Intent Prediction of Vulnerable Road Users from Motion Trajectories Using Stacked LSTM Network

Khaled Saleh, Mohammed Hossny and Saeid Nahavandi
Institute for Intelligent Systems Research and Innovation (IISRI)
Deakin University, Australia
Email: {kaboufar, mhossny, saeid.nahavandi}@deakin.edu.au

Abstract*—Intent prediction of vulnerable road users (VRUs) has got some attention recently from the research community, due to its critical role in the advancement of both advanced driving assistance systems (ADAS) and highly automated vehicles development. Most of the proposed techniques for addressing the intent prediction problem have been focusing mainly on two methodologies, namely dynamical motion modelling and motion planning. Despite how powerful these techniques are, but they both rely on hand crafting a set of specific features which are scene specific, which in return affects their generalization to unseen scenes which involves VRUs. In this paper a novel end-to-end data-driven approach is proposed for long-term intent prediction of VRUs such as pedestrians in urban traffic environment based solely on their motion trajectories. The intent prediction problem was formulated as a time-series prediction problem, whereas by just observing a short-window sequence of motion trajectory of pedestrians, a forecasting about their future lateral positions can be made up to 4 secs ahead. In the proposed approach, we utilized the widely adopted architecture of recurrent neural networks, Long-Short Term Memory networks (LSTM) architecture to form a deep stacked LSTM network. The proposed stacked LSTM model was evaluated on one of the popular datasets for intent and path prediction of pedestrians in four unique traffic scenarios that involve pedestrians in an urban environment. Our proposed approach demonstrated competent results in comparison to the baseline approaches in terms of long-term prediction with small lateral position error of 0.39 meters, 0.48 meters, 0.46 meters and 0.51 meters respectively in the four scenarios of the testing dataset.

I. Introduction

In recent years, the development of autonomous vehicles (AVs) has been widely adopted by both major automotive manufacturers (such as Toyota, Mercedes Benz, GM, Ford, Audi, and more) and unicorn technology companies (such as Google and Uber). Despite the promised benefits that AVs would provide in minimizing the number of traffic accidents that are happening due to the human driver errors, they are still faced with a number of challenges [1]. Away from driving on highways, AVs still have some difficulties specially when it comes to driving in urban traffic environment such as the interaction with Vulnerable Road Users (VRUs) such as pedestrians [2]. Intuitively, interactions take place nowadays between human drivers and pedestrians are based on implicit cues between the two parties.

One of these key implicit cues are the initial motion trajectories of pedestrians which could convey their intentions to do certain actions [3]. Intent prediction of pedestrians in urban traffic environments using motion trajectories have been investigated over the past 5 years by the advanced driving assistance systems (ADAS) research community. However, most of the work focus on a set of pedestrians' dynamical motion models for predicting their intended trajectories [3]–[5]. Since the dynamical motion models assume that the target motion trajectories are all inhibit a similar dynamic, this make their prediction accuracies rather poor when predicting motion trajectories with different dynamics. For example, pedestrians can instantly change their motion dynamics from walking to stopping when trying to cross the road while walking on the road curbside. Thus, dynamical motion models suffer when predicting a longer prediction horizon.

Similarly, other approaches such as planning-based models [6], [7] from the robotics research community have also been investigated for the task of intent prediction of pedestrians in urban traffic environments from motion trajectories. In spite of their resilience to the longer prediction horizons problem of the dynamical motion models, but they still require a prior information regarding the final destination/goal of the pedestrians' motion trajectories [6]. Specially, with the fact that it is pretty hard to infer that goal from the perspective of a moving observer (i.e, the vehicle).

Additionally, both of the two approaches namely, dynamical motion models and planning-based models are rather limited due to their reliance on hand crafting different number of parameters and features selection for their operation which is specific to a particular settings or scenarios rather than obtaining them through data-driven approach. This, in return, restricts their capabilities to generalize to more complex or even unseen scenarios before.

In this work, we are proposing a data-driven approach for intent prediction of pedestrians in urban traffic environment from motion trajectories. Whereas, we formulate the task at hand as a time series prediction problem, and using a variant of Recurrent Neural Networks (RNN), Long-Short Term Memory networks (LSTM), we predict a long-term sequence (up to 4 secs) of pedestrians' motion trajectories in an urban traffic environment from a moving observer (i.e, vehicle). By using the proposed approach, we address the limitations that have been previously discussed regarding the dynamical motion models and the planning-based models.

The rest of this paper is organized as follow. Section II, provides an overview of the related work. In Section III, we present the details of our proposed RNN-LSTM model. In Section IV, quantitative and qualitative experimental results will be presented. Finally, we summarize our paper in Section V.

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

II. Related Work

Intent prediction of pedestrians in urban traffic environments has got some momentum over the past few years in the intelligent transportation and robotic communities. A brief overview regarding the related work in these areas will be discussed in this section.

A. Dynamical Motion Models

The most common approach has been relied on so far in the intelligent transportation systems for the task of intent prediction of pedestrians from motion trajectories is the dynamical motion model approach. Schneider et al. [3] used Bayesian filters such as Extended Kalman filter (EKF) [8] for vehicle-based pedestrian motion trajectory prediction within a short prediction horizon (less than 2 seconds) in four different motion dynamics (namely, bending-in, crossing, starting, stopping). They have also introduced another Bayesian filter based on Interacting Multiple Model (IMM) KF to accommodate different dynamical motion models of the pedestrians such as: constant velocity (CV), constant acceleration (CA) and constant turn (CT).

Similarly, Kooij et al. [5] proposed another dynamic motion model. They utilized a Dynamic Bayesian Network (DBN) for trajectory prediction of a pedestrians intention to cross the street while walking on the curb. They assumed what makes the pedestrian decide stopping to cross the street or continue walking on the curb depends on three factors: whether there is an approaching vehicle exist in a possible collision point, the awareness of this scenario by the pedestrian, the layout of physical environment around the pedestrian. By considering these factors as unobservable variables on top of a Switching Linear Dynamical System (SLDS) as a part of the DBN, they can predict to some extent the changes happen in the motion dynamics of pedestrians.

B. Planning-based Models

On the contrary, planning-based models for pedestrians' intent prediction do not explicitly model the dynamical motion of pedestrians' trajectories. However, they formulate the problem as a motion or path planning task, whereas they assume the pedestrian is a rationale agent have a hidden intention to reach an already known specific destination(s), and as a result will choose an optimal path (usually shortest path) to reach this goal. In [9], Rehder et al. proposed a model for long-term prediction of pedestrian intention to reach a specific destination by estimating the probability distribution over the future positions of the pedestrian using path planning techniques. Given the pedestrian position and orientation, and a grid occupancy map of the environment recorded online, they can estimate the pedestrians goal destination as a latent variable on-line, using a probabilistic planning-based technique. The grid occupancy map is discretized into independent cells with each cell containing a vector of location weighted features. For weights calculation, a supervised learning model was trained with ground truth trajectories of pedestrians and its corresponding grid map. They modeled the goal destination of the pedestrian as a Gaussian Mixture Model. Then, iteratively they improve the destination mixture components using a Particle filter.

III. Proposed Methodology

In this section we will firstly discuss the underlying operation of recurrent neural networks (RNNs), specially in the application of time series prediction tasks. Then, we will describe how our problem of interest, intent prediction of pedestrians from their motion trajectories, can be formulated as a time series prediction task using RNNs. Secondly, we will introduce our proposed approach to tackle the formulated problem. Finally, we will present the details of our proposed model for achieving a long-term intent prediction of the pedestrians in urban traffic environments.

A. Time Series Prediction using Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNN) have been widely used in many predicting sequence-based tasks such as: handwriting imitation [10], human gait analysis [11], human-human interactions [12]. Unlike traditional multilayer perceptron (MLP) neural networks, traditional RNNs have a feedback loop connection that can efficiently capture the temporal dependency in their input time series sequences by maintaining an internal state, called "hidden unit". However, traditional RNNs have difficult times in giving an accurate prediction when it comes to memorizing previous long sequences. Thus, the Long Short-term Memory (LSTM) RNN architecture [13] was introduced to help address this problem of traditional RNNs. The LSTM architecture has a basic unit in its hidden layer called memory block, that is similar to the hidden units of traditional RNNs. The LSTM's memory block has one or more memory cells, in addition to three gates (namely, input, output and forget gates) which are all shared by each cell in the memory block of the LSTM. The forget gate is responsible for deciding which information to throw away from the memory block. The input gate is responsible for deciding which values from the input to update the memory state. The output gate is responsible for deciding what to output based on input and the memory of the block. According to the architecture of LSTM proposed in [10], the operation of the hidden layer of LSTM memory cells (shown in Fig. 2) are calculated and updated each time step t according to the following equations:

$$f_t = sigm(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{1}$$

$$i_t = sigm(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2}$$

$$c_t = f_t c_{t-1} + i_t\, tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

$$o_t = sigm(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{4}$$

$$h_t = o_t * \tanh(c_t) \tag{5}$$

where f_t, i_t, o_t and c_t are the activations for the forget, input, output and cell state gates at time t respectively. While W_{*f}, W_{*i}, W_{*o}, W_{*c}, b_f, b_i, b_o, b_c are their respective weight matrices

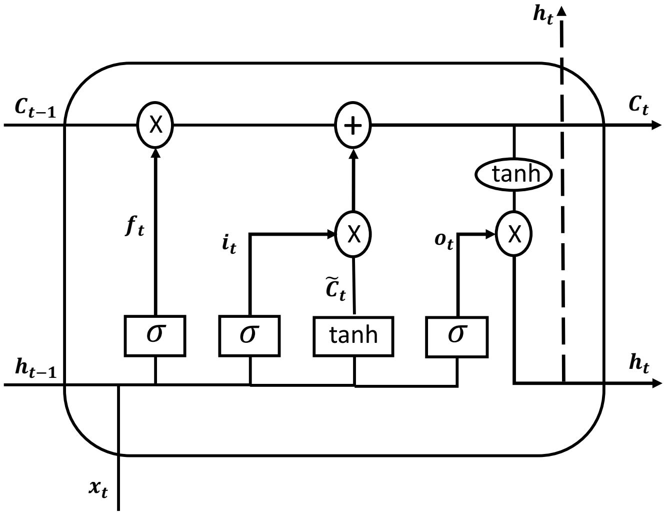


Fig. 1. LSTM memory block architecture [14]

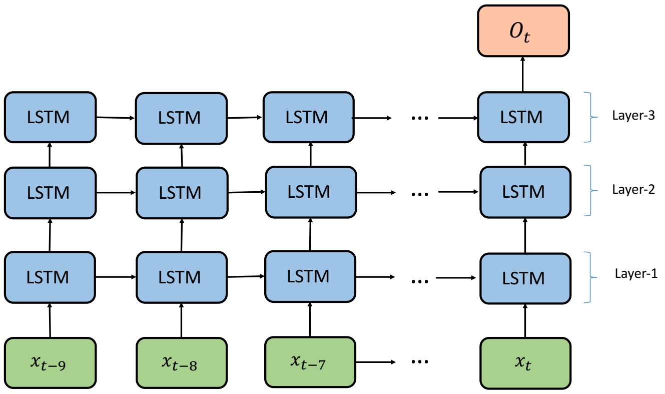


Fig. 2. Our proposed LSTM based model for intent prediction of pedestrians's of their motion trajectories.

and variable biases. x_t , h_t are the memory cell input and final output at time t. From previous equations, we can notice that, each gate from the three gates of the LSTM is just a composition of a sigmoid neural network layer and an element-wise multiplication operation. The sigmoid layer clips its input into a value between one and zero, whereas zero means ”no input will be passed through” and one means ”let the input to be passed through”.

B. Intent Prediction of VRUS using Stacked LSTM Network

Given the nature of motion trajectories of pedestrians, which is a time series signal of their lateral position on the ground measured at a regular time step as formulated in both dynamic motion models and planning-based models [3], [7]. We can formulate the task of intent prediction of pedestrians in urban traffic environment as a time series prediction problem, whereas at any time-step t the intent of any pedestrian on the road can be inferred from her motion trajectory. By observing the lateral position (u_t) of any pedestrian on the ground from time 1 to T_w, a prediction about her future lateral positions from time T_{w+1} to T_{pred} can be obtained.

In our proposed approach, we will be building a deep network of stacked LSTM blocks as discussed in Section III-A, for the task of intent prediction of pedestrians from their motion trajectories. Our stacked LSTM network consists mainly of three stacked LSTM layers as shown in Fig. 2. The first LSTM layer after Firstly, the input layer takes as input a 2-dimension sequence with a window size $w = 10$ of lateral position ($u_{1:w}$) of pedestrian motion trajectories. And the input layer feeds its input to the first LSTM layer of our stacked LSTM network, which has a size of hidden units/neurons equal to the input sequence window size w. The first LSTM layer in turn feeds into the second LSTM layer, which has a 100 hidden units. The second LSTM layer afterwards feeds into the final LSTM layer which has also other 100 hidden units. Finally, the last LSTM layer feeds into a fully connected layer which has only 1 neuron which corresponds to the predicted lateral position (u_{w+1}) at the next time step. The activation function of the final fully connected layer is the linear activation function, since we are predicting real value number. It is worth noting here, that we are only predicting the next time step of the input sequence during the training time only. However, at the inference or the testing time, we are recursively predicting any window-sized sequence determined at the inference time. The reason for that is to give us the flexibility to perform prediction with different sizes of input sequences of pedestrian motion trajectories without being restricted by the network architecture of our model or by the statistics of the data we are testing on. Whereas, if we build our model with a fixed size prediction window output, then we must use that size as the input sequence window size as well.

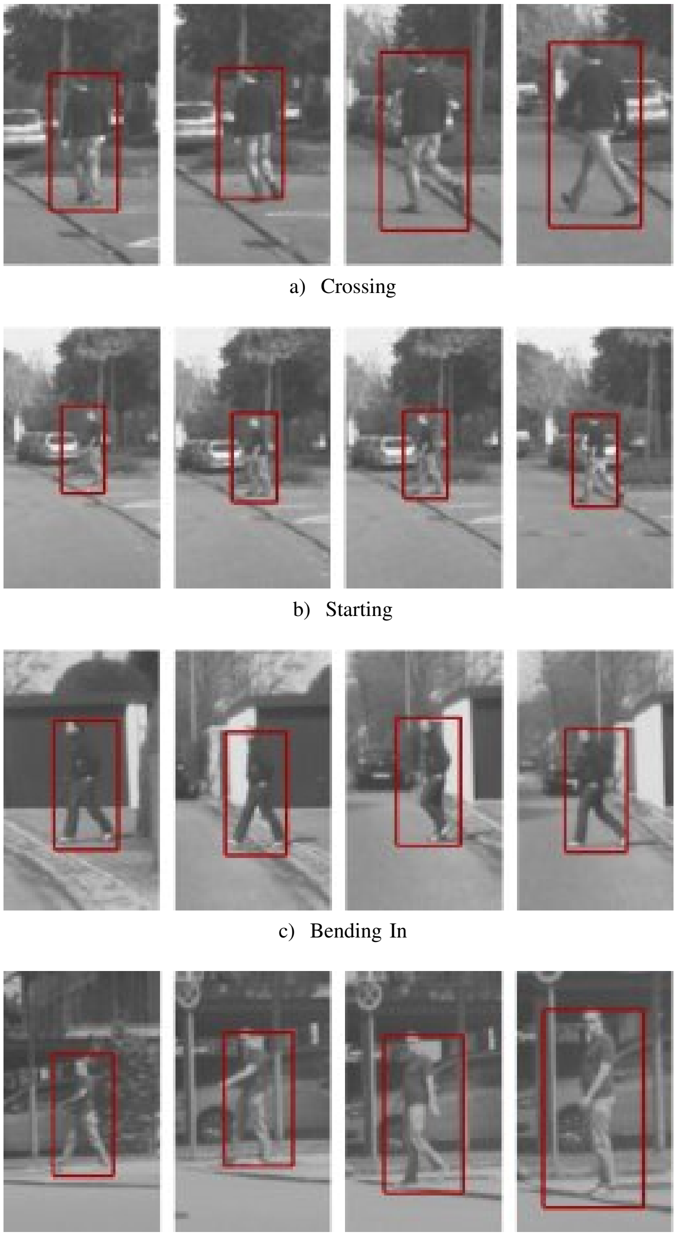
C. Stacked LSTM Network Training

Training any learning-based model for time-series prediction problems, is an optimization problem, where we are trying to minimize a loss function. One of the most commonly used loss functions in time-series prediction models (which is used as our loss function as well) is mean squared error (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2 \tag{6}$$

where N is the number of training samples, $\hat{Y}_i$ and Y_i are the predicted and target values for each sample.

For optimizing this loss function, we utilized the Adam optimizer for training our LSTM model [15]. Adam is a stochastic gradient descent algorithm that estimates the the gradient mean (1st-order moment) and element-wise squared gradient (2nd-order moment) of the gradient with the help of exponential moving average. One of the major reasons for using Adam in training our LSTM model is due to its low number of hyper-parameters (only the learning rate) need to be tuned when compared to other optimizers. In our stacked LSTM architecture for intent prediction of pedestrians we used the following hyper-parameters value: 0.001 for the learning rate. The selection for that value was based on a several



a) Crossing

b) Starting

c) Bending In

d) Stopping

Fig. 3. The four scenarios of Daimler pedestrian path prediction benchmark dataset [3].

training experiments using a 3-fold cross validation technique over the extracted samples of the training sequences. Finally, we have used a dropout with 20% after each LSTM layer to help regularize our network to prevent over-fitting during training [16]. We trained the stacked LSTM network for 10K training iterations with batch size of 512.

IV. Experiments

In this section we will give the details of the used dataset for training and testing our stacked LSTM model with, then we will outline the steps we followed to preprocess the dataset for our stacked LSTM model. Finally, the experimental results of our proposed stacked LSTM model performance on the testing data will be presented with comparison to other two major approaches for intent prediction of pedestrians.

A. Data Description

For training and testing the performance of our proposed stacked LSTM model for the intent prediction of pedestrians from their motion trajectory, we will use the widely popular Daimler pedestrian path prediction benchmark dataset presented in [3] (shown in Fig. 3). The dataset consists of 68 stereo image sequences captured from vehicle based cameras mounted behind the vehicle's windshield. The sequences cover four main scenarios, defined according to [3], involving different motion trajectory dynamics of pedestrian in urban traffic environment. In the first scenario, where a pedestrian is walking towards the street laterally to cross and is referred to as "Crossing" and the dataset contains 18 crossing sequences with 9 for training and 9 for testing. The second scenario, where a pedestrian is also walking laterally facing the street and then stop and is referred to as "Stopping" and the dataset contains 18 stopping sequences with 10 for training and 8 for testing. The third scenario, where a pedestrian is standing at the road curb and starts to walk facing the street laterally and is referred to as "Starting". The dataset contains 9 starting sequences with 5 for training and 4 for testing. The fourth and last scenario, where a pedestrian is walking beside the curb road and try to bend intending to cross the street. This scenario is referred to as "Bending in" and the dataset contains 23 Bending in sequences with 12 for training and 11 for testing.

Each sequence of the 68 sequences is labelled frame by frame with bounding boxes of pedestrian, median disparity of the upper body area of the pedestrian, position of the pedestrian in the vehicle coordinate system, and event tags "Time to Event" (TTE) values. TTE values correspond to the moment in each sequence of the four scenarios when the pedestrian is to cross, stop, start to cross or bend in to cross.

B. Data Preparation

Firstly, the dataset is pre-processed in order to feed it into our stacked LSTM model discussed in Section III during the training phase. The dataset is already split into a 36 training sequences and 32 testing sequences and each split is covering the four scenarios. The first step in our pre-processing stage, is we parse all the ground truth labels for lateral and positions (in meters) of the pedestrian in each sequence from the both training and testing sequences.

Then, we run a sliding windows of size $w+1$ with overlapping of value 1 on each sequence of the training dataset, whereas w is the size of the input layer that feeds into the first LSTM layer of our stacked LSTM model. As it can be shown from Fig. 2, we have an input window size w of 10. This results in a number of 4492 training samples of window size $w+1$ from all the training sequences. We further split our 4492 window samples of window size $w+1$ to a separate a training samples of window size w and target values of size 1. At the final stage of our pre-processing, we randomly extract 5% of the training window samples which corresponds to roughly

TABLE I
MEAN LATERAL POSITION ERROR (IN METERS) OVER ALL THE TESTING SEQUENCES OF EACH SCENARIO WITH TWO PREDICTION WINDOWS OF 70 AND STEPS AHEAD

		Bending in	Crossing	Starting	Stopping
EKF-CV [3]	Mean	1.09	0.72	1.31	**0.22**
	± Std	0.27	0.39	0.50	0.34
IMM-CV/CA [3]	Mean	1.08	0.68	1.32	0.24
	± Std	0.27	0.40	0.52	0.35
LSTM-2L [17]	Mean	0.79	1.21	0.76	1.01
	± Std	0.19	0.30	0.06	0.50
Stacked LSTM	Mean	**0.39**	**0.48**	**0.46**	0.51
	± Std	0.24	0.32	0.07	0.37

255 window samples to be used as a cross-validation set during training our stacked LSTM model.

C. Results

We evaluated the performance of our proposed stacked LSTM model on the testing sequences mentioned in Section IV-A. We followed the same procedure for the evaluation used in [3], whereas the evaluation done on the lateral position of each sequence of the testing dataset in TTE range [10, -50], which corresponds to a sequence of window size 60 in total with 0.60 secs before the event to 3.0 secs after the event. However, since we are interested in long-term intent prediction of pedestrians, we are predicting a 70 steps ahead (more than 4 secs) of lateral position, rather than the 32 steps prediction (1.9 secs) done in [3]. At testing time, we fed a 10 window size sequence before the starting of the TTE range [10, -50], and predict the whole TTE range. As earlier mentioned, in [3] they were relying on dynamical motion models using EKF and IMM-KF for their intent and path prediction system. We designed a similar dynamical models to the EKF (CV) and IMM (CV, CA) models implemented in [3] using the same parameters reported in their work as a baseline to compare our stacked LSTM model performance with. Additionally, we compared against another LSTM-based model that was proposed in [17] for a similar task of intent prediction of VRUs, however they reported it did not improve that much in comparison to other SVM-based model, besides they were also only considering the crossing scenario based on a private collected LIDAR dataset, we refer to that model as (LSTM-2L). They were using an LSTM model with two hidden layers of 64 and 128 units respectively without any fully connected layers. In Table I, we report the average mean error in lateral position (in meters) for 70-steps ahead prediction window over each scenario of the testing sequences. As it can be shown, our stacked LSTM model have a major lead in terms of smaller mean lateral position when compared to the EKF-CV, IMM-CV/CA and LSTM-2L models in all the testing scenarios except the “Stopping” scenario. After a thorough investigation of the “Stopping” scenario specifically, we believe that we found out why it is lagging behind the other two dynamical model approaches. Since we chose not to include the outliers in the training dataset to maintain the same training/testing fold of the dataset for comparison with [3], there are more than two unique testing sequences in the “Stopping” scenario which do not have even a trend-like or a similar sequences in all the training dataset, which in returns affects the mean lateral position error value. Fig. 4 shows four sub-figures which corresponds to lateral position error for the four testing scenarios of our proposed LSTM model compared to the EKF-CV and IMM-CV/CA dynamical models.

The lateral error in Fig. 4, is the step-wise Euclidean distance between the predicted lateral positions and the ground truth lateral positions at each time step with prediction horizon of 70 steps ahead. As it can be noticed, our proposed stacked LSTM model has an improvement in terms of the lateral position error of up to 0.85 m, 0.7 m and 0.24 m in the “Starting”, “Bending In” and the “Crossing” scenarios respectively. We have also compared the performance of our proposed stacked LSTM model with a smaller prediction horizon of only 15 steps ahead in order to see how it performs in comparison to the dynamical motion models that are usually better in prediction with smaller k-steps ahead. However, our proposed stacked LSTM model is generally still performing better than both the dynamical motion models (EKF-CV and the IMM-CV/CA models) and the LSTM-2L model. Another observation from the results, that the stacked LSTM layers do improve the prediction results when compared to shallow LSTM network (LSTM-2L).

TABLE II
MEAN LATERAL POSITION ERROR (IN METERS) OVER ALL THE TESTING SEQUENCES OF EACH SCENARIO WITH PREDICTION WINDOWS OF 15 STEPS AHEAD

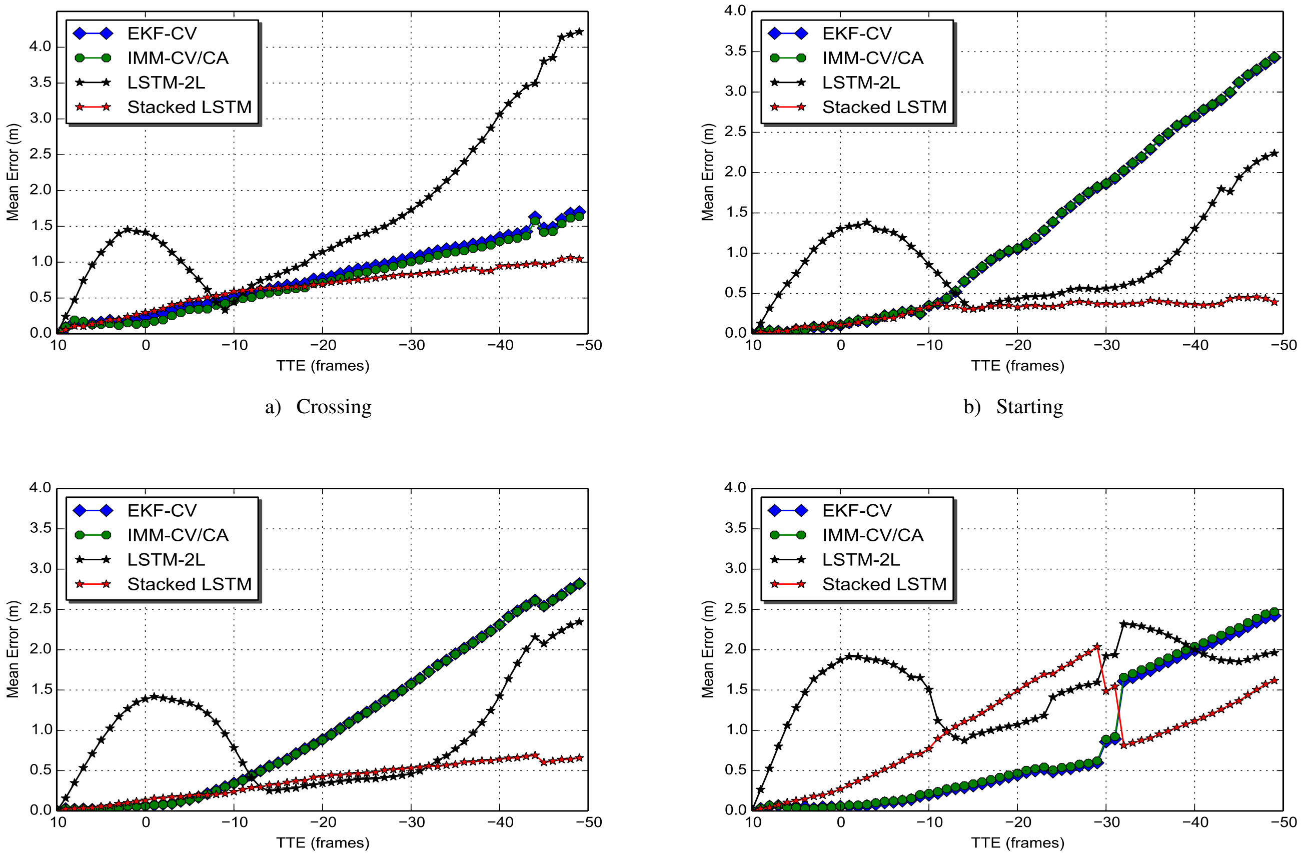
		Bending in	Crossing	Starting	Stopping
EKF-CV [3]	Mean	0.44	0.58	0.44	**0.03**
	± Std	0.12	0.07	0.03	0.01
IMM-CV/CA [3]	Mean	0.48	0.66	0.49	0.05
	± Std	0.13	0.08	0.05	0.01
LSTM-2L [17]	Mean	0.32	0.54	0.34	0.76
	± Std	0.09	0.08	0.04	0.34
Stacked LSTM	Mean	**0.04**	**0.07**	**0.05**	0.09
	± Std	0.02	0.03	0.01	0.05

V. CONCLUSION

In this paper we have presented a novel data-driven approach for the task of long-term intent prediction of pedestrians from motion trajectories. We formulated the problem as a time-series prediction problem and trained a stacked LSTM architecture using a publicly available dataset of pedestrian motion trajectories in urban traffic environment. Our proposed stacked LSTM architecture have achieved a superior result in terms of smaller mean lateral position error in short and long-term prediction horizons over most of the four scenarios of the testing data in comparison with two of the most commonly used dynamical motion models for intent prediction.

ACKNOWLEDGMENT

This research was fully supported by the Institute for Intelligent Systems Research and Innovation (IISRI) at Deakin University.



a) Crossing

b) Starting

c) Bending In

d) Stopping

Fig. 4. Lateral position error with prediction horizon of 70 steps ahead (almost 4 secs) over all the the four scenarios of the testing dataset [3].

References

[1] D. J. Fagnant and K. Kockelman, “Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations,” *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[2] M. Mara and C. Lindinger, “Talking to the robocar - new research approaches to the interaction between human beings and mobility machines in the city of the future,” pp. 86–91, 2015.

[3] N. Schneider and D. M. Gavrila, “Pedestrian path prediction with recursive Bayesian filters: A comparative study,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8142 LNCS, pp. 174–183, 2013.

[4] C. G. Keller and D. M. Gavrila, “Will the pedestrian cross? A study on pedestrian path prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[5] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, “Context-based pedestrian path prediction,” in *European Conference on Computer Vision*. Springer, 2014, pp. 618–633.

[6] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, “Activity forecasting,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7575 LNCS, no. PART 4, pp. 201–214, 2012.

[7] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, “Planning-based prediction for pedestrians,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 3931–3936, 2009.

[8] S. M. Mohamed and S. Nahavandi, “Robust finite-horizon kalman filtering for uncertain discrete-time systems,” *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1548–1552, 2012.

[9] E. Rehder and H. Kloeden, “Goal-Directed Pedestrian Prediction,” *IEEE International Conference on Computer Vision Workshops*, 2015.

[10] A. Graves, “Generating sequences with recurrent neural networks,” *arXiv preprint arXiv:1308.0850*, 2013.

[11] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, “Recurrent network models for human dynamics,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4346–4354.

[12] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-fei, and S. Savarese, “Social LSTM : Human Trajectory Prediction in Crowded Spaces,” *CVPR*, 2016.

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

[14] C. Olah, “Understanding lstm networks,” 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[15] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[16] S. Nitish, “Improving neural networks with dropout,” *Diss., University of Toronto*, 2013.

[17] B. Völz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto, “A data-driven approach for pedestrian intention estimation,” in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 2607–2612.