Reproducibility and Progress in Estimating Time of Arrival, or Can Simple Methods Still Outperform Deep Learning Ones?

Rami Al-Naim
ITMO University
ScPA "StarLine", Ltd.
Saint Petersburg, Russia
rami.naim2010@yandex.ru

Petr Chunaev
ITMO University
Saint Petersburg, Russia
chunaev@itmo.ru

Klavdiya Bochenina
ITMO University
Saint Petersburg, Russia
bochenina@itmo.ru

ABSTRACT

Methods providing the Estimated Time of Arrival (ETA) of a car have wide applications in trip planning and time management. Considering the complexity of a modern city, ETA prediction is a challenging task that is performed nowadays by more and more complex solutions such as Deep Learning (neural) techniques, often with the claim of a substantial progress in ETA prediction quality. Nevertheless, as in the area of other data mining tasks, indications exist of certain reprodicibility problems in today's research practice connected with the choice of state-of-the-art ETA prediction methods. The purpose of our study is to shed light on the problems via an overview of studies proposing new ETA prediction methods and the related reproducibility issues (including the availability of open source code and datasets, especially of GPS trajectory data). Furthermore, motivated by the recent observations in the field of recommender systems that the majority of existing neural approaches can be outperformed by traditional simple methods, we perform an experimental study that surprisingly shows that a fine-tuned combination of simple regression-based ETA prediction methods (we call it *Strat-mETA*) can indeed outperform more complex solutions (including Deep Learning-based) by means of a multi-component quality metric. We perform the study on a new real-world car travel dataset of GPS trajectory data and make its part public as a benchmark in order to encourage future research and partly resolve the problem of reproducibility in the field.

CCS CONCEPTS

• **Theory of computation → Genetic programming**; **Machine learning theory**; • **Mathematics of computing → Evolutionary algorithms**.

KEYWORDS

estimated time of arrival, reproducibility, car travel data, deep learning, multi-objective optimization

ACM Reference Format:
Rami Al-Naim, Petr Chunaev, and Klavdiya Bochenina. 2022. Reproducibility and Progress in Estimating Time of Arrival, or Can Simple Methods Still Outperform Deep Learning Ones?. In *KDD '22: SIGKKD International Conference on Knowledge Discovery and Data Mining, August 14–18, 2022, Washington, DC*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

KDD '22, August 14–18, 2022, Washington, DC

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Nowadays it is hard to imagine a city without cars. Despite the rise of taxi and car sharing services a lot of people decide to purchase their own vehicle [16]. With growing number of cars in urban areas the problem of efficient routing and accurate traffic services become more and more important. One of the most frequently used services by drivers is routing that provides the best route from one place in the city to another. One of the most important characteristic of a route is the so-called Estimated Time of Arrival (ETA) that tells how long the journey along the route will take, or when the trip will end.

In days of the current pandemic, delivery services received a huge boost [18, 20] which introduces the concept of ETA from traffic area to a completely different, everyday life of many people. Businesses related to delivery services are interested in providing an accurate ETA prediction to their customers in order to increase the quality of their service. This is also true outside of the scope of food and goods delivery. Many Business-to-Business companies are also interested in providing or receiving a correct ETA since long waiting time can lead to substantial expenses.

Given that a modern city is a large and intricate system containing a large amount of simultaneously moving cars, ETA prediction becomes a rather challenging task. Indeed, many factors greatly affects the traffic situation in the city: weather, configuration of the city streets themselves [40], actions and behavior patterns of a driver and other participants of the traffic [6] and so on.

Despite the growing interest in urban science, there are no generally accepted universal methods in the field of forecasting urban traffic flow or ETA prediction. The situation is worsened by that the recent progress in the field achieved by industrial navigational services, e.g. Google Maps, Baidu Maps or Yandex Maps, usually stays unrevealed for scientific society due to commercial secrets.

Even if we consider only the studies presented publicly in papers on the topic, indications exist of certain reprodicibility problems in today's research practice connected with the choice of state-of-the-art ETA prediction methods. An example is the lack of public datasets with transportation information (especially of a certain type) that prevents objective comparison between different approaches for ETA prediction and possibly impedes the progress in the field [2, 30]. Let us also mention that due to the complexity of urban traffic and numerous factors affecting a trip it is expected

than different methods may perform similarly to each other on average according to the well-known No Free Lunch theorem [1]. The diversity of possible known and unknown factors emphasizes the absence of a universal method for ETA prediction [11, 38].

Recently, problems of the above-mentioned kind have been reported in the area of other data mining tasks, e.g. in recommender system research [9, 10]. For example, the so-called state-of-the-art recommender systems based on Deep Learning are analyzed and compared with a set of simple ones in extensive experiments in [9, 10]. Surprisingly, it turns out that the complex solutions are often not reproducible in the sense that the corresponding source codes, datasets and even experimental results stay publicly unavailable. Furthermore, even if reproducible, the majority of them can be outperformed by the simple ones on open datasets thus making the performance progress claimed questionable.

Inspired by this troubling finding in the field of recommender systems, we intend to shed light on the analogous problems in the ETA research, especially in the case of methods adapted for GPS trajectory data. More precisely, in this work we review available datasets and existing solutions for ETA prediction from different domains to further highlight the above-mentioned problems of reproducibility and progress. Following the results in [9, 10] indicating that fine-tuned simple methods can outperform more complex Deep Learning solutions, we present *Strat-mETA* (a simple ETA prediction method designed to overcome the impact of the No Free Lunch theorem by stratifying the data and proper parameter tuning in simple regression-based ETA prediction methods) and compare it with several more complex methods by means of a multi-component quality metric and further discuss the surprising results obtained. In short, in this study:

- we overview the recent studies proposing ETA prediction methods found by us and determine the related troubling reproducibility issues (including the availability of open source code and datasets, especially of GPS trajectory data);
- we present and analyse a novel public dataset that includes real-world transportation GPS trajectory data as well as the traffic information at the corresponding moment in time;
- we propose *Strat-mETA*, a simple method based on fine-tuned regression ETA baselines and adapted for GPS trajectory data;
- we perform an experimental study where compare *Strat-mETA* with several existing ETA prediction methods including a Deep Learning-based one using the aforementioned dataset and surprisingly observe that our simple method outperforms them by means of a multi-component quality metric.

Let us emphasize that the code, dataset, and experimental results related to this study are publicly available on GitHub[1].

The paper is organized as follows: in Section 2, the existing methods for ETA prediction are reviewed as well as and the available datasets for them; in Section 3, the new transportation dataset and the motivation behind its preprocessing are presented; Section 4 contains the description of *Strat-mETA* and Section 5 presents the results of the experimental study on the proposed transportation dataset.

[1]https://github.com/AlgoMathITMO/Strat-mETA/

Table 1: Deep Learning-based ETA prediction methods

Deep Learning Method	Testes on open data	Open-source code	Dataset size	Data format
SSML [12]	✘	✘	~6M	own
ST-METANet [28]	✔	✔	~7M	sensors
DeepTTE [34]	✘	✔	~10M	trajectory
GN [11]	✘	✘	>10M	avg. speeds
MRA-BGCN [8]	✔	✘	~8M	sensor
ConSTGAT [13]	✘	✔	~4M	avg. speeds
DCRNN [21]	✔	✔	~8M	sensors

2 AN OVERVIEW OF RELATED WORKS

2.1 ETA prediction methods

There exist a variety of ETA prediction methods based on different technical ideas.

Historical speed-based methods [2, 30] use an approach to ETA prediction in which the overall duration of a trip is calculated from the speeds of each road segment constituting the route. The speeds are computed as average speeds of all vehicles on the road segment for a specific time window. Such approaches require a large amount of historical data and completely ignore the geometry of a route and real-time state of traffic. The latter may result in a rather poor performance.

Regression methods are also used for ETA prediction when the information about the trip is represented as a feature vector of the fixed length. Most of the time regression methods such as Random Forest Regressor [37], Support Vector Regressor [27, 36] or Gradient Boosting Regressor [33] perform well on small datasets and generally require less computational power to train in comparison to other approaches. As a drawback regression methods may not capture some of the implicit relations of an urban traffic environment.

Deep Neural Networks are frequently used for ETA prediction, especially in the relatively recent works, see Table 1. Most of them utilize LSTM (Long Short-Term Memory) layers to capture temporal connection between segments constituting the route. Those methods require large amount of data (see the reported size of datasets in Table 1) and take relatively long time to train. There also exist methods based on Graph Neural Networks that aim at capturing the geometry of a route [11]. This approach generally achieves good performance, however, require sophisticated data pre-processing and is rather difficult to reproduce due to its complexity.

As one probably expects, the methods based on *Historical speed* and *Regression* are relatively simple, have tested in many experimental studies and can be easily reproduced [2]. The situation with the methods based on *Deep Neural Networks* seems to be different, see Table 1. Indeed, it turns out that the corresponding source code is not always available and thus the reproducibility or replication process of them is hardly possible. What is more, some of the papers

use private datasets and thus a direct approval of the experimental results (that report a certain progress in ETA prediction quality) is questionable. This greatly limits the possibility of evaluation of the proposed methods and the objective comparison of them.

Another important feature of the ETA prediction methods that they are adapted for data of specific type. For example, some methods in Table 1 use data collected from road sensors and are not adapted from GPS trajectory transportation data. This will be discussed in the following subsection in more detail.

2.2 Available data and reproducibility in the field of ETA prediction

Despite the raise of the interest in urban sciences, there is a lack of open datasets that are universally accepted benchmarks for ETA prediction methods. Moreover, due to the possible sensitivity of transportation data, some of the researches use only private data and provide only limited statistics about the dataset.

In the works [3, 11, 31], authors provide basic statistics but the data used are *private* which makes it impossible to reproduce the described methods and objectively compare them. As an alternative to using private data or in absence of it some researchers use traffic simulators in order to acquire a dataset of desirable size [39]. While such approach may improve reproducibility, existing traffic simulators may not reflect the real-life traffic and often have major drawbacks which could make the experiments rather complicated [23, 29].

Unfortunately, many authors do not provide precise description of the data used often stating only the area where the data was collected. Table 2 shows the comparison between four datasets which were *likely* used as a source of the data in several papers. (We say *likely* because there are no formal references to the datasets in the papers, but only the names of the region (mainly cities in China) in which the data was collected. We found several datasets that match the descriptions and names and present them in Table 2.) The datasets describe traffic with a different quantities: average speeds (*avg. speeds*) for the average speed of vehicles passed a road during certain time interval, road sensor data (*sensors*) stands for the average speeds of vehicles passed a road during certain time interval acquired from sensors located in fixed locations and trajectories are sequences of GPS points with timestamps and meta-information.

Average speed and vehicle representation contain the information about the traffic at a certain moment of time but have the lack the data about trips at the moment of time. In [35], authors say that most of the modern studies in the ETA prediction area use data from road sensors or cameras and operate with real speeds or traffic flow. However, due to high cost of road sensors, such data is not widely available in many areas. Additionally, this is a *crucial drawback* of those representation types since the absence of real vehicle transportation does not allow for direct ETA predicting.

Due to it, data represented as a trajectory is preferable since it provides data on real trips along with possibility to acquire duration of the trip from GPS timestamps. Moreover, frequently GPS meta-information includes speed of the vehicles which may allow to estimate state of the traffic at a certain moment of time.

In our study, we focus on the ETA prediction methods that use this format of data.

Table 2: Comparison of ETA prediction datasets

Domain	Guangzhou[1]	METR-LA[2]	Guo F. [15]	Shenzhen[3]
type	avg. speeds	sensors	avg. speeds	trajectory
size	1.8M	7M	3B	6M
open	Yes	Yes	No	No

Note: [1]https://github.com/sysuits/urban-traffic-speed-dataset-Guangzhou
[2]https://gitcode.net/mirrors/liyaguang/DCRNN/-/tree/master/data/model/pretrained/METR-LA
[3]https://github.com/cbdog94/STL/tree/master.

2.3 Parameter tuning for baseline methods

Since we will deal with a fine-tuned simple method as a competitor to other ETA prediction methods (including Deep Learning-based), we are interested in the review of methods for parameter tuning.

Parameter tuning could be extremely useful in case when the available data is not large enough for model to properly optimize its parameters for a specific task. In order to overcome it, the model's hyperparameters could also be optimized. The optimization of hyperparameters does not differ a lot from regular function optimization. However, often number of parameters and internal complexity of a model makes the task of finding the analytical representation of a relation between hyperparameters and quality of the model virtually impossible. Due to it, two main things should be considered: optimization strategy and fitness functions.

Optimization strategy includes a vast majority of numerical optimization techniques which include but are not limited to simulation annealing, particle swarm optimization, genetic algorithms and so on [7, 25, 26]. Deterministic algorithms could be also used for hyperparameters optimization. They include linear search, taboo search, grid search and gradient search algorithms [5, 26]. Deterministic algorithms most of the time are rather simple to implement, however, due to sophisticated relation between input of the model and its output may lead to large number of expensive computations [4]. Since model's hyperparameters could be discrete, genetic algorithms are often used for hyperparameters optimization in Machine Learning [17, 22]. The idea is to represent the hyperparameters as an array of individual values of the parameter. The array is referred to as *individual*, and each parameter as *gene*. A number of randomly initialized individuals constitute the population. In order to evaluate the individuals, a fitness function is defined. It takes individual as its argument and outputs the individual's fitness value which is used during the algorithm.

3 DESCRIPTION OF OUR DATASETS

While designing a machine learning solution for the task, a variety of questions arise: studying the available data, choosing the appropriate way to represent data, choosing the model and setting its hyperparameters. Unfortunately, most of the time when working on a new task, it may be difficult to predict how the selected configuration would perform on a specific type of input data and in which conditions the selected configuration is optimal.

3.1 Data preprocessing

Considering the lack of public datasets and aforementioned constraints implied on the data required for ETA prediction, in this study we use the transportation data provided by StarLine Ltd.. The data received from StarLine Ltd. comes in a form of messages from GPS devices located on vehicles. Each message includes the location of the vehicle (latitude and longitude in WGS-84 projection), speed, timestamp, and unique identifier of the vehicle.

The dataset consists of two parts. First one consists of 515,169 tracks collected during one week in November 2021 and grouped by vehicle ID. Second part of the dataset is made public and consists of 298,089 tracks collected during one weekday grouped by vehicle ID. This dataset contains tracks which are up to 2 kilometers long. The real duration of a trip is acquired from timestamps of first and last GPS points comprising the track. The statistics of both of datasets is present in Table 3.

Raw GPS data generally requires some form of prepossessing, most frequently map-matching, i.e., the process of finding the best sequence of road segments which represents the sequence of raw GPS points. For this purpose, a routing machine can be used. Routing machines (or routing engines) are special frameworks which perform shortest path search on road graphs, i.e., graphs that represent the map of the area's roads and streets. One of the most used map data source used is the OpenStreetMap project [24]. The OSM map data is fed into the routing engine in order to find a route between two or more points or match raw GPS data to a routing graph. In this work, the Valhalla [32] routing engine is used.

Due to the imprecision of GPS devices, the received points generally do not align with the routing graph. Therefore we first apply the map-matching step, as described above. The data is aggregated in tracks using the unique vehicle IDs and then fed into Valhalla routing engine which matches the tracks to a road graph.

This expands the data set, adding to raw points the road segments on which the vehicle was when the measurement was done. Those GPS points may be grouped by the road segment they are assigned to. This allows to calculate average vehicle speed on a segment, based on vehicles which pass it during the last 10 minutes and estimate the traffic on the available roads of the city. In many works related to traffic and ETA prediction, authors use the term congestion index r. The congestion index is calculated as a ratio between the current speed of vehicles on the road segment and the free speed v_{free} of the segment (when there is no congestion and vehicles are freely passing through it), provided by StarLine Ltd. Historical data is used to estimate v_{free} as average speed of vehicles on the road segment, generally late at night. Based on the value of the congestion index, each road segment is classified as:

(1) jammed, if $r \in [0, 0.25)$,
(2) slow, if $r \in [0.25, 0.5)$,
(3) normal, if $r \in [0.5, 0.75)$,
(4) free, if $r \in [0.75, \infty)$.

When this classification is done, for each 10-minute interval of a day there is a list of road segments, each having a class assigned to it. Knowing the road segments constituting the route, we define how congested the route is by counting the road segments of each class. These values are used as features which describe the trips in the data set.

Additionally, the Valhalla routing service is used to calculate the ETA of each track, based purely on internal constant weights of the Valhalla's routing algorithm. Such ETA are thus generally not very accurate but can also provide valuable information for more accurate ETA prediction. The resulting feature vector describing each trip includes distance of a trip, Valhalla's ETA, hour when the trip is started and the number of road segments of each congestion class. We partition the data into non-overlapping training and test data, by the ratio of 8:2.

As it could be seen, the proposed dataset include both transportation data as well as the information about the traffic at the moment of the trip in contrast to other available datasets (Table 2).

Table 3: Statistics of our datasets

	N_p	N_t	d_p, m	t_p, s	L, m	τ, s
Private	63M	298, 089	5.7	3.9	9, 806	1, 662
Public	27M	515, 169	4.4	3.7	1, 834	279

Note: N_p is the number of GPS points in the dataset, N_t the number of tracks, d_p the average distance between two consequent GPS points, t_p the average interval between two consequent GPS points' timestamps, L the average distance of a trip, τ the average duration of a trip.

4 DESCRIPTION OF THE PROPOSED SIMPLE METHOD

In this section we present *Strat-mETA*, a simple method stratifying the data and using a parameter tuning approach for optimization of the number of baseline regression-based ETA prediction methods for each strata. A formal description of the method is given in Figure 2 and Algorithm 1. Below we describe the main steps of the method.

4.1 Grid-based stratification

As it was mentioned before, urban transportation depends on a vast spectrum of factors and behaviours of other traffic's participants. Due to that fact, some studies propose to divide data in groups and treat those independently. For example, in [11], authors separate data regions with similar traffic and mention that they use the same proposed method on each group separately.

Such approach allows to eliminate differences introduced by rather generic factors such as local geographical features of a terrain (e.g. changes of altitudes), city's layout and overall drivers behaviours. As an example, it should be safe to assume that traffic in New Dehli differs greatly from the traffic in a small city in Austria. Considering the fact that all the data used during this research was collected in one city, it is proposed to stratify the data by hour of the day and distance of a trip.

The motivation behind splitting the trips by hour is clear since during night it is rather relaxed compared to daytime and especially to rush-hours. Considering the size of the dataset it is implausible that explicit relation between daytime and ETA would be found. By splitting the dataset by starting hour of and treating each group

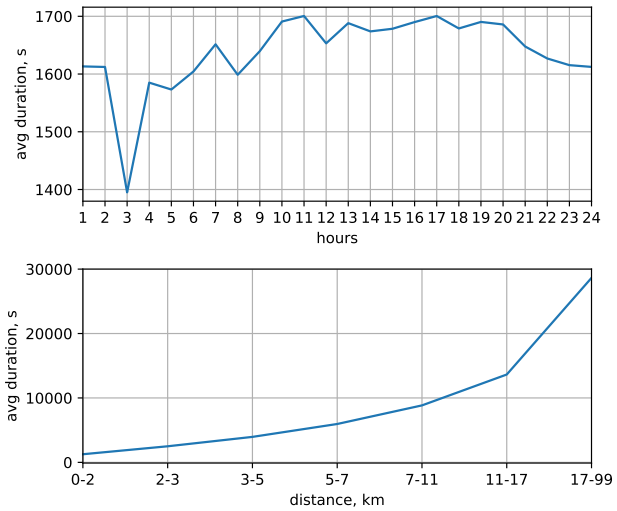

Figure 1: Dependency between trip's duration, hour of the trip and distance.

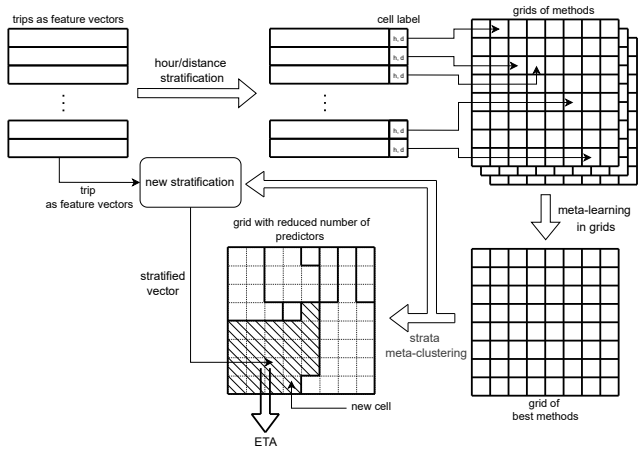

Figure 2: Scheme of the proposed method.

as a distinct task for a prediction it is expected to improve overall quality of a prediction.

Short distance trips generally take less time to complete, have less chances of facing unexpected traffic accidents and generally spend less time in traffic jams and highly congested areas. On the other hand, despite the fact that long trips may take more time to complete, they also may be more optimal: by taking longer route driver may evade highly congested regions of a city. Again, considering the size of the available dataset, it seems reasonable to split the data not only by hour but also by different routes' distances.

The mean distance of a trip in each group is present in Figure 1. As it can be seen, duration of a trip indeed grows larger with the distance of the trip, and also becomes larger during the day hours.

4.2 Parameter tuning and multi-objective optimization in grids

Since the data is stratified by two features, it is easy to represent it as a 2D-grid. Each cell of the grid corresponds to a single group of input data, and can be described by two indexes: starting hour and number of a distance interval. Considering it, we propose to treat each grid cell of the stratified dataset as a separate task.

As it could be seen from [2], different methods for ETA prediction may result in similar overall results. However, they also may show different results during different time of the day. Given the fact that data is also classified by distance, it may be expected that using dedicated method for each class of a trip would increase quality of the overall prediction.

In order to pick best model for each grid cell, the following approach is proposed: using genetic algorithm, we train several models for each grid cell of data. Then, the best method for each cell on the grid is picked based on the fitness function of optimized model. The prediction is made based on the class the input trip is assigned to: the appropriate model is selected from the grid based on the trip's class.

4.3 Strata clusterization

The design proposed ensures that the final model is the best, given the described stratification of the input trips and considered methods. However, the drawback of this approach is that depending on the stratification, the number of cell grids might be large and thus would require a lot of resources in practical use for ETA prediction.

In order to reduce number of the methods in the grid, we propose to cluster the strata (grid cell) meta-information about the selected methods as features for it. Such clusterization of the initial grid cells would allow to find similar grid cells of the data and decrease the overall number of methods used for the prediction. Of course, this procedure may decrease the overall quality of the proposed method. However, since clustering of grid cells is performed based on similarity between performance of estimators for each grid cell it is expected that the drop in prediction quality is relatively small. As a payback, the reduced number of estimators for the whole grid would decrease the amount of resources required for training and prediction. Moreover, the acquired clusters could be used as a baseline for a novel data stratification strategy: instead of classifying data by hour and distance a new classifier could be designed to stratify data based on the similarities between the clusters found.

4.4 Complexity analysis

The analysis is straightforward. The complexity of Algorithm 1 implementing the above-described method *Strat-mETA* linearly depends on the number of cells in the grid. At the same time, complexity of each cell is the sum of complexities of the candidate methods, optimization strategy and optimized hyperparameters. So, overall complexity of the whole algorithm is the sum of complexities of all cells in the grid and complexity of the clustering algorithm.

Considering that, in order to lower the complexity of the proposed algorithm the size of the grid may be reduced. However, according to the No Free Lunch theorem it would result in lower

Algorithm 1: *Strat-mETA*

```
Data: dataset, methods, hour_labels, dist_labels, weights
grid[][];
candidates[];
dataset = stratify(dataset);
for h in hour_labels do
    for d in dist_labels do
        for method in methods do
            train, test = dataset[h, d].split();
            opt_model =
             optimize(method, train, test, weights);
            candidates.put(opt_model);
            best_method = select_best(candidates);
            candidates.clear();
            grid[h][d] = best_method;
        end
    end
end
clustered_grid = cluster(grid);
```

Table 4: Optimized hyperparameters in the methods used

Method	Hyperparameters
RFR	# estimators: [1, 100], criterion: {squared_error, absolute_error, poisson}, min_sample_split: [0.001, 1.0], max_features: {auto, sqrt, log2}
GBR	loss: huber, quantile, learning_rate: [0.001, 2.0], n_estimators: [10, 100], criterion: fiedman_mse, mse, mae
SVR	kernel: rbf, sigmoid, gamma: auto, scale, tol: [0.001, 1.0], C: [0.1, 3]

performance. As it was mentioned before, to overcome this drawback, we propose to use clusterization to lower complexity of the proposed solution while minimizing the performance loss.

5 EXPERIMENTS

5.1 Construction of *Strat-mETA*

Taking into account the conclusion in the comparative study in [2] about the quality of simple ETA prediction methods, we select the following baseline methods to be used in *Strat-mETA*:

- Random Forest Regression (RFR);
- Gradient Boosting Regression (GBR);
- Support Vector Regression (SVR).

In order to optimize hyperparameters of the methods, a genetic algorithm is used. The parameters for tuning are present in Table 4. The simple evolution strategy is characterized by the number of

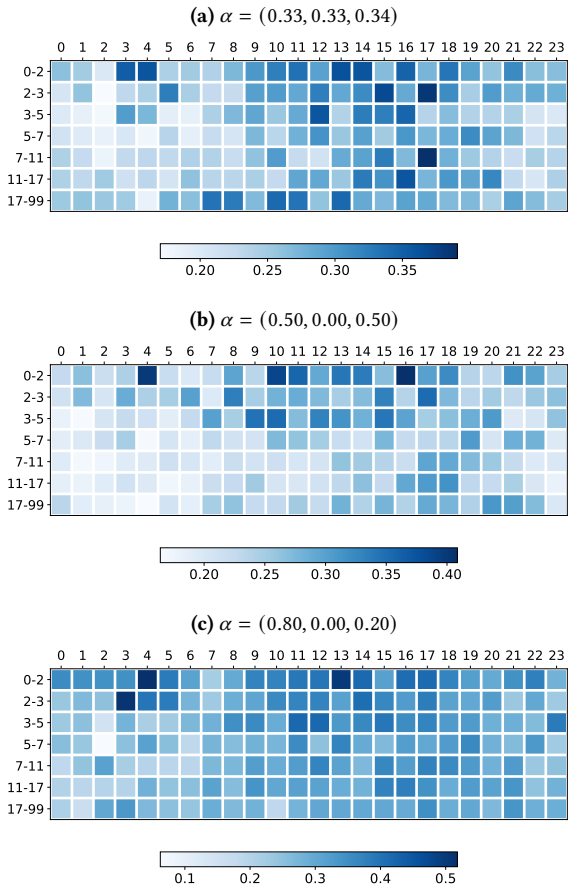

Figure 3: Acquired fitness function values for each set of weights α.

individuals in the experiments, the number of generations and the mutation rate. The configuration for the experiments are 10 individuals, 20 generations and 0.5 mutation rate. *Deap* module [14] was used in order to perform the optimization. The fitness function F used for evaluation is parametric, to control the impact of multiple chosen criteria on the optimization results:

$$F = \alpha_1 MAPE^* + \alpha_2 MAE^* + \alpha_3 T_p^*, \quad \sum_{k=1}^{3} \alpha_k = 1, \quad \alpha_k \in [0, 1]. \quad (1)$$

In (1), $MAPE^*$ and MAE^* are the metric values collected during evaluation of an individual route, and T_p^* is the time spent on one prediction. The symbol * signifies the MinMax normalization, and the maximum and minimum values of the variables are estimated using the sample of the dataset. Impact of the k-th metric could be tuned using α_k weight parameter.

The goal of the genetic algorithm is to minimize the fitness function and consequently the MAPE, MAE and prediction time of the model. In order to define appropriate weights α_k in (1), the algorithm was executed 3 times with different weights.

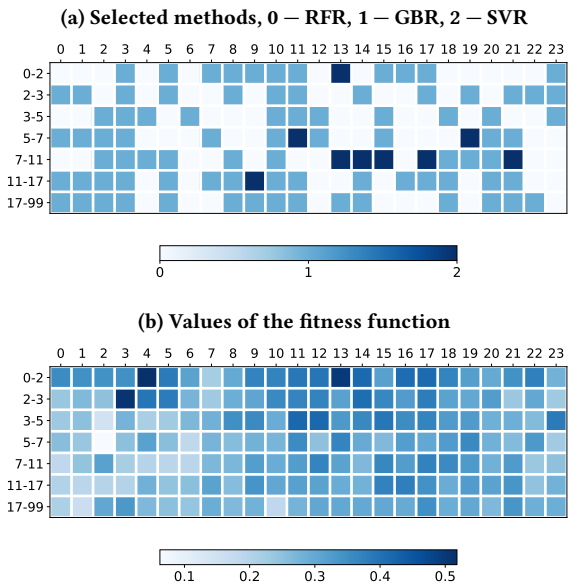

Figure 4: Selected method and best values of fitness function for each grid cell.

Table 5: Results of experiments for different weights α_k. The best result in each row is shown bold.

Metric	(0.33, 0.33, 0.34)	(0.50, 0.00, 0.50)	(0.80, 0.00, 0.20)
F	0.27	**0.24**	0.30
$MAPE$	36.81	34.78	**32.93**
T_p, ms	**0.39**	2.41	0.81

Figure 3 shows the fitness function values of each grid cell for three different sets of weights α. It can be seen that choosing different sets of weights results in noticeable differences between acquired grid: fully omitting MAE metric from fitness functions ($\alpha = (0.50, 0.00, 0.50)$) seemingly provides best overall results based on fitness function values.

The numerical results of the experiment is presented in Table 5. Different weights clearly influence the overall performance of the proposed approach. During further experiments weights $(0.50, 0.00, 0.50)$ will be used due to the lowest fitness function and second best MAPE.

Figure 4 shows best picked methods for each grid cell as well as the fitness values of the said methods for each class.

Each cell in Figure 4 signifies a distinct method with hyperparameters optimized specifically for one trip class, so 168 distinct models should be trained. Hence, to reduce the overall number of used models, we propose to use clusterization. In order to capture the relation between quality of the method and class of the trip it is assigned to we propose to use DBSCAN algorithm [19] with Euclidean metric using fitness function F and prediction time T_p as feature. This way methods which shown similar performance would end up in the same cluster without great increase of prediction time.

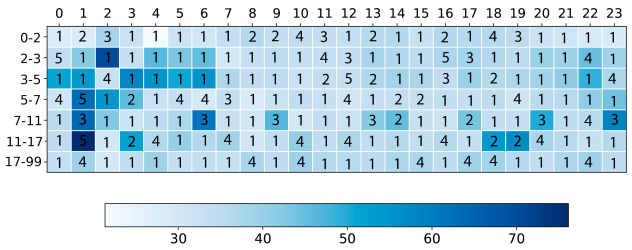

Figure 5: MAPE for each trip class using the found 5 clusters (1-5) for prediction.

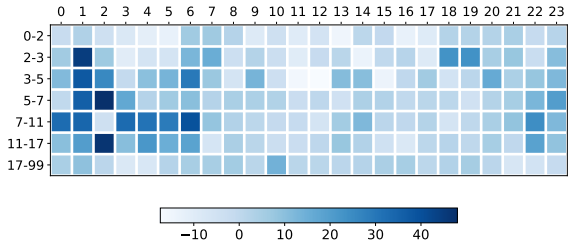

Figure 6: Difference in MAPE for grid cells after clustering, each cluster shown with distinct hatching.

Figure 5 shows results of the clusterization by presenting the MAPE values for each grid cell and displaying found clusters by different hatching types. Overall 4 clusters with similar fitness function and prediction time were found. Inside each cluster the best method was chosen and assigned to the said cluster. For the further experiment, those methods and their hyperparameters were used for training and prediction for all the trips constituting the corresponding cluster.

In Figure 6, one can see the difference between MAPE for each trip class achieved using the initial proposed method ($MAPE$) and with trip classes acquired at the clusterization step ($MAPE_{cl}$):

$$\Delta = MAPE_{cl} - MAPE. \tag{2}$$

As it can be seen, the clusterization improved the MAPE error for most of the initial classes. However, from Figure 6 it can be clearly seen that during night time and rush-ours (approximately from 2 to 6 hours) the performance of the proposed method on average or long trips is significantly worse. It could be explained with the fact that during the night hours there are less trips present in the dataset, and during rush-ours the city is very congested and unpredictable.

Thus we have constructed the following two versions of our simple ETA prediction method on our datasets: *Strat-mETA* without and with the clusterization step.

5.2 Choice of the competitors

Now let us discuss the choice of Deep Learning-based ETA prediction methods as competitors in our comparison experiments. The ones in Table 1 with an open source code provided are *ConSTGAT* [13], *ST-METANet* [28], *DCRNN* [21] and *DeepTTE* [34]. Nevertheless, only *DeepTTE* [34] is adapted for GPS trajectory data as in

our dataset and thus only this method can be a representative of Deep Learning ones in our case.

We also use the well-known *HAS* [2] and *RT-ETA* [11] as competitors that may be thought to be a more complex ETA prediction method than *Strat-mETA* but less complex than DeepTTE [34].

Summarizing, we perform experiments on our datasets using the following suitable methods for ETA prediction:

- *PM* that is an abbreviation for *Strat-mETA* without the clusterization step;
- PM_{cl} that is an abbreviation for *Strat-mETA* with the clusterization step;
- *HAS* [2] that stands for Historical Average Speeds approach where ETA is computed based on speeds of vehicles collected before the actual time of the trip [2]; average speed values is computed for each 15 minutes using the whole dataset;
- *RT-ETA* [11] that stands for Real-time Travel Times is similar to HAS but real-time speeds of vehicles are used for ETA prediction; for each 10-minute interval in the dataset average speed values are computed using the data from the previous 10 minutes;
- *DeepTTE* [34] that is a Deep Neural Network processing trip data as sequences of GPS points; its hyperparameters are set to the default ones proposed by its authors.

Let us mention that the methods *HAS* and *RT-ETA* provide the ETA estimation based on fully pre-computed data and therefore prediction time for the methods is not considered.

5.3 Results

The overall experimental results are presented in Table 6. It is clearly seen that the proposed method with and without the clusterization step (PM_{cl} and *PM*, correspondingly) outperforms the other methods both on our private and public datasets. Furthermore, *PM* is just slightly better in MAPE and MAE than PM_{cl}, while the former is worse than the latter by means of computational time (as expected by construction). Particularly, with the MAPE increase of only 3.15 p.p. on the public dataset, PM_{cl} contains only 5 estimators in comparison to 168 in *PM*. Note that, as in the case of PM_{cl} and *PM*, *RT-ETA* and *HAS* show rather similar performance on both datasets outperforming *DeepTTE*. Surprisingly, *DeepTTE* achieves the worst results by means of MAPE, MAE and computational time.

Let us mention that all the methods show degradation in performance on the public dataset in comparison to that on the private one. It can be explained by the fact that all routes in the private dataset are shorter than 2 km. The degradation is most noticeable in the case of *DeepTTE* and this probably because Deep Learning methods are rather sensitive to the size and diversity of datasets.

Table 6: Results of the comparison experiments. The best result in each row is shown bold.

Dataset	Metric	*PM*	PM_{cl}	*HAS*	*RT-ETA*	*DeepTTE*
private	*MAPE*	**34.81**	38.03	53.19	44.17	47.12
	MAE, s	**483**	504	893	735	947
	T_p, ms	2.41	**1.83**	—	—	78.12
public	*MAPE*	**36.12**	39.27	55.83	51.40	84.23
	MAE, s	**513**	536	953	782	1143
	T_p, ms	1.32	**1.14**	—	—	56.65

6 CONCLUSION AND FUTURE WORK

During this work we compared several modern approaches for ETA prediction along with open transportation datasets in the sense of reproducibility. As it can be seen from Table 1, only a part of modern solutions provide code and data allowing for a reproducible comparison and evaluation. Moreover, the widely used road sensors datasets limits the area of applicability of the methods since such infrastructure is not presented in many cities.

Furthermore, during this work we proposed *Strat-mETA*, a simple method of ETA prediction that takes stratified data as the input, optimizes a number of simple regression models for each strata and performs clustering to reduce the overall number of distinct models. For evaluation of the proposed solution, a novel car travel dataset containing GPS trajectory data was introduced and made *public* in order to encourage the reproducible research in the field of ETA prediction. The dataset was also used for comparing the proposed method and the existing approaches for ETA prediction.

Moreover, our experimental study particularly on the public dataset showed that *Strat-mETA* with and without the clusterization step surprisingly outperforms the other considered solutions (including the Deep Learning-based one) in a significant way. What is more, in accordance to the No Free Lunch theorem, *Strat-mETA* without the clusterization step showed better MAPE and MAE values than than that with the clusterization step. Moreover, the latter one contained only 5 estimators in comparison to 168 in the former at the cost of only 3.15 p.p. of MAPE and 23 s of MAE (on the public dataset). This resulted in that *Strat-mETA* with the clusterization step turned out to be faster than that without the step. Let us also mention that our method was the fastest in overall experiments (much faster than the Deep Learning-based one).

Summarizing, we confirmed the present of reproducibility and progress issues in the field of ETA prediction similar to those reported for recommender systems [9, 10]. In particular, it turned out that many existing methods are not reproducible in the sense that their source codes and datasets and that simple ETA prediction methods can still outperform more complex ones.

As a continuation of this work we plan to consider new strategies of the initial dataset stratification and clusterization methods in order to find more implicit relations in the data. This possibly could allow us to improve performance of *Strat-mETA* method and more efficiently utilize limited traffic data. Furthermore, we are interested in performing more comparison experiments with other ETA methods adapted for GPS trajectory data as in our case.

ACKNOWLEDGMENTS

This work was supported by the Analytical Center for the Government of the Russian Federation (IGK 000000D730321P5Q0002), agreement No. 70-2021-00141.

REFERENCES

[1] Stavros P. Adam, Stamatios Aggelos N. Alexandropoulos, Panos M. Pardalos, and Michael N. Vrahatis. 2019. No free lunch theorem: A review. https://doi.org/10.

1007/978-3-030-12767-1_5

[2] Rami Al-Naim and Yuriy Lytkin. 2021. Review and comparison of prediction algorithms for the estimated time of arrival using geospatial transportation data. *Procedia Computer Science* 193 (2021). https://doi.org/10.1016/j.procs.2021.11.003

[3] Andreas Balster, Ole Hansen, Hanno Friedrich, and André Ludwig. 2020. An ETA Prediction Model for Intermodal Transport Networks Based on Machine Learning. *Business and Information Systems Engineering* 62 (2020), 403–416. Issue 5. https://doi.org/10.1007/s12599-020-00653-0

[4] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011.*

[5] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13 (2012).

[6] Jin Cao and Monica Menendez. 2015. System dynamics of urban traffic based on its parking-related-states. *Transportation Research Part B: Methodological* 81 (2015). https://doi.org/10.1016/j.trb.2015.07.018

[7] Marco Cavazzuti. 2013. *Optimization methods: From theory to design scientific and technological aspects in mechanics.* https://doi.org/10.1007/978-3-642-31187-1

[8] Weiqi Chen, Ling Chen, Yu Xie, Wei Cao, Yusong Gao, and Xiaojie Feng. 2020. Multi-Range Attentive Bicomponent Graph Convolutional Network for Traffic Forecasting. In *Proceedings of the Thirty-Fourth Conference on Association for the Advancement of Artificial Intelligence (AAAI).* 3529–3536.

[9] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Trans. Inf. Syst.* 39, 2, Article 20 (jan 2021), 49 pages. https://doi.org/10.1145/3434185

[10] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19).* Association for Computing Machinery, New York, NY, USA, 101–109. https://doi.org/10.1145/3298689.3347058

[11] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, Peter W. Battaglia, Vishal Gupta, Ang Li, Zhongwen Xu, Alvaro Sanchez-Gonzalez, Yujia Li, and Petar Velickovic. 2021. ETA Prediction with Graph Neural Networks in Google Maps. *International Conference on Information and Knowledge Management, Proceedings.* https://doi.org/10.1145/3459637.3481916

[12] Xiaomin Fang, Jizhou Huang, Fan Wang, Lihang Liu, Yibo Sun, and Haifeng Wang. 2021. SSML: Self-Supervised Meta-Learner for en Route Travel Time Estimation at Baidu Maps. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* https://doi.org/10.1145/3447548.3467060

[13] Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. 2020. ConSTGAT: Contextual Spatial-Temporal Graph Attention Network for Travel Time Estimation at Baidu Maps. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* https://doi.org/10.1145/3394486.3403320

[14] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research* 13, 70 (2012), 2171–2175. http://jmlr.org/papers/v13/fortin12a.html

[15] Feng Guo, Dongqing Zhang, Yucheng Dong, and Zhaoxia Guo. 2019. Urban link travel speed dataset from a megacity road network. *Scientific Data* 6 (2019). Issue 1. https://doi.org/10.1038/s41597-019-0060-3

[16] Yue Guo, Fu Xin, Stuart J. Barnes, and Xiaotong Li. 2018. Opportunities or threats: The rise of Online Collaborative Consumption (OCC) and its impact on new car sales. *Electronic Commerce Research and Applications* 29 (2018). https://doi.org/10.1016/j.elerap.2018.04.005

[17] Ji Hoon Han, Dong Jin Choi, Sang Uk Park, and Sun Ki Hong. 2020. Hyperparameter Optimization Using a Genetic Algorithm Considering Verification Time in a Convolutional Neural Network. *Journal of Electrical Engineering and Technology* 15 (2020). Issue 2. https://doi.org/10.1007/s42835-020-00343-7

[18] Jill E. Hobbs. 2020. Food supply chains during the COVID-19 pandemic. *Canadian Journal of Agricultural Economics* 68 (2020). Issue 2. https://doi.org/10.1111/cjag.12237

[19] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, S. Sarasvady, and Amrita Vishwa. 2014. DBSCAN: Past, present and future. *5th International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2014.* https://doi.org/10.1109/ICADIWT.2014.6814687

[20] E A Kolomak. 2020. Economic effects of pandemic-related restrictions in Russia and their spatial heterogeneity. *R-Economy. 2020. Vol. 6. Iss. 3* 6 (2020), 154–161. Issue 3.

[21] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[22] Sehla Loussaief and Afef Abdelkrim. 2018. Convolutional Neural Network hyperparameters optimization based on Genetic Algorithms. *International Journal of Advanced Computer Science and Applications* 9 (2018). Issue 10. https://doi.org/10.14569/IJACSA.2018.091031

[23] S. M.Sohel Mahmud, Luis Ferreira, Md Shamsul Hoque, and Ahmad Tavassoli. 2019. Micro-simulation modelling for traffic safety: A review and potential application to heterogeneous traffic environment. Issue 1. https://doi.org/10.1016/j.iatssr.2018.07.002

[24] Gunther Maier. 2014. OpenStreetMap, the Wikipedia map. *Region* 1 (2014). Issue 1. https://doi.org/10.18335/region.v1i1.70

[25] Rafael G. Mantovani, André L.D. Rossi, Joaquin Vanschoren, Bernd Bischl, and André C.P.L.F. Carvalho. 2015. To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning. *Proceedings of the International Joint Conference on Neural Networks* 2015-September. https://doi.org/10.1109/IJCNN.2015.7280644

[26] Mario A. Muñoz, Yuan Sun, Michael Kirley, and Saman K. Halgamuge. 2015. Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Information Sciences* 317 (2015). https://doi.org/10.1016/j.ins.2015.05.010

[27] Rafidah Md Noor, Ng Seong Yik, Raenu Kolandaisamy, Ismail Ahmedy, Mohammad Asif Hossain, Kok-Lim Alvin Yau, Wahidah Md Shah, and Tarak Nandy. 2020. Predict Arrival Time by Using Machine Learning Algorithm to Promote Utilization of Urban Smart Bus. (2 2020). https://doi.org/10.20944/PREPRINTS202002.0197.V1

[28] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1720–1730.

[29] Andreas Pell, Andreas Meingast, and Oliver Schauer. 2017. Trends in Real-time Traffic Simulation. *Transportation Research Procedia* 25. https://doi.org/10.1016/j.trpro.2017.05.175

[30] Thilo Reich, Marcin Budka, Derek Robbins, and David Hulbert. 2019. *Survey of ETA prediction methods in public transport networks.*

[31] Tomoki Saito, Shinichi Tanimoto, and Fumihiko Takahashi. 2021. Hierarchical Positional Approach for ETA Prediction. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems.* https://doi.org/10.1145/3474717.3488240

[32] valhalla. [n. d.]. *Valhalla - Open Source Routing Engine for OpenStreetMap.* Retrieved January 31, 2021 from https://github.com/valhalla/valhalla

[33] Sander-Sebastian Värv. 2019. Travel Time Prediction Based on Raw GPS Data.

[34] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. 2018. When will you arrive? Estimating travel time based on deep neural networks. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018.*

[35] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. 2021. Deep learning on traffic prediction: Methods, analysis and future directions. *IEEE Transactions on Intelligent Transportation Systems* (2021).

[36] Bin Yu, Yong Lei Jiang, Bo Yu, and Zhong Zhen Yang. 2008. Application of support vector machines in bus travel time prediction. *Dalian Haishi Daxue Xuebao/Journal of Dalian Maritime University* 34 (2008). Issue 4. https://doi.org/10.11648/j.ijse.20180201.15

[37] Bin Yu, Huaizhu Wang, Wenxuan Shan, and Baozhen Yao. 2018. Prediction of Bus Travel Time Using Random Forests Based on Near Neighbors. *Computer-Aided Civil and Infrastructure Engineering* 33 (2018). Issue 4. https://doi.org/10.1111/mice.12315

[38] Xin Zhang, Yanhua Li, Xun Zhou, Oren Mangoubi, Ziming Zhang, Vincent Filardi, and Jun Luo. 2021. DAC-ML: Domain Adaptable Continuous Meta-Learning for Urban Dynamics Prediction. *2021 IEEE International Conference on Data Mining (ICDM)*, 906–915. https://doi.org/10.1109/ICDM51629.2021.00102

[39] Bing Zhao, Yon Shin Teo, Wee Siong Ng, and Hai Heng Ng. 2019. Data-driven next destination prediction and ETA improvement for urban delivery fleets. *IET Intelligent Transport Systems* 13. Issue 11. https://doi.org/10.1049/iet-its.2019.0148

[40] Shuangming Zhao, Pengxiang Zhao, and Yunfan Cui. 2017. A network centrality measure framework for analyzing urban traffic flow: A case study of Wuhan, China. *Physica A: Statistical Mechanics and its Applications* 478 (2017). https://doi.org/10.1016/j.physa.2017.02.069