Estimating pedestrian intentions from trajectory data

Elena Šikudová*†, Kristína Malinovská*, Radoslav Škoviera*, Júlia Škovierová*, Miroslav Uller*, Václav Hlaváč*
*CIIRC, Czech Technical University in Prague, Czech Republic
Email: Name.Surname@cvut.cz

†Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic

***Abstract*—In autonomous driving systems, intention estimation of traffic participants is one of the most crucial aspects. In this paper, several machine learning methods are used to train classifiers capable of estimating the intention of a pedestrian to cross a zebra crossing. Their results are compared to a Bayesian network – an approach commonly used in autonomous driving. The data used for the estimation contain only position and heading of the pedestrians. The best performing method achieved the F_2 score of 92.37%.**

***Index Terms*—Intention estimation, Machine learning, Automated driving.**

I. Introduction

The goal of scenario understanding in autonomous driving applications is to correctly estimate the other traffic participants' intentions and predict their motion. Pedestrians are the most vulnerable traffic participants. They are also capable to change their speed and heading quite quickly. For these reasons, one of the most crucial tasks of the scenario understanding, is the estimation of the pedestrians' intentions.

Our goal is to evaluate several approaches in the task of estimating the pedestrian intentions to cross the zebra based solely on the trajectory data. The trajectory data are easily obtained using sensors commonly used in autonomous driving applications.

II. Related work

Understanding the current traffic situation is a key intermediate step on the way towards a self-driving car. The estimation of the traffic participant‘s behavior is usually based on the observations of the surrounding context and the previous motion patterns of other involved traffic participants. However, motion prediction of dynamic objects based solely on physical laws (e.g., physics-based motion prediction) does not suffice, but rather the intentions of these dynamic objects have to be taken into account.

The systems for motion and intention predictions are mainly focusing on other vehicles (e.g. [1], [2], [3]). However, the safety of other traffic participants, especially the pedestrians, in complicated urban traffic situations is crucial. Therefore, the correct prediction of pedestrian‘s intention and motion (e.g. intention to cross the street) can be very helpful in an successful effort to avoid accidents.

In the past years, several different systems for pedestrian intention estimation and motion prediction were proposed. These systems work with a variety of input data such as pedestrian dynamics, pose and head orientation. However, our work focuses on more simplistic data containing only the object classification, position, and heading of traffic participants. These data are easier to obtain with more conventional hardware likely to be implemented in consumer autonomous cars.

Based on the study from [4], for dealing with humans around the vehicle, several different aspects have to be analyzed, such as intent and activity prediction, skill and style of the traffic participant, attention model, general activity classification and behavior analysis. This study includes a large number of approaches for pedestrian motion prediction and intention estimation and all of these approaches use pedestrian dynamics. However, most of them are focusing on additional information such as head orientation, social context and body pose. These additional features can be very useful for the correct estimation of pedestrians' intentions. However, the studies regarding pedestrian intention estimation still vary considerably in terms of the type of the cues and computer vision techniques employed.

In [5], the authors proposed a generic context-based model to predict crossing behaviors of pedestrians in intra-city scenarios. This model provides accurate predictions at an early time. The authors considered a multi-model system, where they explored specific locations such as zebra crossings, which where based on expert driving experience.

The approach from [6] presented an intent prediction system for the use in advanced driver assistance systems. This system uses a monocular view of the road. The authors used cues such as GPS position and pedestrian dynamics, similar to the data used in our study. For the behavior prediction, the authors used a particle filter. The predictions have been computed for the desired prediction horizon, and a bi-variate Gaussian was fitted over the particles. The authors indicate a prediction time 5 steps ahead. However, they did not define the length of the step. Another shortage is in the map information, where the authors used map information from the OpenStreetMap, which can be insufficient in multiple cases. Since the OpenStreetMap is a collaborative community project the quality of the map often varies. It also lacks some categories of information such as the structure and type of the traffic lanes. Last, but not least, the proposed method does not utilize ego-motion compensation, so the system does not work for moving vehicles.

In [7], the image-based 2D pose estimation using CNNs is presented. The authors combined CNN-based pedestrian detection, tracking and pose estimation for prediction of the intention to cross the street. In [8] the authors proposed a real-time framework, which learns intention recognition using weak-supervision and locomotion dynamics of intention from pose information using transfer learning. The method for prediction of future pedestrian paths, poses, and intentions up to 1s in advance is presented in [9]. This method is based on balanced Gaussian process dynamical models (B-GPDMs), which reduce the 3-D time-related information extracted from key points or joints placed along pedestrian bodies into low-dimensional spaces. This method contains four main models of pedestrian activity (walking, stopping, starting and standing) and selects the most similar model to estimate future pedestrian states.

The pedestrian intention estimation using motion information similar to our data is presented in [10]. This information is obtained by accelerometer carried by pedestrians. The authors used different variations of kNN and SVM classifiers and obtained prediction accuracy 84.8%.

Similar research is presented in [11], where the authors use Support vector machines and dense and Long-Short-Term-Memory networks to classify the intention of the pedestrians. They use handcrafted features as well as CNN derived features. They evaluate the results in respect to time and distance to cross. Support vector machines achieved over 90% accuracy in predicting crossing pedestrians correctly up to 2 seconds and 3 meters to cross.

III. Our approach

Our current focus is on pedestrians, namely on intention estimation near zebra crossings. Our approach is heavily dependent on the available data.

A. Data description

Due to the project setting we only have access to the high-level preprocessed data. We could not use any semantic cues e.g. body posture or head orientation extracted from the video stream. The objects (cars, pedestrians) in our data are represented by their centroid positions.

We extract our data from high-level representation of the map of the environment and the processed information from the ego-car. The data are processed on the frame-level, so each data item in our dataset represents one pedestrian in one particular time moment. We have cleaned the data to include only pedestrians to avoid the noise introduced in the preprocessing.

The data available from the system sensors include

- time: timestamp, frame number
- identity: pedestrians, zebras, ...
- position (x,y): pedestrians, cars, ...
- velocity: pedestrians, cars
- orientation: pedestrians' headings
- map data: lanes, zebra anchor points, curbs, ...

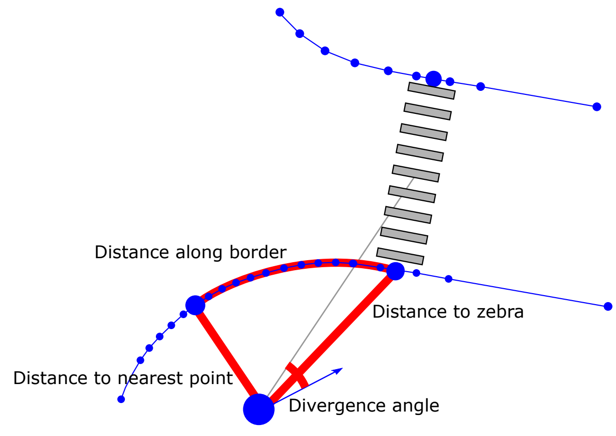

Fig. 1: Raw data from the sensors in blue. Computed features in red.

From the pedestrian data and the map data we can compute more informative data like the time of the pedestrian crossing or entering the road, the position of the nearest road or the crossing entry point, the distance from the zebra (Euclidean or along the road) or from the nearest road border, the angle between pedestrian's heading and the line to the zebra center, etc.

Our final derived features used in classification tasks are

- **Distance to zebra**: the distance between the position of pedestrian and the zebra anchor point,
- **Distance to nearest point**: the distance between the nearest road border point and the pedestrian position,
- **Distance along border**: the distance between the nearest road point and the zebra anchor, measured along the border curve and
- **Divergence angle**: the angle between pedestrian's current heading direction and pedestrian-to-zebra direction.

The features are visualized in Figure 1. All features are measured in two consecutive time points. The delay between the timepoints of measurement was chosen to be 0.6s (representing the current situation and the situation 0.6s ago). The dataset contains continuous values for standard ML methods and also discretized values for Bayesian nets or classical reasoning.

In the raw data available to us, there is a strong imbalance among classes. Namely, most of the tracked pedestrians are not crossing the lane. This phenomenon can be observed in Figure 2, where the crossing and non-crossing pedestrian trajectories are color-coded. The majority of the pedestrians are not crossing (magenta colored dots) and some are crossing (blue colored dots).

When using an imbalanced dataset, the prediction models learn to classify all pedestrians as non-crossing. We compensated the imbalance in the dataset by oversampling the minority class (the crossing pedestrians). The amount of data available in the augmented dataset is 2278882 rows.

B. Intention estimation

We considered and tested several machine learning (ML) approaches to pedestrians' intention estimation. The first ap-

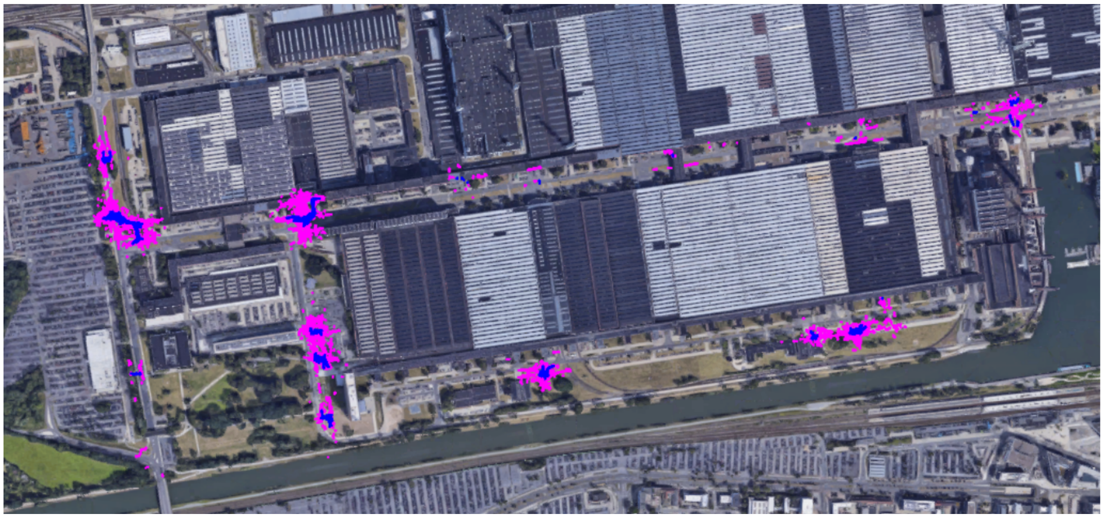

Fig. 2: Collected data. Color-coded trajectories of non-crossing (magenta) and crossing (blue) pedestrians.

proach - the Bayesian network (BN) - was inspired by our previous work [12]. The data (the derived features) for the BN prediction were discretized into two classes

$$x_{discr} = \begin{cases} \text{Far} & \text{if } x > \theta_x \\ \text{Close} & \text{otherwise,} \end{cases}$$

where θ_x is the threshold for the feature x. The thresholds were found by via grid search. The outcome of the Bayesian network classification is the probability of the pedestrian crossing the zebra.

Bayesian networks are generally slow and computationally demanding. Therefore we searched for alternatives and tested several other methods. These classifiers were trained with the same features as the Bayesian net, but without discretization (e.g. distances in meters instead of Far/Close). Again data from the current and previous time frame were used. We investigated several simple ML methods, their bagged versions and other meta-estimators:

- K nearest neighbors (KNN),
- Bagged K nearest neighbors (bKNN),
- Linear Discriminant Analysis (LDA),
- Bagged Support Vector Machines (bSVM),
- Random forests (RF),
- Extremely randomized trees (ERT).

The outcome of these methods is a binary classification of the pedestrian intention to cross the zebra.

In the following sections we will describe these methods in detail. The chosen parameters of the presented methods were estimated using a grid search on a smaller dataset.

1) Bayesian network: Bayesian network [13] is a probabilistic graphical model consisting of a directed acyclic graph and a join probability distribution. In the graph, a directed connection $X \rightarrow Y$ means that X causes Y with a given probability. The conditional distribution for each node is given its parents only $P(X_i|\text{Parents}(X_i))$. For discrete variables it is represented as a conditional probability table giving the distribution over $X_i$for each combination of parent values.

The structure of the proposed Bayesian network for the intention estimation of pedestrians near a zebra crossing can be seen in Figure 3.

The implemented Bayesian network solution processes the data from two consecutive timepoints of pedestrian trajectory history and locations of nearby roads and crosswalks.

2) K nearest neighbors: KNN classifier [14] is a memory based classifier that needs all the training data stored for classification. To classify a feature vector $\mathbf{x}$, the distances to all training vectors are computed and K closest points are selected. The most occurring class is the resulting class.

In our experiments we used $K = 501$, Manhattan distance measure

$$d_{\text{Manhattan}}(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

and KD tree algorithm for finding the nearest neighbors. The votes of the neighbors were inverse distance weighted, meaning that closer neighbors influence the decision more than the distant neighbors.

3) Bagged K nearest neighbors: bKNN is an ensemble estimator. Each of the KNN base estimators is trained on a bootstrap replica of the training set. The decision is made by voting. We used $K = 40$, Chebyshev distance measure

$$d_{\text{Chebyshev}}(\mathbf{p}, \mathbf{q}) := \max_i(|p_i - q_i|)$$

with uniform neighbor weighting, where all the neighbors influence the decision equally.

4) Linear discriminant analysis: LDA is a classifier with a linear decision boundary using Bayes' rule [14]. It assumes that the conditional probability of X given Y is a multivariate normal distribution with equal covariance matrices $\mathbf{\Sigma}$ in the classes

$$P(X = \mathbf{x}|Y = y) = \frac{1}{\sqrt{(2\pi)^D |\mathbf{\Sigma}|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_y)},$$

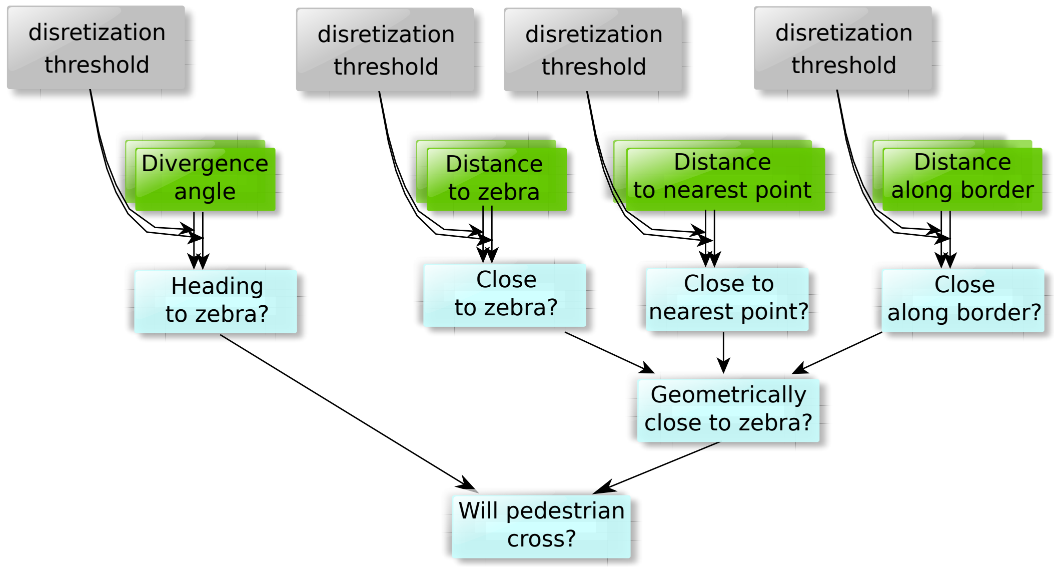

Fig. 3: Graphical representation of our Bayesian network.

where $\boldsymbol{\mu}_y$ is the mean vector of each class. A feature vector $\mathbf{x}$ is classified into the class y^* that maximizes the value of $P(X = \mathbf{x}|Y = y)$.

The LDA classifier tends to overfit less than the quadratic discriminant analysis (QDA), where the covariance matrices are different across the classes and the decision boundary is quadratic.

5) Bagged Support Vector Machines: bSVM is an ensemble classifier [14] of several kernel SVMs. The decision function of the binary SVM classifier is $\text{sign}(\mathbf{w} \cdot \phi(\mathbf{x}_i) + b)$, where $\mathbf{w}$ and b are found by solving

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} & y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \text{ for all } i. \end{array}$$

Here, ξ_i are the slack variables, C is the penalization constant of misclassification and $\phi(.)$ is the kernel function. We used the $C = 1$ and the radial basis function (RBF) kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

with $\sigma = 2$ in our experiments.

The classification is done based on majority voting of the individual 31 SVMs.

6) Random forest: RF algorithm has been proposed as an enhancement of Tree Bagging [15]. It uses a CART algorithm without pruning. At each test node, K random features are selected and and a bootstrap replica of the training set is used to find the optimal feature and cut-point.

The classifier with 500 decision tree estimators was used in our experiments. The number of investigated features in each node was set to 3. Entropy was used as the measure of the quality of the split. To prevent over-fitting, the minimum number of samples in the leaf nodes was set to 0.5% of the training set. The classification is done based on majority voting of the individual trees.

7) Extremely randomized trees: ERT are similar to RF, but during induction, the algorithm splits nodes by choosing the best feature with random cut-point and it uses the whole learning sample to grow the trees [15].

The classifier with 100 decision tree estimators was used in our experiments. At each node 7 random features were selected to find the best split. Entropy was used as the measure of the quality of the split. To prevent over-fitting, the minimum number of samples in the leaf nodes was set to 0.5% of the training set. The classification is done based on majority voting of the individual trees.

IV. Evaluation

We evaluate the methods described in Sec. III using our dataset created from high-level representations of the UP-Drive ego-car experience. Here we describe how we preprocess and split our data, the performance measure we use for evaluation, and the results of our experiments.

A. Data preprocessing

As we mentioned in Sec. , each observation in our dataset contains one pedestrian in one particular time frame. In order to better understand the available data, we plot the distribution of the observations over the time of the crossing event. This distribution is depicted in Figure 4. We can see that the majority of crossing will happen in the time span of 3 seconds. Therefore we trained the classifiers on the data where only pedestrians crossing in less than 3 seconds were considered as truly crossing.

To test the previously described methods, we decided to use 3-folded cross-validation. We make use of the fact that our data contain also identities of the pedestrians and of the zebras which they entered and we split the data by these attributes to form the training and testing data folds. This way we make sure that although the data contain each time frame

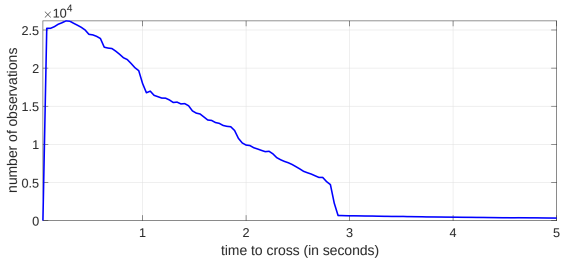

Fig. 4: The distribution of crossing pedestrians over the time to cross.

separately we make a reasonable split and do not mix different trajectories and different pedestrians.

B. Performance measures

The performance of the binary classifiers can be expressed by a confusion matrix containing the number of correctly classified instances (true positives (TP) and true negatives (TN)) and incorrectly classified objects (false positives (FP) and false negatives (FN)). The meaning FP and FN is illustrated in Table I.

TABLE I: Confusion matrix notation

	Not crossing class 0	**Crossing** class 1
Predicted Not crossing class 0	TN	FN
Predicted Crossing class 1	FP	TP

From the confusion matrix, other measures can be derived. We use precision and recall, since recall can tell how well we can detect all relevant objects (crossing pedestrians) and precision is the measure how precise our detection is, i.e. how many of the detected objects are relevant. Mathematically

$$\begin{aligned} \text{Precision} &= \tfrac{\text{TP}}{\text{TP+FP}}, \\ \text{Recall} &= \tfrac{\text{TP}}{\text{TP+FN}}. \end{aligned}$$

By using a Precision/Recall graph we can compare the investigated ML methods. For each method we draw the pair [Recall, Precision] into the graph. This graph is also convenient to compare the performance of the Bayesian network with the binary classifiers. Since the result of the BN is the probability of crossing, we can select a threshold to make the binary classification decision. By sliding this threshold from 0 to 1, we get several [Recall, Precision] couples forming a curve. The area under the curve (AUC) is yet another performance measure we use for evaluation.

The performance of the classifiers can be evaluated by

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}},$$

but we decided to look also at the F_β measure with $\beta = 2$, because F_β measures do not take the true negatives into account (non-crossing predicted correctly)

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}.$$

In our research it is crucial to minimize the false negative rate, which indicates the cases in which the pedestrians were predicted as not having the intention to cross, but in reality they did cross the road.

We used accuracy when evaluating with respect to time to cross, because in this way only crossing pedestrians can be evaluated. When evaluating with respect to distance to zebra and overall evaluation we use the F_2 measure.

C. Results

1) Original dataset: The summary of the performance of all our proposed methods in terms of average F_2 score from the 3-folded cross-validation on the training and the testing sets is displayed in Table II. Resulting confusion matrices for the binary classifiers are displayed in Tables III-VIII. The Precision/Recall graph in Figure 5 displays the performance of our Bayesian network with variable threshold plus the results from the binary classification ML methods experiments. The best F_2 score achieved by the BN was 86.54% at the threshold 0.224529.

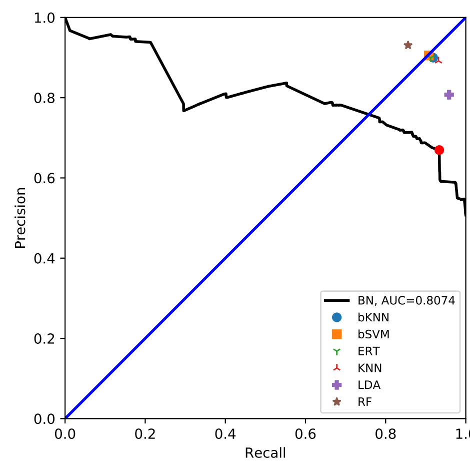

Fig. 5: PR graph. The point of highest F_2 score of the BN is marked by red circle.

In order to see how well the tested methods work in particular time periods, we evaluated the accuracy with respect to time to cross. In Figure 6 we can see that all methods except the random forests achieve over 90% accuracy for pedestrians crossing in less than 1 second. For the time of crossing up to 3 seconds, BN and LDA keep the high accuracy, while the accuracy of other methods is between 80% and 90%. For pedestrians crossing after 3 seconds, the accuracy drops. This corresponds with the fact that the methods were trained on pedestrians crossing in less than to 3 seconds. The accuracy

TABLE II: F_2 score and Accuracy of the tested methods in %

Method	F_2 train	F_2 test	Accuracy train	Accuracy test
K-nearest-neighbors	100 ± 0	92.33 ± 0.012	100 ± 0	**90.79 ± 0.006**
Bagged KNN	97.56 ± 0.001	91.07 ± 0.014	94.75 ± 0.002	90.33 ± 0.007
Linear Discriminant Analysis	92.56 ± 0.002	**92.37 ± 0.009**	86.42 ± 0.001	86.28 ± 0.006
Bagged SVM	94.68 ± 0.003	90.68 ± 0.008	93.04 ± 0.002	90.48 ± 0.004
Random forests	98.37 ± 0.001	87.02 ± 0.015	96.92 ± 0.001	89.49 ± 0.007
Extremely randomized trees	94.97 ± 0.001	90.83 ± 0.012	92.98 ± 0.001	90.27 ± 0.007

TABLE III: Confusion matrix for KNN classification.

KNN	class 0	class 1
class 0	330325	43609
class 1	26276	359176

TABLE IV: Confusion matrix for bKNN classification.

bKNN	class 0	class 1
class 0	333584	40350
class 1	33019	352433

TABLE V: Confusion matrix for LDA classification.

LDA	class 0	class 1
class 0	285833	88101
class 1	16075	369377

TABLE VI: Confusion matrix for bSVM classification.

bSVM	class 0	class 1
class 0	337384	36550
class 1	35750	349702

TABLE VII: Confusion matrix for RF classification.

RF	class 0	class 1
class 0	349430	24504
class 1	55281	330171

TABLE VIII: Confusion matrix for ERT classification.

ERT	class 0	class 1
class 0	334363	39571
class 1	34311	351141

of BN drops to 0 for pedestrians crossing after 3 seconds, which influence the overall accuracy of the method.

Figure 7 show the accuracy as the function of distance to the zebra. All methods achieve over 80% for distances up to 10 meters. BN and keeps accuracy over 95% in this interval.

2) Extended dataset: After the analysis of the previous results we decided to extend the dataset by including an additional feature of pedestrian velocity into the feature vector. The information about the velocity is taken from the preprocessed high-level data abstracted from the ego-car sensors.

As a preliminary study we computed the F_2 measure of the ML methods using the same parameters as with the original dataset. The resulting scores are reported in Table IX and the PR graph is in the Figure 8. We can see that the results of the

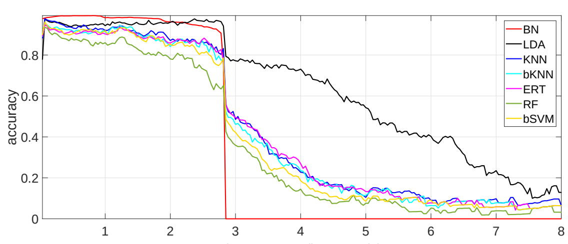

Fig. 6: Accuracy evaluated with respect to the time of crossing.

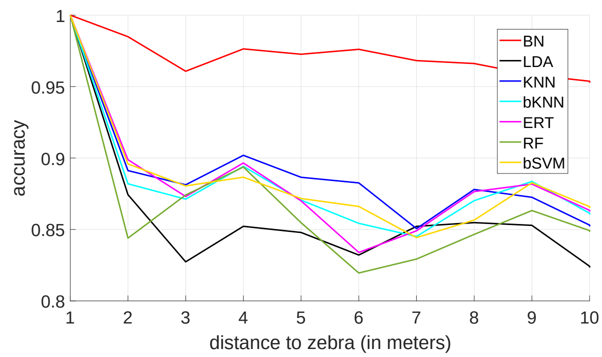

Fig. 7: Accuracy with respect to the zebra distance.

binary classifiers are similar to the previous results, but the AUC of the BN with velocity (0.8536) is bigger than the AUC of the BN with only 8 features. The best F_2 score achieved by the BN was 87.98% at the threshold 0.204965.

TABLE IX: F_2 measure of the ML classifiers on the extended dataset

Method	F_2 train	F_2 test
K-nearest-neighbors	100.00 ± 0.00	**92.33** ± 1.21
Bagged KNN	97.89 ± 0.09	91.17 ± 1.24
Linear Discriminant Analysis	92.25 ± 0.29	92.16 ± 1.21
Bagged SVM	95.55 ± 0.22	89.57 ± 1.12
Random forests	98.69 ± 0.07	85.71 ± 1.78
Extremely randomized trees	94.95 ± 0.23	91.07 ± 1.19

D. Discussion

The results of our Bayesian network approach in [12] were promising, so, after collecting more data we decided to use BN approach to evaluate the method more thoroughly. We decided to use the same features and a simplified structure of the BN. To be able to asses the quality of the intention estimation we also used binary classifiers with the features coming from the same data observations as used by the BN, but in their continuous form without applying the thresholds.

From the results we can see that the Bayesian approach did not fulfill our expectations yet. The AUC of the PR curve is only 0.8074. The highest achieved F_2 was similar to the lowest score of the binary classifiers, but from the Figure 5 we see that the binary classifiers get the Precision and Recall closer to the ideal point [1,1].

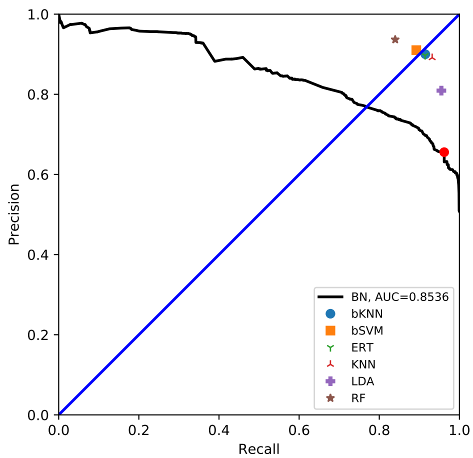

Fig. 8: Evaluation on the extended dataset. PR graph. The point of highest F_2 score of the BN is marked by red circle.

The KNN and bKNN classifiers achieved high scores, but we have to keep in mind, that they are memory based classifiers that need all the training data stored for classification. To classify a feature vector $\mathbf{x}$, the distances to all training vectors must be computed, which is very unfeasible for the autonomous cars application, due to large demands on processing power.

LDA classifier achieved the highest score and the model is compact enough to use it in our application.

In order to compare our approach with the existing work in full, we would need a benchmark dataset with the high-level representation data similar to ours. However, these representations are often bounded to particular manufacturers and there are no standards or benchmarks on the market yet. From the state-of-the-art research described in Sec. II we can make only a weak comparison and conclude that our approach yields slightly better accuracy in general, i.e. 90.79% for the best method (kNN), given 84.8% accuracy achieved in [10] and over 85% accuracy in predicting crossing pedestrians correctly up to 3 seconds and 10 meters to cross (over 90% accuracy for 2 seconds and 3 meters in [11]).

Looking at the extended dataset we conclude that the Bayesian approach is still promising and better feature selection and structure of the BN could lead to a better performance.

V. Conclusion

In our paper we provided a survey of methods capable of solving a special problem of pedestrian intention estimation from trajectory data. Due to the project setup we can utilize only the information related to trajectory of the tracked pedestrians on the map, rather than building our own computer vision methods to process the sensory data. Therefore, we do not have access to information on the body posture or gaze direction of the pedestrians.

Our highest achieved F_2 score was 92.37% (accuracy 86.28%) by the linear discriminant analysis. The highest achieved accuracy was 90.79% (F_2 score 92.33%) by the k nearest neighbors classifier. From the results we can see that a simple Bayesian network approach does not reach the performance of the binary classification machine learning methods in the F_2 score terms, but it has high accuracy when investigating time and distance to cross. Therefore we conclude that the F_2 score can be improved by better BN structure tailoring and feature selection.

Acknowledgment

This work was funded by the European Union H2020 Framework Programme for Research and Innovation under the grant agreement No. 688652, UP-Drive.

References

[1] A. Armand, D. Filliat, and J. Ibañez-Guzmán, "Ontology-based context awareness for driving assistance systems," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 227–233.

[2] V. Gadepally, A. Krishnamurthy, and U. Ozguner, "A framework for estimating driver decisions near intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 637–646, April 2014.

[3] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 69–79, Spring 2015.

[4] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 90–104, March 2016.

[5] S. Bonnin, T. Weisswange, F. Kummert, and J. Schmüdderich, "Pedestrian crossing prediction using multiple context-based models," in *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, 10 2014.

[6] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund, "Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations." in *Intelligent Vehicles Symposium*. IEEE, 2015, pp. 330–335.

[7] Z. Fang and A. M. López, "Is the Pedestrian going to Cross? Answering by 2D Pose Estimation," *CoRR*, vol. abs/1807.10580, 2018.

[8] O. Ghori, R. Mackowiak, M. Bautista, N. Beuter, L. Drumond, F. Diego, and B. Ommer, "Learning to forecast pedestrian intention from pose dynamics," in *Intelligent Vehicles, IEEE, 2018*, 06 2018, pp. 1277–1284.

[9] R. Quintero Mínguez, I. Parra Alonso, D. Fernández-Llorca, and M. Á. Sotelo, "Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1803–1814, May 2019.

[10] S. G. Konrad and F. R. Masson, "Pedestrian intention estimation from egocentric data," in *2017 XVII Workshop on Information Processing and Control (RPIC)*, Sep. 2017, pp. 1–5.

[11] B. Volz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto, "A data-driven approach for pedestrian intention estimation," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 11 2016, pp. 2607–2612.

[12] J. Škovierová, A. Vobecký, M. Uller, R. Škoviera, and V. Hlaváč, "Motion prediction influence on the pedestrian intention estimation near a zebra crossing," in *VEHITS*, 2018.

[13] F. Jensen and T. Nielsen, *Bayesian Networks and Decision Graphs*, 2nd ed. New York, NY: Springer, 2007.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[15] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, Apr 2006.