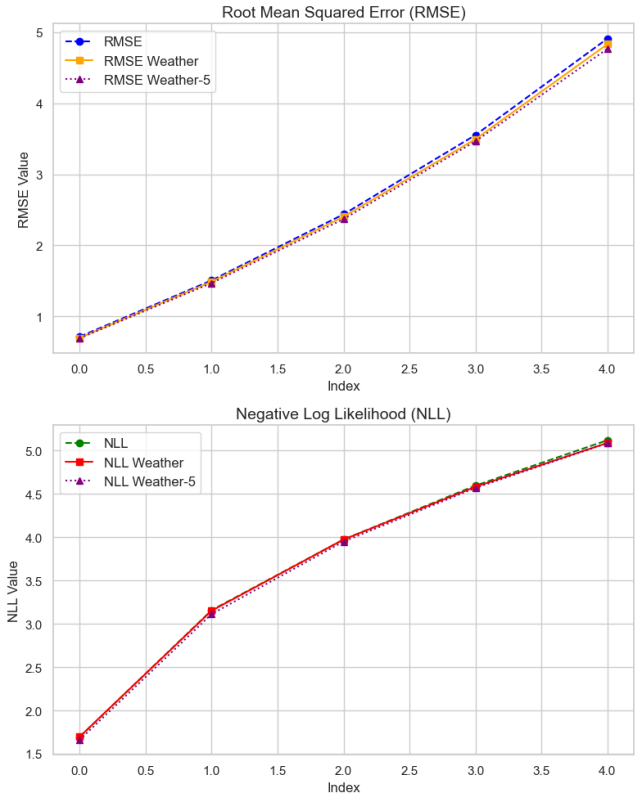


Figure 8: Metric comparison for CS-LSTM(M), CS-LSTM(M)(3) and CS-LSTM-Weather(M)(5). The index here represents the time of the predicted horizon minus 1 second.

paired values. Table 2 contains the t-statistic and p-values for all the pairs of compared models.

Comparison	t-statistic	p-value
CS-LSTM(M) vs CS-LSTM-Weather(M)(3)	3.4260	0.0266
CS-LSTM-Weather(M)(3) vs CS-LSTM-Weather(M)(5)	3.0837	0.0368
CS-LSTM(M) vs CS-LSTM-Weather(M)(5)	3.3522	0.0285

Table 2: Paired t-test results comparing CS-LSTM(M) and its weather-enhanced versions.

To interpret the result, the p-value is compared to a chosen significance level, which is decided to be of a default value of 0.05. If the p-value is smaller, reject the null hypothesis and conclude that the models differ significantly; otherwise, retain the null hypothesis. The p-values for all the comparisons are 0.0266, 0.0368, and 0.0285, all less than the significance level. Thus, the null hypothesis is rejected, and the performance difference is indeed statistically significant.

7 Conclusion

In conclusion, our research successfully integrates weather data with trajectory data and reconstructs the model structure. Most importantly, the proposed model outperforms the original model [8] on RMSE and majorly on NLL metrics. This improvement was proven to be statistically significant. It was discovered that incorporating more weather features can further improve model performance. A great limitation is the rare frequency at which the weather data was recorded. As a direction for future research, one can collect and incorporate dynamic weather data and compare it with the rarely sampled data to observe if the performance could benefit significantly. Such dynamic data can be obtained by equipping vehicles with weather sensors to capture real-time conditions. Additionally, the signs or landmarks on the road may provide useful information for trajectory prediction; however, there is no related information in our data. Overall, while implementing weather data poses certain challenges, our findings provide valuable insight into the significant impact of weather information on trajectory prediction.

Acknowledgments

Thanks for Professor Mitra. Urban Computing provides several different data types, such as temporal and temporal data. In addition, we learned how to visualize and analyze those data. Combining real-life topics and practical tools enhances our learning experience, boosting our study motivation.

References

[1] [n. d.]. Paired T-Test — statisticssolutions.com. https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/paired-sample-t-test/. [Accessed 08-01-2025].
[2] [n. d.]. Statistical functions (scipy.stats) &x2014; SciPy v1.15.0 Manual — docs.scipy.org. https://docs.scipy.org/doc/scipy/reference/stats.html. [Accessed 08-01-2025].
[3] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 961–971. https://doi.org/10.1109/CVPR.2016.110
[4] Mohammad Bahram, Constantin Hubmann, Andreas Lawitzky, Michael Aeberhard, and Dirk Wollherr. 2016. A Combined Model- and Learning-Based Framework for Interaction-Aware Maneuver Prediction. *IEEE Transactions on Intelligent Transportation Systems* 17, 6 (2016), 1538–1550. https://doi.org/10.1109/TITS.2015.2506642
[5] J. Colyar and J. Halkias. 2007. Us highway 101 dataset. Federal Highway Administration (FHWA). *Tech. Rep.* (2007).
[6] J. Colyar and J. Halkias. 2007. Us highway i-80 dataset. Federal Highway Administration (FHWA). *Tech. Rep.* (2007).
[7] Visual Crossing Corporation. [n. d.]. Weather Data & Weather API | Visual Crossing — visualcrossing.com. https://www.visualcrossing.com/. [Accessed 18-12-2024].
[8] Nachiket Deo and Mohan M Trivedi. 2018. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 1468–1476.
[9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[10] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. 2014. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal* 1 (2014), 1–14.
[11] Keiron O'Shea and Ryan Nash. 2015. An Introduction to Convolutional Neural Networks. *CoRR* abs/1511.08458 (2015). arXiv:1511.08458 http://arxiv.org/abs/1511.08458
[12] Yutian Pang, Xinyu Zhao, Hao Yan, and Yongming Liu. 2021. Data-driven trajectory prediction with weather uncertainties: A Bayesian deep learning approach. *Transportation Research Part C: Emerging Technologies* 130 (2021), 103326.