

Towards Accurate Ego-lane Identification with Early Time Series Classification

Yuchuan Jin, Theodor Stenhammar, David Beijmer, Axel Beauvisage, Yuxuan Xia, and Junsheng Fu
Zenseact, Gothenburg, Sweden

Email: firstname.lastname@zenseact.com

Abstract—Accurate and timely determination of a vehicle’s current lane within a map is a critical task in autonomous driving systems. This paper utilizes an Early Time Series Classification (ETSC) method to achieve precise and rapid ego-lane identification in real-world driving data. The method begins by assessing the similarities between map and lane markings perceived by the vehicle’s camera using *measurement model quality* metrics. These metrics are then fed into a selected ETSC method, comprising a probabilistic classifier and a tailored trigger function, optimized via multi-objective optimization to strike a balance between early prediction and accuracy. Our solution has been evaluated on a comprehensive dataset consisting of 114 hours of real-world traffic data, collected across 5 different countries by our test vehicles. Results show that by leveraging road lane-marking geometry and lane-marking type derived solely from a camera, our solution achieves an impressive accuracy of 99.6%, with an average prediction time of only 0.84 seconds.

I. INTRODUCTION

Localization is an critical module for autonomous vehicles to navigate safely and efficiently in real-world environments. Traditionally, vehicle localization has relied on the fusion of sensor data of diverse types, such as visual, inertial, or Global Navigation Satellite System (GNSS) inputs, to yield pose estimations. Over the past decades, a range of algorithms, including e.g., INS/GNSS navigation [1]–[3], Visual Odometry (VO) [4], [5], Simultaneous Localization and Mapping (SLAM) [6], [7], and Visual-Inertial Navigation Systems (VINS) [8], have been developed to address the challenges in localization. Nevertheless, these techniques may encounter challenges when applied to large datasets. For instance, Visual Odometry (VO) and SLAM algorithms often exhibit drift over time in the absence of loop closure strategies [8]. Additionally, GNSS can have substantial biases in urban environments, leading to significant degradation in localization accuracy [9]. In recent years, the adoption of High Definition (HD) maps has increased in popularity, primarily due to their ability to offer support for precise localization [9]–[12]. An HD map in our context is a semantic map that contains lane-level information such as lane markings, barriers, and traffic signs with positioning and types information [13], [14].

Identifying the correct ego-lane within an HD map by leveraging vehicle onboard sensors is a crucial task during the initialization phase of HD map-based localization. In our context, the initialization phase spans from the moment the automated driving system receives sensor measurements and map data to the point at which the system selects an ego-lane for the ego vehicle. Initialization (or re-initialization) can occur

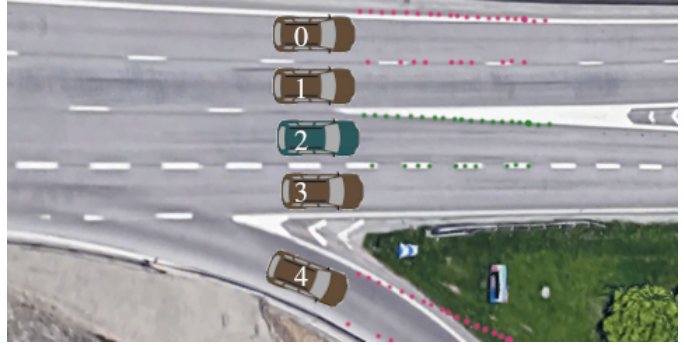


Fig. 1. An illustration depicts perceived lane markings from various hypotheses’ viewpoints. The red dots represent perceived lane markers from Hypotheses 0 and 4, which do not align well with the map. In contrast, the green dots depict lane markers perceived from Hypothesis 2 (the correct one), showing a more accurate alignment with the map.

for various practical reasons and at different locations, such as when sensor measurements are lost for an extended period, navigating through areas with low-quality mapping (e.g., road work), or traversing unmapped regions. Ego-lane identification poses particular challenges in areas with numerous lanes or even multiple road layers as GNSS could fail to provide sufficient accuracy to identify the specific lane the vehicle occupies [15].

Road lane markings are ubiquitous features on most roads, and HD maps meticulously capture both their geometry and types. Additionally, lane markings can be quite easily detected by a vehicle’s camera. By utilizing GNSS measurements as the rough estimate of the vehicle’s position and leveraging the perceived geometry and types of lane-markings captured by the vehicle’s camera, it is possible to identify the current ego-lane within an HD map. This paper focuses specifically on ego-lane identification during the initialization phase by using the road lane-marking geometry and types.

Several techniques based on multi-hypothesis tracking have been proposed to address the ego-lane identification problem [16]–[18]. These approaches initially generate hypotheses in various lanes. Subsequently, they jointly track the location of these hypotheses across different lanes by incorporating sensor measurements, aiming to determine which one accurately represents the vehicle’s position. Fig. 1 illustrates the process of generating multiple hypotheses along different lanes on a highway. The automated driving system then

utilizes the camera-perceived road lane-marking geometry and types to identify the correct ego-lane in an HD map. In this scenario, hypothesis 2 is determined to be the correct one. Accomplishing this task presents a significant challenge due to the necessity of identifying the correct hypothesis with a high level of confidence, and this confidence naturally grows with the accumulation of more observations. However, it should also avoid unnecessary delays. Therefore, striking a balance between accuracy and promptness is essential, a task commonly referred to as the Early Time Series Classification (ETSC) problem.

In this paper, we utilize an effective ETSC solution for ego-lane identification in HD map with lane-marking geometry and lane-marking types. For camera perceived lane-marking geometry and lane-marking types, we compute the similarity metric called *measurement model quality* (MMQ, refer details to subsection III-B) by comparing perceived lane-marking's geometry and types against the HD map. These metrics are then inputted into a selected ETSC solution, which comprises a probabilistic classifier and a tailored trigger function. This solution is then further optimized through multi-objective optimization to strike a balance between early prediction and accuracy.

The main contributions of this paper are:

- 1) We conducted an evaluation and adaptation of the ETSC method for the ego-lane identification task. The results indicate that our modified approach shows improvements in terms of promptness and availability.
- 2) Our experiments utilized 114 hours of real-world driving data to assess the efficacy of the ETSC solution in the domain of ego-lane identification, confirming its robust performance.

The rest of the paper is organized as follows. In Section II, we summarize existing work in multiple-hypothesis tracking and the ETSC problem. Section III outlines the entire methodology we employ to address the multiple-hypothesis ETSC problem. Our experimental procedures and their results are detailed in Section IV, while Section V delves into the discussions and conclusions drawn from our findings.

II. RELATED WORK

Multiple-hypothesis tracking is a classic method for tracking multiple targets by propagating multiple data association hypotheses over time [16]. The idea of maintaining multiple hypotheses at a time has been adapted to the global localization problem (also commonly known as the *kidnapped robot* problem) to localize an autonomous agent in its environment without prior knowledge [17]. In addition, the use of digital maps can be incorporated into the framework for automotive navigation purposes [18], [19].

The multi-hypothesis localization problem can be formulated as an ETSC task since the data is continuously collected by the vehicle and evaluated regularly with the goal to reduce the uncertainties captured by multiple hypotheses as early as possible. Then, similar to ETSC [20], machine learning

techniques can be leveraged in solving multi-hypothesis localization. In machine learning, classification of time series data differs from classification of tabular data in the sense that sequential information are being processed rather than individual snapshots [21]. Therefore, it is important to employ the right machine learning method that fits the type of data in our specific application [22].

A practical issue with building an ETSC model is that the time series data used as model input are of different lengths. This often requires a different approach to classification than offline time series classification. The reason for this is that in ETSC the full time series is not available. Some methods for time series classification, such as the Multi-Layer Perceptron proposed in [23], depend on the length of time series. A method that is independent of the length of the time series can be used any time new data is collected, but a method that is dependent on the length can only be used for the specific time series length that it is constructed for. The application of such length-dependent methods for ETSC requires first defining a set of lengths, at which time series are evaluated, and then constructing a different classifier for each possible length of the time series data.

To overcome the limitation of length-dependent methods, a two-tiered structure has been proposed in [24], [25], which first divides a time series into segments of different lengths and then uses a classifier trained to output class probabilities for the different lengths of time series segments. Depending on the prediction certainty, a reject-classifier either chooses to output a prediction or wait for more data. Recently, it has been observed in [26] that methods for ETSC problems are sensibly adapted in the Early Open Time Series Classification problem (EOTSC) and the transformation methods of ETSC for EOTSC problem have been illustrated. However, there is still ongoing investigation regarding the trade-off among accuracy, earliness, and availability. In the context of localization tasks, both earliness and availability are crucial factors for timely performance. Therefore, it is imperative to conduct a comprehensive study on EOTSC methods, particularly from the perspectives of earliness and availability.

III. METHOD

In this section, we describe our Swift Multi-objective Early Open Time Series Classification method (SMEOTSC), which can be considered as an extension of the previous methods in [24] and [26], named Multi-objective ETSC (METSC) and Multi-objective EOTSC (MEOTSC), respectively. Our method is tailored to the problem ego-lane identification within an HD map.

A. Hypothesis generation

To perform ego-lane identification, we first employ a multiple-hypothesis tracking method. We dynamically generate hypotheses, with a maximum limit of K , placing one hypothesis in each lane on the HD map. This process progresses from the nearest to the farthest lane, determined by

the distance between the latest GNSS measurement and the center of each lane.

B. Measurement model quality generation

For each hypothesis, we use a Kalman filter to track its in-lane position (x, y, yaw) by utilizing the detected left and right lane-marking of the ego-lane from the vehicle camera. At each subsequent time step, and for each hypothesis, we gather various measurements such as lane marking sample points (indicating geometry) and their types (e.g., solid, dashed, doubled). These measurements are then associated with the HD map segments surrounding the latest GNSS measurement. The measurement model defines the relationship between the state of the hypothesis and the lane-markings measurements.

To assess the likelihood of the different hypotheses and to discern the true one from the others, it is necessary to translate the measurements acquired by the sensors into matching scores. We name such metric *measurement model quality* (MMQ) since they are derived from the measurement model. The MMQs are computed for each sensor measurement, based on the Normalized Innovation Squared (NIS) and the chi-square χ^2 distribution. The NIS value is computed as

$$\text{NIS} = (z_t - \hat{z}_t)^T S^{-1} (z_t - \hat{z}_t), \quad (1)$$

where z_t , \hat{z}_t denote the observed measurement and the predicted measurement at time step t respectively; S denotes the innovation covariance matrix computed in the Kalman update; $z_t - \hat{z}_t$ corresponds to the innovation term (also known as residual).

Though each type of measurement possesses its own measurement model and produces its own innovation term, the fact that it is normalized by its associated covariance matrix makes it possible to treat all measurements in the same way. Therefore, the MMQ is a scalar value defined as:

$$\text{MMQ} = -\log \frac{\text{NIS} + \text{penalty}}{\text{chi2inv}(p, d)} \quad (2)$$

where $\text{chi2inv}(p, d)$ denotes the inverse cumulative distribution function of χ^2 distribution with d degrees of freedom and evaluated at probability p ; penalty represents a threshold that depends on the number of outlier measurements.

The motivation for this penalty term is to penalize the case when no association can be made between the measurement and the map element or when their distances are larger than some threshold. As a result, the association cost is normalized based on the number of samples that were used, following a χ^2 distribution, where the degrees of freedom of the χ^2 distribution corresponds to the number of samples.

The MMQs are treated as a matching cost, and they reflect how well the perceived measurement reflects the system expectation from its estimated state. The χ^2 normalization ensures that all MMQs have similar magnitudes, regardless of their number of samples. This way, it is easier to combine model qualities and compare hypotheses. For example, a traffic sign position should produce a similar MMQ value to a lane marking represented by 10 points if their innovation term

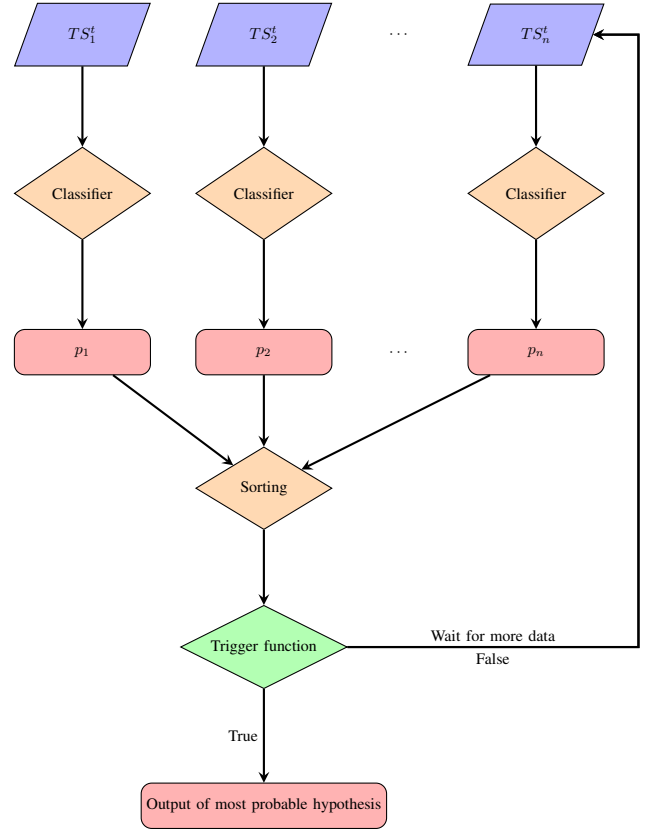


Fig. 2. An overview of the ETSC architecture, based on the process described in [24]. TS_n^t is the n :th Time Series (TS) at time t ; p_n is the predicted true-probability given by classifier for hypothesis n ; Trigger function is in charge of multi-hypotheses selection.

is equally consistent with their uncertainty (variance). These MMQs values are used as derived features in the subsequent learning algorithms.

C. Learning-based hypothesis inference

The full architecture of the hypothesis inference solution is shown in Fig. 2. The input to the model is a set of time series $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t)$ from multiple hypotheses. At each time step t , $\mathbf{x}_t = (x_t^1, x_t^2, \dots, x_t^m)$ is represented as a vector consisting of MMQ values discussed in Section III-B. The final output is the index of the hypothesis with the highest probability of being correct. The solution can be divided into the following two steps.

First, the multivariate time series data is fed to a set of probabilistic classifiers, which outputs the confidence of hypothesis n at a given time t . The classifiers are trained with labeled multivariate time series set \mathbf{X} . Second, the certainty of making a selection is determined by comparing the confidence levels of different hypotheses. The hypotheses' confidences are sorted by magnitude and provided to a trigger function s_γ , which determines the appropriate action. It decides whether to make a selection at this moment or if the model requires additional data (time steps) to confidently identify the most probable hypothesis indicated by the classifiers.

The trigger function's role is to assess whether the system can confidently generate a prediction at a given time t based on the probabilities of provided hypotheses, or if additional data is needed. As illustrated in Fig.2, the trigger function outputs either True or False based on the classifier's result. If the trigger function determines the prediction to be reliable (outputting True), then the hypothesis with the highest probability will be selected. Conversely, if the trigger function decides that the prediction is not yet reliable (outputting False), further data is then needed before making a reliable prediction.

The trigger function proposed in [24] is shown as follow:

$$s_{\gamma}^1(\mathbf{p}, t) = \begin{cases} 0, & \text{if } \gamma_1 p' + \gamma_2(p' - p'') + \gamma_3 \frac{t}{T} \leq 0 \\ 1, & \text{otherwise.} \end{cases} \quad (3)$$

where $\mathbf{p} = (p_1, p_2, \dots, p_n)$ is a vector of probabilities output from classifiers for each hypotheses; p' and p'' are, respectively, the highest and the second-highest hypothesis probabilities obtained at time step t ; $\gamma = (\gamma_1, \gamma_2, \gamma_3)$ is a vector of parameters with $\gamma_i \in [-1, 1]$; t is current time step; T is the total time steps of time series input.

The approach from [24], named METSC, is aimed to solve the multi-class classification problem, where a time series is classified after some number of observations of a time series of indeterminate length. However, as discussed in Section II, this method is constrained in the context of EOTSC problems due to the absence of a total time series duration for such problems. Consequently, [26] has adapted the trigger function (3) specifically for the EOTSC domain:

$$s_{\gamma}^2(\mathbf{p}, t) = \begin{cases} 0, & \text{if } \gamma_1 p' + \gamma_2 p'' + \gamma_3 \frac{\eta - t}{\eta} \leq 0 \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

where η is predefined total time steps for the time series input.

As can be seen, trigger function (4) employs a predetermined total time steps for EOTSC problems and its value decreases over time. However, the question persists: How should we define the total time steps in advance? Firstly, determining an appropriate total number of time steps for various types of tasks is challenging, resulting in adding another parameter to be learned, which complicates the learning process. Secondly, this approach retains the time penalty element, which might actually impede the algorithm's ability to determine the reliability of predictions at an early stage. To address these challenges and to tailor to our specific ego-lane identification problem, we adopt trigger function (4) as follows:

$$s_{\gamma}^3(\mathbf{p}) = \begin{cases} 0, & \text{if } \gamma_1 p' + \gamma_2(p' - p'') + \gamma_3 \leq 0 \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

$$s_{\gamma'}^4(\mathbf{p}) = \begin{cases} 0, & \text{if } \gamma_1 p' + \gamma_2(p' - p'') \leq 0 \\ 1, & \text{otherwise} \end{cases} \quad (6)$$

where $\gamma' = (\gamma_1, \gamma_2)$ is a vector of parameters with $\gamma_i \in [-1, 1]$

In trigger function (5), γ_3 is kept to ensure consistency in parameter numbers with (3) and (4) for comparative purpose.

Additionally, trigger function (6) also removes parameter γ_3 to let algorithm give a prediction as early as possible when the classifier results are reliable enough. For the lane-level localization application, it is desired that a prediction is made once the information state is good enough to determine the ego-lane index with certitude. The solution should therefore be reactive to fluctuations in the information state and output as quickly as possible once a selection is feasible and reliable. However, it should exercise restraint in making a selection in instances where different hypotheses are indistinguishable. Detailed analysis related to trigger function (5) and (6) will be presented in Section IV-C.

D. Multi-objective optimization

Three metrics can be utilized to evaluate the performance of the method: availability, earliness, and accuracy. Their definitions are as follows:

Definition 1. Availability is defined as the fraction of sequences for which a prediction was made within the observation period. In our experiments, we establish a maximum allowable time for generating a prediction. If the system exceeds this time limit without producing a prediction, we consider it a failure in our assessment. Then, the availability can be calculated by dividing the number of successful predictions by the total number of sequences.

Definition 2. Earliness is defined as the average number of observations needed before a prediction was made, normalized by second.

Definition 3. Accuracy is defined as the fraction of correct predictions for all sequences where a prediction is available.

To obtain a robust estimation of the parameter vector γ , a multi-objective function targeting availability and accuracy is optimized in a similar manner as in [24]. The motivation behind optimizing on availability instead of earliness is that the availability of a prediction is more important in automated driving scenarios. Consider a scenario where a driver employs an Advanced Driver Assistance System (ADAS). Typically, the initialization phase occurs automatically when the journey begins. It's common for drivers to cover some distance before utilizing the ADAS features, thus surpassing the initialization phase. Indeed, in cases where no prediction can be made with high confidence, the model should not be forced to predict based on an earliness factor.

The accuracy cost is defined as the prediction error, which is the percentage of incorrectly predicted time series. The earliness cost is defined as the ratio of the time step at which the prediction is made to the total time steps of the time series. The availability cost is defined as the non-prediction rate, indicating the percentage of time series not predicted within the total length of the time series. Their expressions are given as follows:

$$C_{ac}(\mathbf{X}, s_{\gamma}) = \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x} \in \mathbf{X}} \iota(\hat{y}_{\mathbf{x}} \neq y_{\mathbf{x}}), \quad (7)$$

$$C_{ea}(\mathbf{X}, s_\gamma) = \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x} \in \mathbf{X}} \frac{t_x^*}{L_x}, \quad (8)$$

$$C_{av}(\mathbf{X}, s_\gamma) = \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x} \in \mathbf{X}} \iota(\# \hat{y}_x), \quad (9)$$

where $\iota()$ is a boolean function, taking a value of 1 if the condition is true and 0 otherwise; y_x and \hat{y}_x are the ground truth value and prediction value, respectively; t_x^* is the time step when prediction is made for time series \mathbf{x} ; and L_x is the total number of time step for time series \mathbf{X} .

The explained cost function regarding availability is similar to the earliness cost function by [24]. The availability cost can be seen as a discrete earliness cost, which only outputs a positive value if no prediction has been made during the sequence. The earliness cost is more continuous in the sense that it returns a growing value from the first time step.

The optimization problem is defined as follows:

$$\min_{\gamma} (C_{av}(\mathbf{X}, s_\gamma), C_{ac}(\mathbf{X}, s_\gamma)) \quad (10)$$

The objective of this multi-objective optimization problem is to minimize both the availability cost and the accuracy cost. Following the approach outlined in [24], we apply meta-heuristic algorithms to address this optimization problem due to its complexity involving multiple objectives.

IV. EXPERIMENTS

In this section, we present and analysis the experimental results on real-world data.

A. Data

We gathered 114 hours of experimental data from across 5 countries: Germany, France, Belgium, Sweden, and the United States. The test vehicle has the same setup as the one in our Zenseact Open Dataset [27]. Table I shows the statistics regarding the number of sequences and driving hours for each country. This dataset encompasses diverse measurements, but our experiments have mainly used GNSS and front-looking camera data. The GNSS is employed solely to provide a rough location estimate of the vehicle, facilitating the retrieval of surrounding HD map data. Subsequently, the ego-lane identification only relies on camera data and the retrieved HD map data. The driving scenarios recorded were primarily situated in multi-lane highway settings because the HD map coverage is limited in Highway in our dataset. Visual data, if discernible, facilitated the identification of lane-markings pertinent to both the ego-lane and its adjacent lanes. The dataset further delineated the geometrical characteristics of these lanes, and the quality of the measurement models was evaluated as explained in Section III-B.

The inputs for ETSC comprise 8 measurement model qualities. 4 of them focus on the geometric properties of both the ego-lane and its adjacent lanes, while the remaining 4 characterize the types of lane markings within the ego-lane. All these measurements are consistently computed at a frequency of 40Hz:

- ego-lane left lane-marking geometry

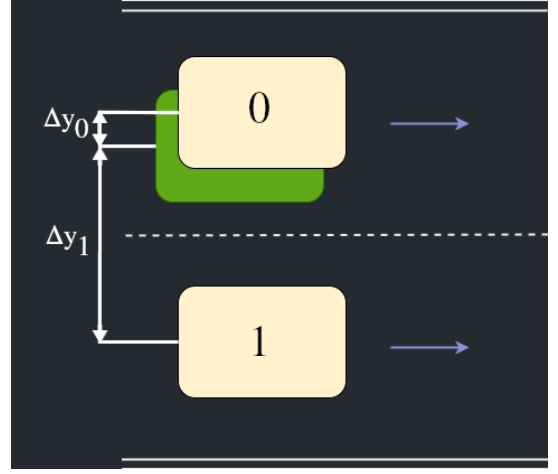


Fig. 3. Visualisation of the annotation scheme with the green box being the reference (ground truth). Δy is the difference between the center of ground truth and hypothesis. Hypothesis 0 is labeled as true, and hypothesis 1 is labeled as false.

- ego-lane right lane-marking geometry
- adjacent-lane left lane-marking geometry
- adjacent-lane right lane-marking geometry
- ego-lane left lane-marking type
- ego-lane right lane-marking type
- adjacent-lane left lane-marking type
- adjacent-lane right lane-marking type

For any given time step or attribute where data is absent, the measurements are considered missing. Such gaps in data might arise from inconsistencies in the map, or perception discrepancies related to lane markers.

To train a supervised machine learning model, the availability of ground truth labels is crucial. These ground truth labels are derived from a reference model, which is generated offline by utilizing data acquired from high-precision external sensors during the data collection phase. The sensor setup is same as [27], which comprises a set of GNSS/INS positions collected by an Oxts RT3000 unit and this ground truth pose is further refined by aligning Lidar scans with HD map with an ICP [28]. The Velodyne sensors are a top-mounted VLS-128 Lidar and two side-mounted VLP-16 Lidars. The experiments are conducted on a laptop with an Intel(R) i7-1365U processor (1.80 GHz), 10 cores, and 32 GB of RAM.

Fig. 3 illustrates the annotation scheme. The time series are categorized as either *true* or *false* based on their lateral proximity to the reference pose, as established by the reference system. A hypothesis is designated as *true* only if it is the singular hypothesis closest to the reference pose, with all other hypotheses consequently labeled as *false*. A complete time series is deemed valid if every time step for a particular hypothesis is labeled as *true*. Consequently, sequences exhibiting ambiguity in their ground truth are excluded. Such ambiguities may arise from discrepancies in the reference pose or misalignment in the map.

TABLE I
NUMBER OF SEQUENCES AND DRIVING HOURS COLLECTED IN 5 COUNTRIES

| Country | Number of Sequences | Driving Time (Hours) |
|---------------|---------------------|----------------------|
| Germany | 423 | 17.4 |
| France | 656 | 26.6 |
| Belgium | 328 | 13.4 |
| Sweden | 768 | 32.2 |
| United States | 581 | 24.1 |

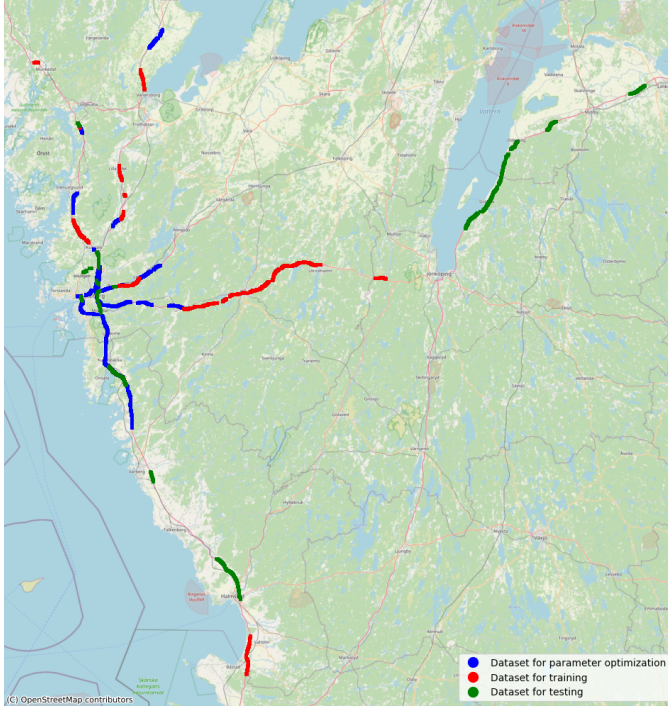


Fig. 4. Visualization of the 32-hour driving logs collected around Gothenburg, Sweden. The dataset was divided into 3 groups, red, blue, and green, which have been used for training, parameter optimization, and testing respectively. The map is drawn with openstreet basemap [29].

B. Experiments Setup

As mentioned in Section III-C and III-D, a set of classifiers need to be trained for estimating the probability of different hypotheses at different time and a meta-heuristic optimization algorithm is needed to solve the multi-objective optimization problem. Analyzing the performance of different classifiers and meta-heuristic optimization algorithm to solve this problem is beyond the scope of this paper. In order to perform our experiments, we have chosen gradient boosting algorithm [26] as classifier training algorithm and NSGA II optimization algorithm [24] as multi-objective optimization algorithm.

Gradient Boosting is a powerful boosting algorithm in machine learning used for classification, which combines several weak learners into strong learners. The algorithm has been implemented in Python using the *sklearn* package and the default parameters are used. NSGA II is a fast and elitist genetic algorithm which is established in multi-objective optimization scenarios. The algorithm has been implemented in Python using the *pymoo* package. Except for a parameter

change in population size to 8 and termination step of 16, the default parameters are used. The mentioned adjustments to the default parameters have been made to make the algorithm run faster.

The dataset was divided into three subsets for training the classifier, performing the multi-objective optimization, and evaluating the results. Feature extraction for providing the probability estimates was performed by averaging the features over a set of windows with different sizes. There are 2756 sequences and around 114 hours of driving in total. Each sequence has a maximum duration of 30 seconds and contains up to 6 hypotheses. These hypotheses will be generated for each lane on the HD map (up to 6), advancing from the nearest to the farthest lane, determined by the distance between the previous GNSS measurement and the lane center of each lane. Not all hypotheses are present for the full 30 seconds as they may have been deactivated early. An early deactivation of an hypothesis can be due to the hypothesis running out of the map, which happens when there is no map available for the road. The evaluation was performed using the earliness, accuracy, availability metrics as described in Section III-D. The other performance evaluation metric is hypervolume [30], which is an indicator used in multi-objective optimization as both a proximity and diversity performance metric. Given that the optimal scenario arises when both availability and accuracy achieve a value of 1, we will use the reference point (1,1) to assess performance. In this context, a higher hypervolume signifies superior model performance.

Each driving sequence, lasting up to 30 seconds and comprising numerous coordinates, was represented by the average of these coordinates to denote the geographical center of the sequence. Subsequently, the data were categorized by country and weather conditions, followed by the application of a K-means clustering algorithm to create three clusters within each group. Finally, data sharing the same label across different groups were consolidated into a singular dataset. Fig. 4 shows an example of dataset split in Sweden. The reason for this split as opposed to a conventional random split was to avoid evaluating on road segments that occur in the training set, which may be the case with a random split since the dataset contains multiple sequences of the same route. Also, the weather condition for different drives should be relatively evenly distributed in all datasets. Based on this split, dataset for training probabilistic classifier has 1343 sequences (55-hour driving); dataset for optimizing trigger function parameters has 930 sequences (38-hour driving); and the dataset for evaluating has 497 sequences (21-hour driving).

C. Result and Analysis

In Fig. 5, the significance of using the trigger function is emphasized by contrasting its performance with the accuracy obtained when predictions are made at each time step. SMEOTSC2 is developed based on [24] but the trigger function is replaced with Equation (6). The red dot indicates the average prediction time and the corresponding average accuracy of that decision made by SMEOTSC2. The blue

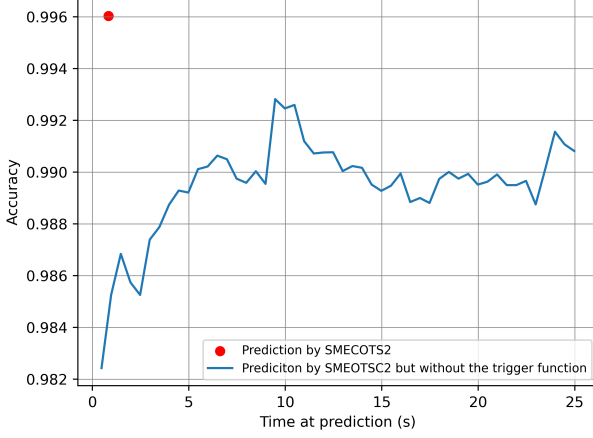


Fig. 5. Compare the prediction performance of our best-performing ETSC method, SMEOTSC2, in all driving data with and without a trigger function. Without a trigger function, the method makes independent predictions at each time step. However, when incorporating the trigger function, it achieves an accuracy of 99.6% with an average prediction time of 0.84 seconds.

TABLE II
PERFORMANCE METRICS ACROSS FOUR ASPECTS FOR EACH METHOD.
THE METRICS, FROM LEFT TO RIGHT, INCLUDE EARLINESS,
AVAILABILITY, ACCURACY, AND HYPERVOLUME, RESPECTIVELY.

| Methods | Ear.(s)↓ | Ava.↑ | Acc.↑ | HV↑ |
|-------------|-------------|---------------|---------------|---------------|
| METSC [24] | 3.89 | 0.9286 | 0.9960 | 0.6565 |
| MEOTSC [26] | 1.53 | 0.9663 | 0.9940 | 0.7711 |
| SMEOTSC1 | 2.44 | 0.9405 | 0.9940 | 0.7562 |
| SMEOTSC2 | 0.84 | 0.9881 | 0.9960 | 0.8405 |

line represents the accuracy obtained by always selecting the hypothesis with the highest probability without activating the trigger function. It shows that activating trigger function not only yield higher accuracy compared to always selecting the hypothesis with the highest probability, but also facilitate earlier decisions with greater confidence.

With the dataset being used, Fig. 5 points to a correlation between the duration of the prediction time and the probability of achieving an accurate decision. The analysis reveals diminishing returns associated with extended waiting times before making a prediction. While a brief delay enhances prediction accuracy, the advantages diminish as the waiting period extends. Consequently, these findings underscore the potential for striking a balance between early decision and accurate decision in lane-level localization problems.

Table II presents the four performance metrics for each method. Both SMEOTSC1 and SMEOTSC2 are developed based on [24] but the trigger functions are replaced with Equation (5) and Equation (6) respectively. Notably, our adapted SMEOTSC2 method demonstrates superior performance than the three other listed methods. It has the earliest prediction time (0.84 seconds), the highest availability (0.9881), and the highest accuracy (0.9960). Hypervolume (0.8405) also shows the highest results corresponding to each trigger function,

serving as an evaluation metric for their performance. It's remarkable that our method can deliver predictions in under one second on average and achieve an availability of 0.9881, indicating it can accurately identify the ego-lane in 98.81% of scenarios without compromising accuracy when compared to other methodologies.

As discussed in Section III-C, the incorporation of a time element in the trigger functions (3) and (4) from METSC and MEOTSC has been identified as a limiting factor for early prediction capabilities. MEOTSC, in its adaptation to the open time series challenge, attempts to mitigate this by defining a total time step upfront and decreasing the time value as time progresses. Although this modification shows an improvement over its predecessor, it still restricts the algorithm's potential for making timely predictions. SMEOTSC1 is based on the trigger function described in Equation (5), and it retains the parameter γ_3 , which is found to further hinder early-stage prediction abilities when compared to MECOTS. This suggests that maintaining a constant parameter does not enhance the algorithm's performance in the context of open time series problems, likely because a constant parameter is challenging to adapt to the time series context and struggles to accommodate temporal variations. Availability appears to be linked to earliness; SMECOTS2 not only delivers predictions in the shortest time step but also exhibits the highest availability, with MECOTS following closely in both aspects. The accuracy of the algorithms does not significantly differ, underscoring that precision in predicting the ego-line is primarily influenced by parameters γ_1 , γ_2 , p' , and p'' , with γ_3 and the time step affecting only the earliness and availability of predictions. In our experiments, the hypervolume indicator, which measures the proximity of optimization objectives to the reference point (set at (1, 1)), highlights that since accuracy remains consistent across the board, the algorithm boasting the highest availability (SMECOTS2) achieves the superior hypervolume.

V. CONCLUSIONS

In this paper, we employ Early Time Series Classification to tackle the ego-lane identification task in autonomous driving, leveraging the *measurement model quality* (MMQ) metrics to represent similarities between the map and the lane markings perceived by the vehicle camera. This approach effectively balances prediction time and accuracy, ensuring prompt and confident decision-making. Our experiments in real-world traffic driving data reveal that utilizing only road lane marking geometry and lane type information, our method achieves an impressive accuracy of 99.6% and availability of 98.81% with an average selection time of only 0.84 seconds.

In the future, this work could be extended by incorporating additional MMQ. For instance, MMQ calculated from the position of tracked traffic signs or the position of road edges which could provide further insights into the position of the ego-lane. Similarly, tracked surrounding vehicles could offer valuable cues about the lanes they occupy, enhancing ego lane assignment algorithms.

REFERENCES

- [1] A. Almagbile, J. Wang, and W. Ding, "Evaluating the Performances of Adaptive Kalman Filter Methods in GPS/INS Integration," *Journal of Global Positioning Systems*, vol. 9, no. 1, pp. 33–40, 6 2010.
- [2] K.-W. Chiang, G.-J. Tsai, H.-J. Chu, and N. El-Sheimy, "Performance enhancement of INS/GNSS/refreshed-SLAM integration for acceptable lane-level navigation accuracy," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2463–2476, 2020.
- [3] Y. He, J. Li, and J. Liu, "Research on GNSS INS & GNSS/INS integrated navigation method for autonomous vehicles: A survey," *IEEE Access*, 2023.
- [4] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, 2004, pp. 652–659.
- [5] H. Zhan, C. S. Weerasekera, J.-W. Bian, and I. Reid, "Visual odometry revisited: What should be learnt?" in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 4203–4210.
- [6] V. Ila, L. Polok, M. Solony, and P. Svoboda, "Highly Efficient Compact Pose SLAM with SLAM++," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 210–230, 8 2016.
- [7] S. Zheng, J. Wang, C. Rizos, W. Ding, and A. El-Mowafy, "Simultaneous localization and mapping (SLAM) for autonomous driving: Concept and analysis," *Remote Sensing*, vol. 15, no. 4, p. 1156, 2023.
- [8] G. Huang, "Visual-Inertial Navigation: A Concise Review," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 5 2019, pp. 9572–9582.
- [9] H. Cai, Z. Hu, G. Huang, D. Zhu, and X. Su, "Integration of GPS, Monocular Vision, and High Definition (HD) Map for Accurate Vehicle Localization," *Sensors*, vol. 18, no. 10, p. 3270, 9 2018.
- [10] R. Liu, J. Wang, and B. Zhang, "High Definition Map for Automated Driving: Overview and Analysis," *The Journal of Navigation*, vol. 73, no. 2, pp. 324–341, 3 2020.
- [11] K. Jo, Y. Jo, J. K. Suhr, H. G. Jung, M. Sunwoo, F. Li, P. Bonnifait, J. Ibanez-guzman, J. Ibañez-guzmán, F. Li, P. Bonnifait, J. I.-g. Using, H. Definition, E. Gnss, F. Li, P. Bonnifait, J. Ibanez-guzman, I. L. Cardenas, J. Du, and M. J. Barth, "Using High Definition Maps to Estimate GNSS Positioning Uncertainty," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 6, 2017.
- [12] F. Li, P. Bonnifait, J. Ibanez-Guzman, and C. Zinoune, "Lane-level map-matching with integrity on high-definition maps," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 6 2017, pp. 1176–1181.
- [13] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nusenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.
- [14] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.
- [15] J. Laconte, A. Kasmi, R. Aufrère, M. Vaidis, and R. Chapuis, "A survey of localization methods for autonomous vehicles in highway scenarios," *Sensors*, vol. 22, no. 1, p. 247, 2021.
- [16] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 12 1979.
- [17] P. Jensfelt and S. Kristensen, "Active global localization for a mobile robot using multiple hypothesis tracking," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 748–760, 2001.
- [18] F. Abdallah, G. Nassreddine, and T. Denoeux, "A Multiple-Hypothesis Map-Matching Method Suitable for Weighted and Box-Shaped State Estimation for Localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1495–1510, 12 2011.
- [19] Jong-Sun Pyo, Dong-Ho Shin, and Tae-Kyung Sung, "Development of a map matching method using the multiple hypothesis technique," in *Proceedings in Intelligent Transportation Systems*. IEEE, 2001, pp. 23–27.
- [20] T. Santos and R. Kern, "A literature survey of early time series classification and deep learning," in *CEUR Workshop Proceedings*, vol. 1793, 2017.
- [21] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 5 2017.
- [22] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 35, no. 2, pp. 401–449, 3 2021.
- [23] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 5 2017, pp. 1578–1585.
- [24] U. Mori, A. Mendiburu, I. M. Miranda, and J. A. Lozano, "Early classification of time series using multi-objective optimization techniques," *Information Sciences*, vol. 492, 2019.
- [25] P. Schäfer and U. Leser, "TEASER: early and accurate time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1336–1362, 9 2020.
- [26] Y. Achenchabe, A. Bondu, C. Antoine, and L. Vincent, "When to classify events in open times series?" in *Asian Conference on Machine Learning*. PMLR, 2023, pp. 1–16.
- [27] M. Alibeigi, W. Ljungbergh, A. Tonderski, G. Hess, A. Lilja, C. Lindström, D. Motorniuk, J. Fu, J. Widahl, and C. Petersson, "Zenseact open dataset: A large-scale and diverse multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 178–20 188.
- [28] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.
- [29] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.
- [30] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International conference on parallel problem solving from nature*. Springer, 2004, pp. 832–842.