https://doi.org/10.3311/PPtr.15849
Creative Commons Attribution ⓘ

Periodica Polytechnica Transportation Engineering, 48(4), pp. 327–333, 2020

Lane Change Prediction Using Gaussian Classification, Support Vector Classification and Neural Network Classifiers

Olivér Rákos[1*], Szilárd Aradi[1], Tamás Bécsi[1]

[1] Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, H-1111 Budapest, Stoczek street 2., Hungary
[*] Corresponding author, e-mail: rakos.oliver@mail.bme.hu

Received: 03 March 2020, Accepted: 11 March 2020, Published online: 30 June 2020

Abstract
It is essential for a driver assistant system's motion planning to take the vehicles moving in the surroundings into account. One of the most crucial driver intentions which should be predicted is lane changing. It has been investigated whether it is possible to reliably classify lane-changing maneuvers in a highway situation using learning algorithms such as Gaussian-classifier, SVM, and LSTM neural networks. Real vehicle trajectories are extracted from the NGSIM US-101 and I-80 datasets. The input for the classifiers is derived from the trajectory by selecting a subset of the features: lateral and longitudinal position coordinates, longitudinal acceleration, and velocity. In such an environment, the vehicle movement is limited, so it has been tested that how sufficient if only the mean and the variance of the derivative of lateral coordinate was taken as input for the classification had been tested. Different strategies for labeling the input sequences were tested.

Keywords
behavior prediction, maneuver detection, SVM, LSTM, NGSIM

1 Introduction

According to the most widespread approach, the architecture of autonomous vehicles is described as a model of hierarchically based subsystems (Paden et al., 2016), where the system input is the desired destination from the perspective of the travel, which is aided by the environment sensing layer collecting information from sensors. The perception layer fuses the data provided by the sensors, performs path detection, models static and dynamic objects, and classifies them in order to map the vehicle environment. The output is a geometric model of the path and a list of static and dynamic objects with trajectory data. This, as well as the user-provided information, is received as input by the path planning layer, which is responsible for determining the traffic situation, behavior planning, and short-term route planning. The output of this layer is the input to the motion control layer, which includes the logic that controls the feasibility of the given itinerary, designing the trajectory with various constraints and comfort considerations in mind. This layer instructs the actuators to execute the actual motion and then senses the vehicle's motion status. Behavior analysis and prediction of other agents in the traffic situation is an essential part of behavior planning. For planning, it is essential to estimate as accurately as possible the trajectories of other vehicles and other road users.

This article discusses lane change maneuver prediction in a highway situation. It compares different machine learning techniques, such as the Gaussian classifier, Support Vector Classification (SVC), and Long-Short Term Memory (LSTM) recurrent neural network. To conduct this research, the vehicle trajectory data from the Next Generation Simulation (NGSIM) US-101 (Colyar and Halkias, 2007) and I-80 Highway Database (Colyar and Halkias, 2006) were used.

The challenging task of behavioral prediction for autonomous driving attracts great attention. There is a wide variety of probabilistic approaches in the literature due to the intensive research that has taken place over the past decade. To list all the important milestones or higher results for the sake of completeness is not the objective of this paper, but to mention some inspirational article affecting our work. In Dagli et al. (2002) a hierarchical dynamic Bayesian network has been utilized for predicting behavioral patterns. Gaussian process regressions have been

Cite this article as: Rákos, O., Aradi, Sz., Bécsi, T. (2020) "Lane Change Prediction Using Gaussian Classification, Support Vector Classification and Neural Network Classifiers", Periodica Polytechnica Transportation Engineering, 48(4), pp. 327–333. https://doi.org/10.3311/PPtr.15849

also used for trajectory pattern classification and prediction for autonomous driving (Trautman and Krause, 2010).

Another approach is behavioral prediction applying Bayesian changepoint detection. This aims to benefit the tracked historical data to infer the likelihood of their possible future actions (Galceran et al., 2017). In that work, a decision-making algorithm has been introduced to approximate Partially Observable Markov Decision Processes (POMDP) to evaluate the predicted outcomes of interactions between vehicles.

In (Wiest et al., 2012), a Gaussian mixture model for trajectory prediction is proposed. It can predict the vehicle's trajectory several seconds in advance via learning the previously observed patterns to infer a joint probability distribution as a motion model. The future trajectory is predicted by calculating the probability for the motion, conditioned on the current observed trajectory. The novelty is that the result is not only a prediction, but a distribution over the future trajectories and a specific scenario can be predicted by the evaluation of the statistical properties.

Kalman and particle filters also can be used to estimate the motion of a car that is near the ego vehicle (Toro et al., 2018). They proffer a method by which the state of maneuvering vehicle can be estimated with a good result, and the correct mode of operation from a previously defined set is chosen.

A novel approach presents an LSTM model for motion prediction of surrounding vehicles on freeways, which is aware of all the surrounding vehicles (Rodrigues et al., 2016). This model's output is not a single motion trajectory but a multi-modal distribution over future motion, just like in the previous article.

The accuracy of the lane change maneuver prediction on the highway based on the trajectory of the past was examined. In the studied situation, the vehicles can travel in multiple parallel lanes on a straight section. Such a maneuver is preceded by a decision, and the trajectory of the vehicles traveling in the same and in the adjacent lane in front of and behind it plays a significant role. Accordingly, this information must be considered if the intention of changing lanes is needed. However, if not the intention but their occurrence in the short term is required to predict, the trajectory data of a given vehicle may be sufficient.

High-reliability lane change maneuver prediction was achieved by combining Support Vector Machine (SVM) and Artificial Neural Network (ANN) methods (Dou et al., 2016). Our hypothesis was that the change in the lateral coordinate (lateral velocity) carries sufficient information to predict the occurrence of the lane change and that the mean and standard deviation of this time series is sufficient. The theoretical part provides a summary of the data set, data preparation, labeling strategies, and algorithms tested.

2 Theoretical part

In order to reach the goal that has been defined in the Section 1, different classifiers' performances were examined. The details of the fitting and evaluation are unfolded in Sections 3 and 4.

One is the Gaussian Classifier, which suggests that the class likelihood functions are Gaussian distributed with unique mean vectors and covariance matrices. To fit this model to each class, one must estimate the mean, the covariance, and the class priors. Thus, the posterior probability of any input vector for a given class can be expressed. Assuming a common covariance matrix for each class, the discriminant function and the decision boundaries are linear, while in the case of different covariance matrices, they are quadratic. The Gaussian assumption makes the model very limited, however easy to train and interpret, thus helping to highlight the importance of labeling.

The other is the Support Vector Classifier (SVC). This considers different suggestions for class likelihood functions. It strives for maximum marginal separation, so better generalization ability is expected. Polynomial, sigmoid, and RBF kernel functions were applied. There are two types for the regularization of SVC: the C SVC and the ν SVC. C and ν are regularisation parameters by which a penalty can be applied to the misclassifications. The parameters' range $C \in [0, \infty)$ while $\nu \in (0,1]$ is an upper bound on the ratio of training errors and a lower bound of the ratio of support vectors.

The last one is the Long-Short Term Memory (LSTM) recurrent neural network. This unit can deal with long time series without the occurrence of the so-called exploding and vanishing gradient problem. In this research, the model performances were compared with different input features.

2.1 The training dataset

The NGSIM US-101 (Colyar and Halkias, 2006) and I-80 (Colyar and Halkias, 2007) datasets were used for training and evaluating the lane change classification problem. This trajectory data has precise location, velocity, and acceleration of each vehicle within a particular area every seconds. Furthermore, it provides relative positions to surrounding vehicles and lane position in every frame. There are 11.8 million rows and 25 columns in this dataset.

Each row represents one vehicle in a specific frame with all the information. The columns in ascending order are the following:

- Vehicle identification number,
- frame identification number ascending by start time,
- total number of frames in which the vehicle appears,
- global time,
- lateral (*x*) and longitudinal (*y*) coordinate of vehicle's front center to the left-most edge of the section in the direction of travel,
- global *x*, *y* coordinate,
- ength and width of vehicle,
- vehicle type (1 - motorcycle, 2 - auto, 3 - truck),
- instantaneous velocity and acceleration,
- lane position.

Although the available data is rich, this research intends to find out if less data, small computation, and memory requirements could be used to classify a specified trajectory part as a lane changing or keeping maneuver.

An aerial photograph on Fig. 1 shows the US 101 study area from where the vehicle trajectory data were collected. A schematic picture describes the lanes and the on-ramp and the off-ramp locations.

2.2 Define the input for the classification

The input in the k^{th} frame is constructed by T times delayed lateral coordinate vector $x(k,T)=\left[x(k),x(k-1),...,x(k-T)\right]$. The model based on the supposition that the maneuver can be determined purely by the $\Delta x(k,T)=\left[\Delta x(k),\Delta x(k-1),...,\Delta x(k-T)\right]$, where $\Delta x(k)=x(k)-x(k-1)$ because only is the lane changing important, not the actual lane index itself. The smallest input space consists only of the average value and the standard deviation of $\Delta x(k)$. After the evaluation, it became clear that the performance of the Gaussian classifier or the SVM can be improved considerably by adding additional features to the input, such as the longitudinal velocity or acceleration. In the case of the neural network, the $\Delta x(k)$ values with have been supplemented by the velocity *v*(*k*) and acceleration *a*(*k*) values, or the longitudinal positions as well. The input is for this the Gaussian Classifier. Since the SVM algorithms are not scale-invariant, it is recommended to standardize the input. In order to do this, the input was transformed to zero mean and unit standard deviation before fitting the SVM models.

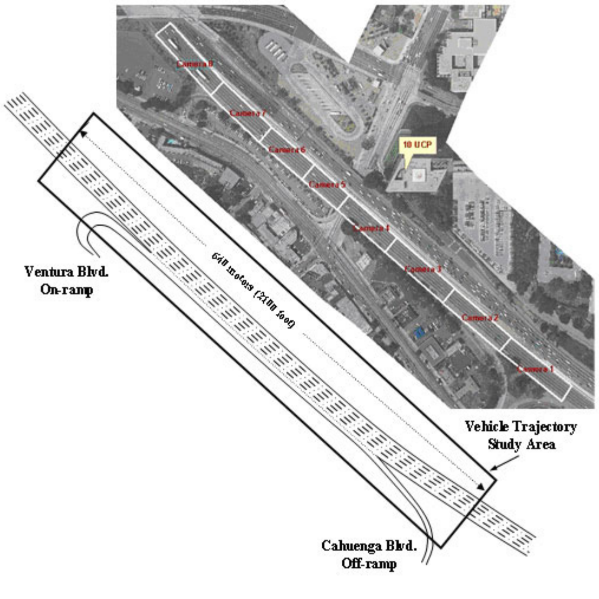

Fig. 1 Top view of the freeway from which the data was extracted.

2.3 Gaussian classification

Let us describe the posterior probability for a given input *x*, which corresponds to classes $\mathcal{C}_1=\text{lane keep},$ and $\mathcal{C}_2=\text{lane change}.$ It can be written as[1]:

$$p(\mathcal{C}_2\,|\,x)=\frac{p(x\,|\,\mathcal{C}_2)p(\mathcal{C}_2)}{p(x\,|\,\mathcal{C}_1)p(\mathcal{C}_1)+p(x\,|\,\mathcal{C}_2)p(\mathcal{C}_2)}=\sigma(a), \tag{1}$$

where:

$$a=\ln\frac{p(x\,|\,\mathcal{C}_2)p(\mathcal{C}_2)}{p(x\,|\,\mathcal{C}_1)p(\mathcal{C}_1)}, \tag{2}$$

and $\sigma(a)$ is the logistic sigmoid function. For the case of $K=3$ classes $\mathcal{C}_1=\text{lane keep}$ while $\mathcal{C}_2=\text{lane change left}$ and $\mathcal{C}_3=\text{lane change right},$ one gets:

$$p(\mathcal{C}_k\,|\,x)=\frac{\exp(a_k)}{\sum_j\exp(a_j)}, \tag{3}$$

which is known as the softmax function and

$$a_k=\ln p(x\,|\,\mathcal{C}_k)p(\mathcal{C}_k). \tag{4}$$

One can live on the assumption, that likelihood densities for classes $\mathcal{C}_k$ are Gaussian with mean μ_k and common covariance matrix $\sum$, that is $p(x\,|\,\mathcal{C}_k)=\mathcal{N}(x;\mu_k,\Sigma)$, then considering two classes and using Eq. (2), one has:

$$p(\mathcal{C}_1\,|\,x)=\sigma\left(w^T x+w_0\right), \tag{5}$$

[1] Notation: *t*(*k*)represents the correspondence of *x*(*k*) input vector to one of the classes $\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_K$. In the case of $K=2,$ the two classes are lane keep and lane change, while in the case of $K=3,$ the three classes are lane keeping, lane change left, and lane change right.

where:

$$\begin{aligned} w &= \Sigma^{-1}\left(\mu_1-\mu_2\right) \\ w_0 &= -\frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1+\frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2+\ln\frac{p\left(\mathcal{C}_1\right)}{p\left(\mathcal{C}_2\right)}. \end{aligned} \tag{6}$$

The quadratic term in x is cancelled due to common covariance matrix.

For the general case, using Eq. (4), one has:

$$a_k\left(x\right)=w_k^Tx+w_{0,k}, \tag{7}$$

where:

$$\begin{aligned} w_k &= \Sigma^{-1}\mu_k, \\ w_{0,k} &= -\frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k+\ln p\left(\mathcal{C}_k\right). \end{aligned} \tag{8}$$

Without the assumption of a common covariance matrix, it will no longer be linear but quadratic.

2.4 Support vector classification

The concept of SVM is to maximize the smallest distance between the decision boundary and the closest samples (maximum margin). The data set is $x\left(k\right)\in\mathcal{X}, k\in\mathcal{I}=\left[1,K\right]$, where N is the number of data, the corresponding labels are $t(k)$. This is equivalent to solve:

$$argmin_{w_b}\frac{1}{2}w_b^2, \tag{9}$$

subject to the condition:

$$t\left(k\right)\left(w^T\phi\left(x\left(k\right)\right)+b\right)\geq 1, \forall k\in\mathcal{I}. \tag{10}$$

This optimization can be used for multiclass problems either with one-vs-one or one-vs-rest shape as long as the dataset is linearly separable in the feature space $\phi\left(\mathcal{X}\right)$. The class distributions overlap. The modification to this approach is that data points are allowed to be misclassified, that is, to be on the wrong side of the margin boundary with a penalty ξ_n decreasing with the distance from that boundary. Therefore, the minimization problem is:

$$argmin_{w_b}\frac{1}{2}w_b^2+C\sum_{n=1}^{K}\xi_n;0<C\leq\infty, \tag{11}$$

subject to the condition:

$$t\left(k\right)\left(w^T\phi\left(x\left(k\right)\right)+b\right)\geq 1-\xi_k, \forall k\in\mathcal{I}. \tag{12}$$

An alternative formulation of C-SVC, known as the v-SVC has the parameter v. Which replace C, can be interpreted as an upper bound on the fraction of points which lie on the wrong side of the margin, and also as a lower bound on the fraction of support vectors.

The *SVC* formalism:

$$argmin_{w_b}\frac{1}{2}w_b^2+C\sum_{n=1}^{K}\xi_n+\widehat{\xi}_n;0<C\leq\infty, \tag{13}$$

subject to the conditions:

$$\begin{aligned} t\left(k\right)-\epsilon-\xi_k &= w^T\phi\left(x\left(k\right)\right)+b \\ t\left(k\right)+\epsilon-\hat{\xi}_k &= w^T\phi\left(x\left(k\right)\right)+b\ \forall k\in\mathcal{I} \end{aligned}. \tag{14}$$

2.5 LSTM recurrent neural network

LSTM recurrent neural networks were trained with multiple input single output model to predict the lane changing maneuver. The input is a window-sized sequence of frames. In a current timestep, the input dimension is $N = 1$, 2, or 3. The output is 3 dimensional. A single layer LSTM was used with an N-dimensional input size, 7-dimensional hidden state size, and the output was reduced by a linear layer from 7 to 3. Finally, softmax nonlinearity was used to generate the output. Increasing complexity, such as increasing the number of layers, or neurons, slows down the training and does not produce better results.

2.6 Labelling approaches

Several approaches have been used to label time series. From the dataset, the moment is exactly known when the lane index change occurred. This is used to decide when to consider the movement of a vehicle in a given window to be changing or keeping the lane.

In the first approach, all time windows are assumed to be lane keeping in which the lane index does not change, and all that shifts are assumed to be a lane changing. This method is insensitive to the occurrence within the time window. If the lane change is at the border of the time window, then two very similar time windows will be labeled in different classes, leading to a high degree of overlap between them.

In the second approach, the following non-overlapping time window is used to label the given time window. Based on this, the time window is considered to be a lane keeper if there is no lane index change in that and the next one. A lane-changer is considered to have a lane change in one of the time windows. This reduced overlap but did not yield better results.

The basic idea of the third approach is to leave the ambiguous time windows out of the labeling; that is, one should refuse to label the transitional windows. At the beginning and at the end of the time window, a range corresponding to a given ratio is left, in which if the lane change occurs, the time window is not labeled, and there will be no part of either the training data nor the test data. It is considered to be a keeper if lane change in the entire time window does not occur.

In the fourth approach, the lane-keeping samples are prepared by sampling vehicles that have no lane change at all in the data set. From the data of lane changing vehicles, samples of right and left shifting trajectories were taken. In addition to the length of the time window, a shift parameter was also introduced, which is an integer number lesser than the time window. This is the number of frames by which the moving time window is shifted during the sampling. This prevents overly similar sequences and avoids over-fitting and data memorization.

3 Results of Gaussian classifier and the SVM

The SVC and *v*-SVC models were fitted with different hyper-parameters. The kernel function can be any custom function, but this research focused on the linear, polynomial, RBF, and sigmoid functions. The kernel coefficients are always set to be $1/N \cdot Var(x)$, where *N* is the number of features, and *Var*(*x*) is the variance of the data. The constants *C* and *v* are chosen by grid search. The F_1 scores calculated on the test set are in Table 1.

4 Results of neural networks

The fourth approach has been used to label the sequences. The training samples are given by sequences cut out of the 2 window size ranges preceding the lane change. Since the lane-changing frequency is low, the training set would be unbalanced, so the size of the training set had been adjusted to the minimum of the left and right lane changes. The data loader is capable of generating a training set with customizable window size and a shift parameter so that models can be widely tested. The shift parameter is the time step delay of the consecutive sequence. During the training, ADAM optimizer was used, and the loss function was binary cross-entropy. The data loader defines what data and how to be included in the training set.

Table 1 The results of the classifiers.

Classifier	parameter	(left, keep, right)
SVC (ovr)	*C* = 3.16	0.777; 0.683; 0.639
SVC(ovr)	*v* = 0.45	0.769; 0.689; 0.648
Gaussian	-	0.26; 0.11; 0. 57

Three cases for the input had been examined. Firstly: $\Delta x(k), v(k), a(k)$, secondly: $\Delta x(k), y(k)$, and finally, only the $\Delta x(k)$ values. The best scores are produced by de second case. The F_1 scores of the three maneuver classes are shown in Figs. 2 to 4 with respect to the number of epochs. The learning curves and accuracy with learning rate 0.05 are on Figs. 5 to 7. The window size and the shift parameter are chosen to be 30 and 5 empirically.

The training was performed using multiple learning rates from 0.001 to 0.5 in logarithmic scales. With too small learning rates, the accuracy increases to near 0.7 after thousands of epochs, while the value of training loss and validation loss freezes around 0.4 without over-fitting. The explanation for this may be that it oscillates around a local minimum.

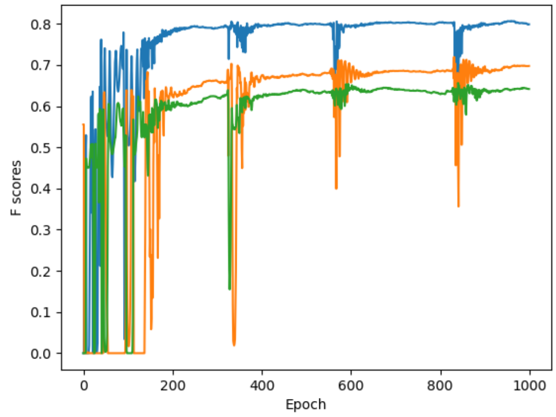

Fig. 2 The F_1 scores of left, keep, right maneuver classes (blue, orange, green respectively) Input: $\Delta x(k), v(k), a(k)$.

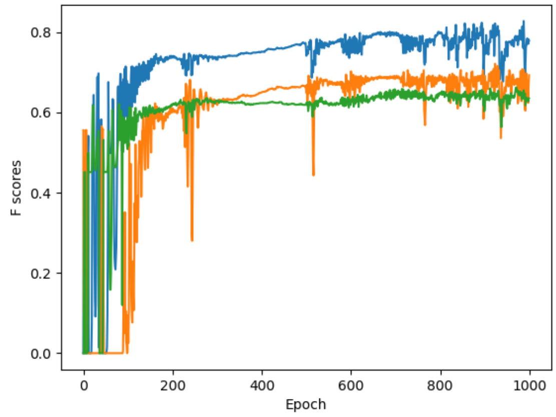

Fig. 3 The F_1 scores of left, keep, right maneuver classes (blue, orange, green respectively) Input: $\Delta x(k)$.

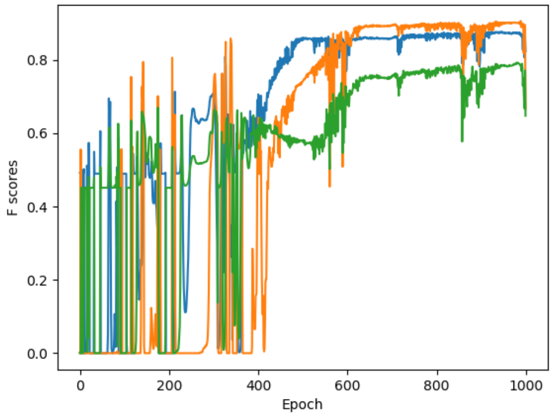

Fig. 4 The F_1 scores of left, keep, right maneuver classes (blue, orange, green respectively) Input: $\Delta x(k), y(k)$.

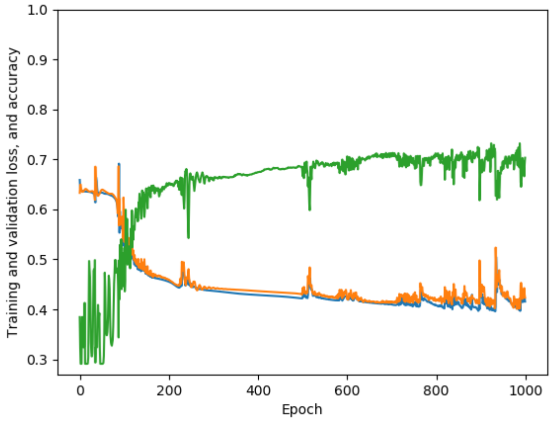

Fig. 6 The training and validation loss (blue, orange) and prediction accuracy (green) with respect to the number of epochs. Input: $\Delta x(k)$.

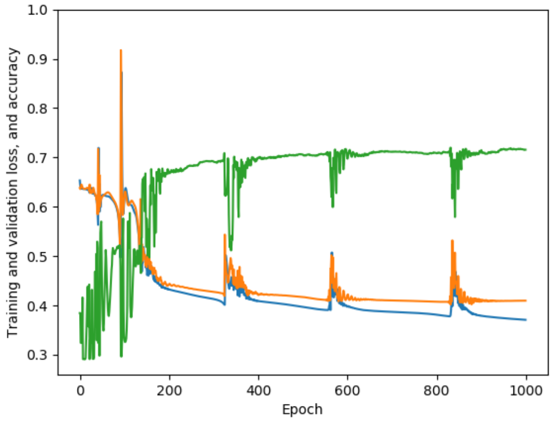

Fig. 5 The training and validation loss (blue, orange) and prediction accuracy (green) with respect to the number of epochs. Input: $\Delta x(k), v(k), a(k)$.

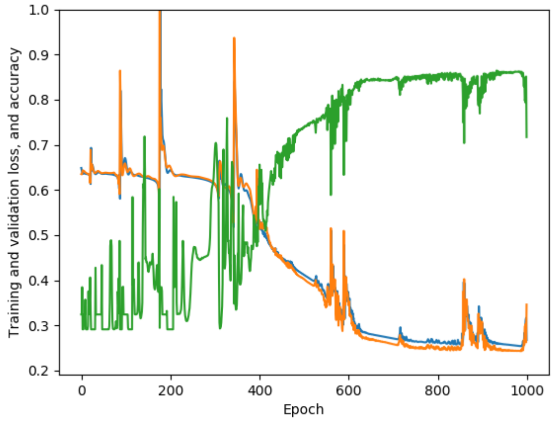

Fig. 7 The training and validation loss (blue, orange) and prediction accuracy (green) with respect to the number of epochs. Input: $\Delta x(k), y(k)$.

At higher learning rates, learning either does not converge or fluctuates too much. Using weight decay could not achieve better results. If the value was set to 0.05, the learning curve converged after 1500 epochs, and the accuracy fluctuated around 0.76. The training and validating loss are the smallest in this case. Increased complexity has also been tried, with two layers of LSTM and 7 neurons with learning rate 0.01. At 1000 epochs, the system over-fitted and gave worse results. If done at learning rate 0.05, it became unstable and did not converge. Both the validation and training loss are lower while the accuracy and F_1 scores are higher in case of the input is . The accuracy fluctuates around 0.86, which is the best result, bigger by at least 0.1 than in the other cases.

5 Conclusion

The evaluated algorithms are capable of detecting the occurrence of the lane change maneuver in a freeway situation with different results. Little information was used from trajectories to train Gaussian Classifier and the whole time series to train SVM and neural networks. The Gaussian classifier trained with the two-dimensional vector of the mean and variance proved to be unuseful for making high accuracy predictions. Approximately the same result were achieved for the SVM trained by the time series of $\Delta x(k)$ values and neural network classifiers trained by longitudinal values $\Delta x(k), v(k), a(k)$. In a highway situation and in this particular classification task, the lateral position changing (lateral velocity) carries sufficient information, and the longitudinal parameters do not contribute.

Despite all these, the LSTM network fed by the lateral position change and the longitudinal position $(\Delta x(k), y(k))$ yielded significantly higher accuracy. It means that this is the most appropriate input for this particular classification task with three different lateral maneuver classes. As future work, the optimal input should be thoroughly investigated for the improved task in which the three lateral classes are doubled by longitudinal ones. Longitudinal classes could be braking and not braking based on the consideration that the driver receives information that the stop lamp is operating or not.

Discrete values were used for labeling. However, continuous labels could be used to encode that after how many seconds will the lane change occur. It would make the prediction more accurate. Hyper-parameters for the SVM, like C, ν, γ, and the kernel type, are not directly learned during the training. The hyper-parameter space can be searched for the best cross-validation score using an exhaustive grid search.

If the hyper-parameter space is too wide, Sequential Model-Based Optimization (SMBO) (Hutter et al., 2009) or Sequential Model-based Algorithm Configuration (SMAC) (Hutter et al., 2010) could be applied. The best-performing labeling could also be achieved by optimizing the SVM classification parameters and then doing the training with the labeled data.

In order to make better predictions, it is essential to consider more information about the traffic situation. As a first approach, one should take into account the trajectory or even the type of surrounding vehicles. This possibility will be investigated in the future. It would be worthwhile to examine whether a complete set of trajectory data for nearby vehicles is needed or the standard deviation and mean are sufficient. This would reduce the runtime of the algorithm or the resource requirement in real-life automotive applications.

As a next step, the predictive abilities of Auto Encoders shall be investigated. By training such an incomplete network for copying the input trajectory to the output can yield a context vector with a smaller dimension than the input. This encoded context vector can be used for classification, but most importantly, it could be used for trajectory prediction tasks as well. It would be possible to test whether such networks can encode the trajectory of the vehicle under investigation with the information of vehicles moving around it. First of all, one should train the network for yielding the future trajectory of the vehicle to the output by feeding the historical trajectory of the ego vehicle and the surroundings to the input. Secondly, it is possible to separate the input encoding (and copying) task to the prediction task. The comparison of the two approaches can be instructive in designing the architecture of the networks.

References

Colyar, J., Halkias, J. (2006) "Us highway 80 dataset", Federal Highway Administration, U.S. Department of Transportation, Emeryville, CA, USA, Tech. Rep. FHWA-HRT-06-137.

Colyar, J., Halkias, J. (2007) "US highway 101 dataset", Federal Highway Administration Research and Technology, U.S. Department of Transportation, Los Angeles, CA, USA, Rep. FHWA-HRT-07-030.

Dagli, I., Brost, M., Breuel, G. (2002) "Action Recognition and Prediction for Driver Assistance Systems Using Dynamic Belief Networks", In: Net.ObjectDays: International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World, Erfurt, Germany, pp. 179–194. https://doi.org/10.1007/3-540-36559-1_15

Dou, Y., Yan, F., Feng, D. (2016) "Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers", In: 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, Canada, pp. 901–906. https://doi.org/10.1109/AIM.2016.7576883

Galceran, E., Cunningham, A. G., Eustice, R. M., Olson, E. (2017) "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment", Autonomous Robots, 41(6), pp. 1367–1382. https://doi.org/10.1007/s10514-017-9619-z

Hutter, F., Hoos, H. H., Leyton-Brown, K., Murphy, K. P. (2009) "An experimental investigation of model-based parameter optimisation: SPO and beyond", In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, Montreal, Québec, Canada, pp. 271–278. https://doi.org/10.1145/1569901.1569940

Hutter, F., Hoos, H. H., Leyton-Brown, K. (2010) "Sequential Model-Based Optimization for General Algorithm Configuration (extended version)", University of British Columbia, Computer Science, Vancouver, Canada, Tech. Rep. TR-2010-10.

Paden, B., Čáp, M., Yong, S. Z., Yershov, D., Frazzoli, E. (2016) "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles", IEEE Transactions on Intelligent Vehicles, 1(1), pp. 33–55. https://doi.org/10.1109/TIV.2016.2578706

Rodrigues, M., McGordon, A., Gest, G., Marco, J. (2016) "Adaptive tactical behaviour planner for autonomous ground vehicle", In: 2016 UKACC 11th International Conference on Control (CONTROL), Belfast, UK, pp. 1–8. https://doi.org/10.1109/CONTROL.2016.7737631

Törő, O., Bécsi, T., Aradi, S., Vellai, A. (2018) "Multimodel state estimation in road traffic using constrained filtering", In: 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, pp. 000205–000210. https://doi.org/10.1109/CINTI.2018.8928200

Trautman, P., Krause, A. (2010) "Unfreezing the robot: Navigation in dense, interacting crowds", In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, pp. 797–803. https://doi.org/10.1109/IROS.2010.5654369

Wiest, J., Höffken, M., Kreßel, U., Dietmayer, K. (2012) "Probabilistic trajectory prediction with Gaussian mixture models", In: 2012 IEEE Intelligent Vehicles Symposium, Alcala de Henares, Spain, pp. 141–146. https://doi.org/10.1109/IVS.2012.6232277