ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA241064

Estimating the Time of Arrival of a Shipment with Machine Learning

Francesco Gibellini[a,*], Bart van Gool[a], Murat Firat[a] and Stefano Bromuri[a]

[a]Open University of the Netherlands, Computer Science Department
Heerlen, Netherlands

Abstract. Effectively estimating the arrival day of a shipment is an important capability for an express courier company to ensure both customer satisfaction and internal operations efficiency. This paper studies predicting the estimated time of arrival (ETA) of a package, shortly denoted by PETAP, under a collaboration with an industrial partner. Our approach employs machine learning (ML) techniques: CatBoost, multi-layer perceptrons (MLPs) with categorical embeddings, and Transformer neural networks, to predict delivery dates based on shipment locations and status. Challenges such as complex inter-modal networks and high-cardinality categorical features are addressed. Our paper contributes to the literature by formalizing the PETAP problem in the context of express shipping: the proposed models outperform the current business baseline accuracy by more than 25%. In our experimentation with a dataset including millions of data-points we propose a tabular vs sequence to sequence approach observing the superiority of the former. Future research directions include explicit modeling of transportation networks and exploring alternative ML approaches for improved prediction accuracy.

1 Introduction

Knowing when a shipment is likely to be delivered to the customer is a key piece of information an express courier business should know at all times about all shipments. In [7] several hundred respondents highlighted how *reliability* is the most important service quality factor when it comes to courier, this includes sub-factors such as successful and timely delivery which are both highly connected to estimated time of arrival (ETA) prediction. This knowledge is needed for two main reasons: (i) effective communication with the receiver on the delivery date, and (ii) effective internal tracking of shipments that are likely to be late, providing concrete input for dealing with shipment delays. An accurate communication with the consignee was measured to reduce the number of re-delivery attempts by at least 10%, reducing last mile costs greatly. Moreover it was measured that correct communication about ETA can reduce the number or customer service calls by more than 30%.

To our knowledge the relevant literature contains two main groups of approaches for finding the ETA of a shipment. First one explicitly models the underlying transportation network [2] and one other one does not explicitly model it, often referred to as 'data-based' as they can leverage patterns contained in large datasets to implicitly learn the underlying system [10]. The most advanced techniques to follow the latter approach belong to the field of Machine learning (ML), a field focusing on the development of algorithms enabling computers to learn patterns from data without explicit programming.

The application under consideration of this paper is to predict the delivery date of a shipment transported by our industrial partner, we refer to this problem as PETAP (Predicting the Estimated Time of Arrival of a Package). This application includes multiple data analytics challenges. First, an international shipment typically travels through several facilities of an international network and a supervised learning approach has to deal with data duplication, sequences, noisy trajectories, embedded in environments following different rules. Second, the data includes a set of categorical features of high-cardinality, and continuous features. Even advanced machine learning models face challenges when dealing with categorical features and their relation to continuous features.

The goal of this paper is to explore potential Machine Learning approaches to make reliable predictions concerning the arrival day of a shipment at any intermediate location on its route. For this purpose, the paper discusses the relevant features for the modeling of the problem and considers a number of machine learning algorithms of different representation complexity. Specifically, in our study, concerning tabular data, we compare CatBoost [5], with multi-layer perceptrons using categorical embeddings (following a categorical embedding schema [17]) and, by adapting the data to be sequential, we also use a Transformer neural networks [14]. Other Advanced deep learning methods, often used when road networks are involved ([4], [3], [19]) are not easily adaptable to our problem given the temporal nature of the air cargo network utilized by the express courier, and given the fact that the network in which the PETAP shipments travel is of multi-modal nature, sometimes involving a combination of road networks with air cargo networks.

The ETA prediction problem has two main challenges. First it is a high-precision regression problem in hours / minutes. Second, it requires developing advanced ML models to learn highly complex shipment processing patterns in depth. This is too complex to handle at once, hence in this paper we settled to define a prediction problem to estimate the delivery day of the shipment and we developed a hierarchical approach to tackle it. In our opinion, only later it is reasonably possible to estimate the exact arrival time. ([16]).

The contribution of this paper is twofold. First, obtaining more than 25% improvement in the ETA prediction accuracy of a large express courier by developing advanced Machine Learning models to benchmark the current business practice. Second, to the our knowledge our work is the first attempt for creating a first formalization of the features and data structures needed for this application that, in the

* Corresponding Author. Email: francesco.gibellini@ou.nl.

context of express shipping, particularly concerning evaluating daily precision.

The rest of this paper is organized as follows: Section 2 discusses related work, Section 3 describes the PETAP problem and its business relevance, Section 4 discusses the different techniques used to solve the problem and address its many challenges, Section 5 describes the various experiments we conducted in order to chose the best methodology to solve the problem, Section 6 discusses the experiment results and, last Section 7 highlights our conclusions and suggestions for future work.

2 Related Work

ETA prediction using ML techniques was explored in several different transportation types (air, ground, sea) and contexts, thanks to its high relevance for transportation planning and customer satisfaction. Depending on the specific problem context, sometimes an explicit modeling of the underlying system is necessary. In [2] for example this is modeled directly and, a combination of ML and business driven logic is used to drive the ETA calculation. On the other hand systems such as the one developed at Google([4]) or in [19] highly leverage state of the art Deep Learning techniques rather than manually crafted business logic to drive the ETA calculation.

Unimodal Transportation The paper [9] attempts to estimate the arrival of ships at the destination port using a combination of internal historical and traffic data, helping in managing resources in the downstream supply chain. ETA prediction of aircrafts using advanced geo-temporal modelling is considered by [15] and the trained model considers both historical and traffic data.

Work from Google Maps [4] shows how in the context of street traffic using Graph Neural Networks can help improve ETA prediction by several percentage points when compared with more simple model types; knowing this became relevant to our work during performance improvement iterations.

The above mentioned contributions predict ETA of a certain vehicle based on its current position and destination. The problem in our application on the other hand is to predict ETA of a package being transported by one or more vehicles in a heterogeneous and international network. Nonetheless it is relevant to compare our work also with these works for two main reasons, (i) these works focus on a subset of the problem we try to solve and, (ii) their advanced usage of geo-spatial features as well as traffic related features could be adapted to our work. As a matter of fact, this work was influenced by the above mentioned studies, as some of the aggregated features defined in this contribution aim precisely at representing properties of the subsystems composing the network in which the shipments travel.

Shipping Most of the related work on ETA in Express shipping focuses on predicting the arrival time of a shipment assuming the day of delivery is known, this is mainly a function of the courier's route. Multiple papers ([3] [19][16]) showed that models accounting for (courier) route information are superior to the ones ignoring it. The contribution in [16] analyzes different ways of predicting the route and ETA in a two-step approach in which first, a courier route is calculated and then an ETA is predicted for the packages to deliver (or pick-up) based on that route. In order to calculate the most likely route of a courier both search (Vehicle Routing Problem of the courier van) and deep learning techniques are tested showing the superiority of the second over the first. This is due to two main reasons: (i) computational performance of the optimization methods is inferior to that of (pre-trained) deep learning models due to the NP-hard nature of the Vehicle Routing Problem and, (ii) low accuracy of arrival time prediction due to the (too simple) linear function used after the route calculation step.

While the above mentioned papers provide interesting insights into which methodologies are best suited to solve the prediction of the time at which a courier will reach a certain location, our work problem requires us to predict which day the said courier will receive each package. Despite this difference important findings such as the high relevance of the traveled route could be incorporated in the solution to the PETAP.

Our paper has remarkable similarity to the study in [2] as it aims to predict ETA of shipments in a complex inter-modal network. While this network is made of ships and trains, the network we will use is made mainly of aircrafts and trucks. Moreover the authors model the transportation network explicitly such that the consequence of a possible delay is incorporated by taking lately scheduled transportation event. This results in obtaining a set of scenarios, each having a corresponding probability distribution of the delivery.

3 Problem Definition

Predicting the day a certain shipment will be delivered is a task dependent on several factors which can be divided into two main categories; transportation network and the nature of shipment.

The transportation network (of the courier company) entails the locations and the connections between them for the successive movements of a shipment. The movement among these connections does not only depend on geographical distance, but also the transport channels of the courier company. For example, two locations might be very close in terms of geo-distance but if there is no way to transport goods between them due to a lack of flights / trucks, shipments will have to wait to be transported until a way to transport them is available.

The nature of a shipment is also an important factor if it will be prioritized or not. Certain medical shipments for example are more urgent than other shipment types, this causes them to usually have shorter transit times. Heavier shipments being harder to handle might be transported differently than lighter shipments which are easier to handle.

The partner courier company transports millions of packages every month globally. Each of these packages receives dozens of checkpoints (also called 'events') during it's journey from origin to destination. The prediction can be needed at any point in time during the course of the shipment journey to ensure not only clear and accurate communication with the customer but also real-time internal tracking of a shipment's status.

Our approach does not explicitly consider the routing of the package in transportation network, hence it solves the PETAP without involving the highly complex combinatorial challenges that have stochastic nature. Figure 1 shows examples of such routes for a few shipments.

Predicting ETA can be stated both as a classification and a regression problem, since the target feature, i.e. the number of days till the delivery, is of numerical type with discrete values. In our computational experimentation, we benchmarked the classification and regression models by converting the prediction results from one type to another.

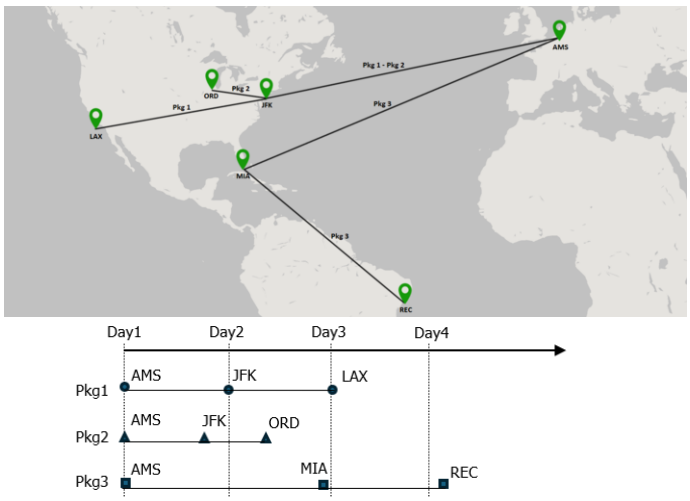

Figure 1. Example of routes taken by multiple international shipments

4 Predictive Modeling Approach

This section will describe how an effective ML system for solving the PETAP was developed starting from the proprietary dataset of the courier business we partnered with.

4.1 Historical Data

Data descriptives The data analysed in this study has been extracted from the IT system of our industrial partner. An instance of our dataset is a progress scan of a shipment during its journey between the pickup at the origin and the delivery at the destination. A shipment usually has between 20 and 50 data points, i.e. progress scans in the transport network on its route. The data size for one month time span reaches the magnitude of billions for several millions of shipments.

Since the shipment flow volume can vary significantly at different origins and destinations, data sampling should be carefully done to prevent some classes of features from getting under-represented. A high amount of features are of categorical type and have a class set of high cardinality, e.g. "shipment type" (with classes 'slow', 'fast', 'international'...), and "scan type" (with classes 'arrival', 'departure', 'security', 'customs'...).

For the sake of efficiency in the data content, we define *statistical features* that contain statistical summary of transport characteristics of each shipment type in the last three-week period before the start of the time span of training data, taken as 10 days in our experimentation. Including aggregated features is useful for covering the information of much longer time than only the training period. In Section 5 we will further discuss the impact of aggregated features.

The test data time span is the following 10 days after training time period. Since the data has time-series aspect, we split the training and test data such that no shipment has progress scans in both training and test data, hence no leaks are allowed between training and test data.

Features There are three main groups of features:

- *Shipment features* describe the static properties of a shipment, e.g. weight, price, and product code.
- *State features* include the dynamic information of the shipment at the time point of the progress scan, like its location and its process phase there, e.g. arrived or processed at a facility.
- *Statistical features* are aggregations from past data about the general behaviour of a shipment given its state in the network (e.g. average, minimum, and maximum time to get from current location to delivery). These features are calculated aggregating historical data by different subsets of features, in our work 8 possible combinations of grouping columns are used. This is done to cope with the very diverse shipment states present in our dataset as, for certain data-points a given aggregation level might be too general (as this is a common type of data-point) while for other data-points it might be too specific (as this type of data-point is not very common hence an historical mean might not describe historical behavior accurately due to small sample size).
  An example of historical aggregation states are :
  - **Current Facility, Destination Facility, Event Code**: This is a very specific grouping helping the model by providing it features of historical behavior for shipments in the same facility, receiving a specific scan as well as with the same destination as our current data-point.
  - **Current Country, Destination Country, Event Weekday**: This is a general grouping helping the model by providing it features of historical behavior for shipments in the same country, on the same weekday, as well as with the same destination country as our current data-point.

A complete list of the features used and their nature can be found in Table 1. All the statistical features listed in this table are available for each of the 8 different state definitions.

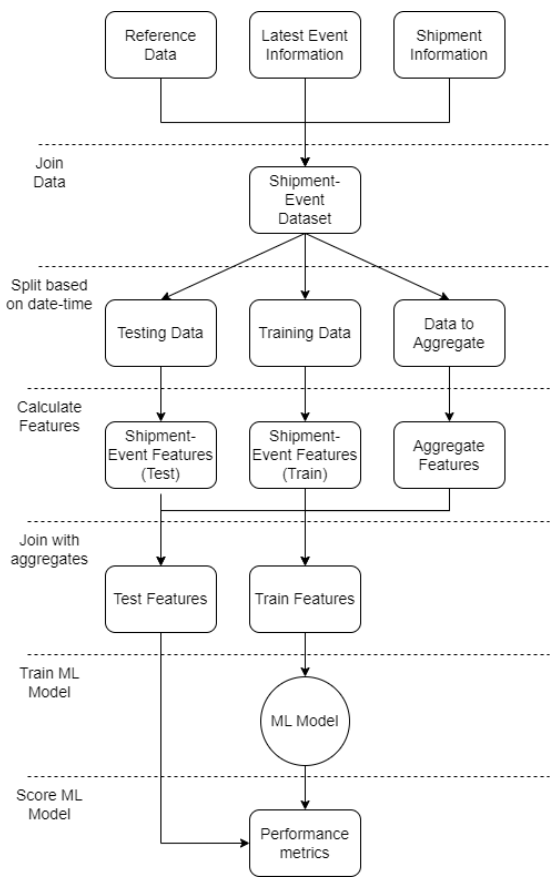

Figure 2. High level representation of the data flow

Table 1. Feature description, type, and number of classes

Feature	Type	# Classes
Shipment features		
Package Weight (KG)/Value (EUR)	Continuous	-
Package Dutiable	Categorical	2
Package Urgent	Categorical	3
Package Domestic	Categorical	3
Package Weight (> 30KG)	Categorical	2
Shipment Type	Categorical	33
Origin Country	Categorical	200
Destination Latitude/Longitude	Continuous	-
Shipment Destination Region	Categorical	5
Shipment Destination Country	Categorical	200
Shipment Destination Facility	Categorical	970
Shipment Destination Remote Area	Categorical	2
Shipment Destination Facility Type	Categorical	5
Key Account Shipment	Categorical	2
State features		
Event Code	Categorical	64
Event Type	Categorical	3
Event UTC Hour/Weekday	Continuous	-
Event Location Latitude/Longitude	Continuous	-
Event Location Facility	Categorical	970
Event Location Country	Categorical	200
Event Location Region	Categorical	5
Event Location Facility Type	Categorical	5
Event Location is Destination	Categorical	2
Event Location is Transit	Categorical	2
Time to Delivery Deadline	Continuous	-
Statistical features (8 groups)		
Min/Mean/Max Time To Arrive	Continuous	-
STD Time To Arrive/Deadline	Continuous	-
Min/Mean/Max Time To Deadline	Continuous	-
Mean Ratio Time to (Arrive / Deadline)	Continuous	-
Max Ratio Time to (Arrive / Deadline)	Continuous	-
Min Ratio Time to (Arrive / Deadline)	Continuous	-
STD Ratio Time to (Arrive / Deadline)	Continuous	-

Target The target feature is the number of days needed for delivery from the prediction moment on. The range of the target value is $\{0, 1, \ldots, 7\}$ where 0 means the same day when prediction is done and 7 the out-of-range class as define business requirements.

In Figure 3 the fractions of data available for each class in our test and train period used for the experiments are shown. It is important to notice that the distributions of target values are often not stable week by week, making the prediction problem more complex due to a lack of the stationarity property. This therefore causes a frequent re-training of the ML models to be needed in order to ensure stable performance. This work will only cover one period training and testing data.

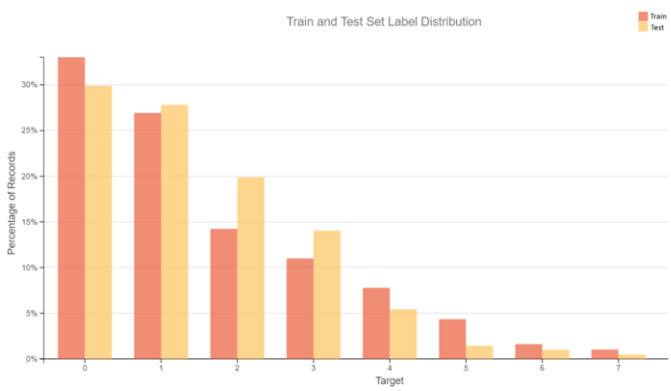

Figure 3. Proportion of data in each class during train and test period when framing the problem as a classification task

4.2 Model development

This section explains the Machine Learning models that are trained and benchmarked to find the one with the best performance in solving the PETAP. In the rest of this section these models are briefly discussed. In our experimentation it is observed that classification models outperformed regression models. Therefore, in the following the explained models are classifiers, unless otherwise specified.

Baseline Model The baseline model is the set of business rules that are defined by our business partner with service quality need (to make competitive offers) and time-zone related limitations.

CatBoost Boosted Decision Tree (BDT) algorithms have been shown to be superior to Artificial Neural Networks (ANN) in an extensive experimentation by [11] using 36 datasets of tabular nature. Additionally an advantage of BDTs is the ease with which they can represent categorical features; these can often just be ordinally encoded to be used by these models with acceptable performance. After testing multiple different implementations of BDTs, CatBoost (for more details we refer to [5]) was chosen due to its superiority when dealing with high dimensional categorical features as well as for its speed during training given by GPU acceleration which enables fast hyperparameters search on large datasets.

Neural Networks with Cat2Vec Deep Learning models are powerful due to their capability of dealing with structured and unstructured data. They are superior to BDTs in prediction tasks with unstructured data, e.g. image, text, speech and so on. These models can be trained in batches without requiring loading the complete dataset (in memory) which turns out to be a great advantage for the ETA prediction task of this paper.

These models, due to their nature, have difficulty representing discrete (categorical) features and hence, leverage these features to make meaningful predictions. Many alternatives ways to represent categorical features can be used ([8] gives a good overview) each having its pros and cons; it is almost always true that these different representations increase the dimensionality of the problem by many times, increasing its complexity.

One technique to represent categorical features in Deep Learning models is Cat2Vec [18]. Cat2Vec employs an unsupervised pairwise interaction embedding strategy aimed at understanding the representation of multi-field categorical data. Analogously to Word2Vec in natural language processing, Cat2Vec generates embedding vectors for individual categorical values. Its innovation consists in calculating these embeddings by learning the latent relationships between pairwise categorical features.

Cat2Vec draws inspiration from Word2Vec, but it is not trained with a sliding window, instead Cat2Vec incorporates intermediate layers to capture pairwise interactions, succeeded by pooling layers, culminating in a fully connected layer to perform a classification task. Consequently, the intermediate layers capture the categorical feature embeddings. Similarly to Word2Vec, Cat2Vec has reduced performance with unseen categories. In the context of PETAP, these unseen categories typically manifest as infrequent combinations of origins and destinations or seldom-encountered products or shipment types.

Transformers for Sequence Classification and Seq2Seq Transformers [13] are a type of Deep ANN architecture capable of reaching high levels of performance in complex tasks such as Natural Language Processing (and Understanding). This type of model is becoming highly popular thanks to the recent advances it brings to conversational agents such as ChatGPT [20].

This type of architecture is particularly powerful when sequence-like data is involved. Our specific dataset when ordered is such a way that all the checkpoints for each package are ordered by time can also be viewed as a dataset of sequences. Each of these would represent the step by step process a package went through from its origin to its destination in chronological order. For the purpose of this paper, we used the same architecture presented in [13], with multi-head attention.

Given the recent developments of this technology we considered relevant to experiment with it, in case the additional information provided by the sequence data could improve the overall predictive performance on the PETAP. Two main ways to use transformers were considered and compared in this work:

- **Classification**, is a sequence to class task, in which the model receives as an input the checkpoints a package received until the current moment and returns the predicted delivery date. In this modeling strategy the relationship between the current state of the shipment and its ETA is only implicitly modeled, hence the model does not have to explicitly show an understanding for the underlying process allowing a package to go from its current state to the predicted ETA state.
- **Seq2Seq**, is a sequence to sequence task, in which the model receives as an input the checkpoints a package received until the current moment and predicts the next most likely checkpoints until delivery. In this strategy, the relationship between the current state and ETA is explicitly modeled. As the model is forced to predict the full checkpoints sequence between the current state and ETA, a deeper understanding of the underlying process might be required, hence a more powerful model could potentially be obtained.

We use two main loss functions; the categorical cross-entropy (CCE) loss function and its ordinal version, since it is commonly used in multi-class classification problems. The CCE loss function quantifies the difference between the true target value and predicted value of data points, and it penalizes incorrect predictions, whereas correct ones are encouraged.

The ordinal categorical cross-entropy (OCCE) loss function is used in ordinal predictive tasks where the target variable has ordered categories (such as how many day will a package be travelling until its delivery).

$$\mathrm{CE}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = -\sum_{i=1}^{K} \boldsymbol{y}_i \log(\hat{\boldsymbol{y}}_i) \tag{1}$$

$$w\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \frac{\left|\arg\max_i \boldsymbol{y} - \arg\max_i \hat{\boldsymbol{y}}\right|}{K-1} \tag{2}$$

$$\mathrm{OCCE} = \left(w\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) + 1\right) \mathrm{CE}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) \tag{3}$$

where $\boldsymbol{y_i}$ denotes the portion of data points under the target class of i, $\hat{\boldsymbol{y}}_i$ the data points with predicted target class i, K the number of classes, and CE the cross-entropy loss function. The re-weighting of categorical cross-entropy happens by means of weight w, that is a balancing factor that penalizes misclassification based on the difference between the indices of the maximum elements in $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, normalised by the maximum value of the index $(K-1)$. If the index of the prediction is coincident with the actual target, then this formula will just behave like categorical cross-entropy. Otherwise the categorical cross entropy loss is reweighed, depending on how far the prediction is from the real target. The rationale of this loss, for the case of PETAP, is that a prediction that misses the target for many days is worse than a prediction that is closer to the target.

5 Experiments and Results

Numerical experiments involved training the different ML models mentioned above on a global train dataset of 2 million samples and testing the obtained model on a (global) test set (from a later time period) made of 10 million data points. Different train and test sets were obtained from the same period by uniformly sampling the original dataset made of 1.2 billion lines (30 days of data) in order to cross-validate the numerical results. The accuracy metric was used as the main way to evaluate model performance. For completeness more metrics such as precision and recall will also be presented in the results tables. A detailed review of different classification metrics can be found in [6].

The train and test set size were chosen via numerical experiments aimed at verifying that:

- Model performance does not substantially change by increasing the train set size
- Performance on a (large enough) test set remains within +- 1% of real performance, measured on the full test dataset (300 million lines).
- Experiment execution time is minimized, ensuring a higher number of numerical experiments can be run in a short time.

Hyper-parameters optimization experiments were conducted by using the hyper-parameter optimization suite *Optuna* [1], allowing for efficient search for optimal model hyper-parameters.

CatBoost In this work hyperparameter search of CatBoost classifier entailed finding the optimized values of number of trees, tree max-depth, learning rate, and L2 regularization weights. *Learning rate* parameter controls the weight of each tree added in training process, and *L2 regularization* weights work for generalizability by preventing over-fitting via discouraging of complex trees.

Numerical experiments show that the tree max-depth and number of trees are the most influential parameters for performance. In particular, the number of trees needs to be high enough to ensure the whole problem space is covered and modelled by the ensemble. Moreover the depth of the trees has to be kept small to avoid overfitting behavior. As these two parameters are related as they both control the complexity of the model, it is also possible to see in Table 2 that inverting the above statements (hence using less trees which are deeper) can sometimes bring to similarly (good) performance. In this contribution, CatBoost was also used to test (i) whether framing the problem as regression could help in improving accuracy as well as (ii) what is the impact of the historical aggregate features on performance. Results for these experiments can be found in Table 3.

Num. Trees (300;2500)	**Max Depth** (1;9)	**Learning Rate** (0;1)	**L2 Reg** (0;1)	**Accuracy**
2000	3	0.31	0.07	0.781
1200	5	0.17	0.28	0.774
1050	7	0.19	0.69	0.770
505	3	0.10	0.23	0.766
900	4	0.01	0.01	0.751

Table 2. Selected explanatory experiment instances for CatBoost Model Hyperparameters Tuning

Cat2Vec Artificial Neural Networks were tested in order to verify whether any performance improvements could be gained. The categorical features were replaced with their corresponding embeddings

calculated using a Cat2Vec network, resulting in a 600 dimensional vector for the categorical features and an additional 119 for the numerical features. Dimensions of all layers in the network are shown in Figure 4, a dropout layer was also used to avoid over-fitting. The Ordinal Categorical Cross Entropy loss function was used to train the model.

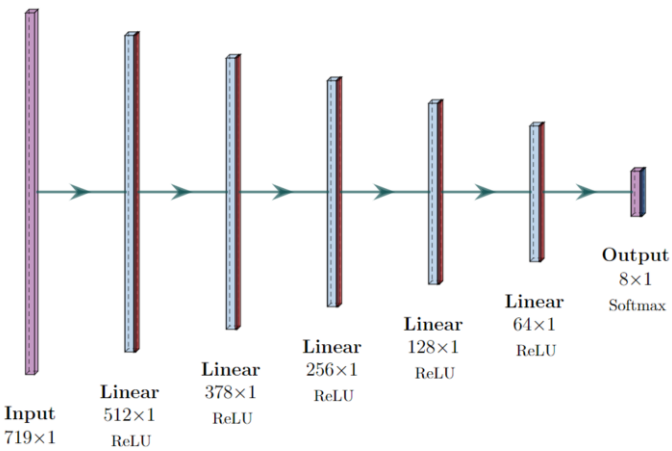

Figure 4. Architecture of Cat2Vec MLP

Transformers The original dataset is extended with the partial route information for each data point. A sequence is defined for every data point and it includes all progress scans from the pickup location till the current one. The initial datasets were expanded to include for each data-point (checkpoint of a shipment), all the previous checkpoints the specific shipment received. This creates a vector of features of size $n\_features * n\_previous\_checkpoints$ for each original data-point, generating new datasets for this experiment.

The finetuned Transformer architecture consists of 2 encoder and decoder layers, a hidden dimension of 1024 and 8 attention heads. Finally, as with the Cat2Vec MLP, the Ordinal Categorical Cross Entropy loss function was used to train the model. When adding past history of the shipment we noticed no accuracy improvement is obtained when compared to a tabular data representation. Basic transformers architecture are hence not suitable to solve this problem due to its complex spatial-temporal nature.

	prec	recall	f1	roc_auc	acc
Catboost Classifier	0.513	0.490	0.496	0.728	0.781
Catboost Regressor	0.422	0.417	0.416	0.685	0.697
Catboost No Agg. Features	0.512	0.481	0.489	0.722	0.770
MLPCat2Vec Classifier	0.510	0.490	0.495	0.728	0.786
Transformer Seq2Class	0.406	0.206	0.181	0.555	0.452
Transformer Seq2Seq	0.372	0.170	0.127	0.531	0.389
Baseline	0.333	0.362	0.311	0.644	0.436

Table 3. Performance of the baseline and developed models across all relevant metrics

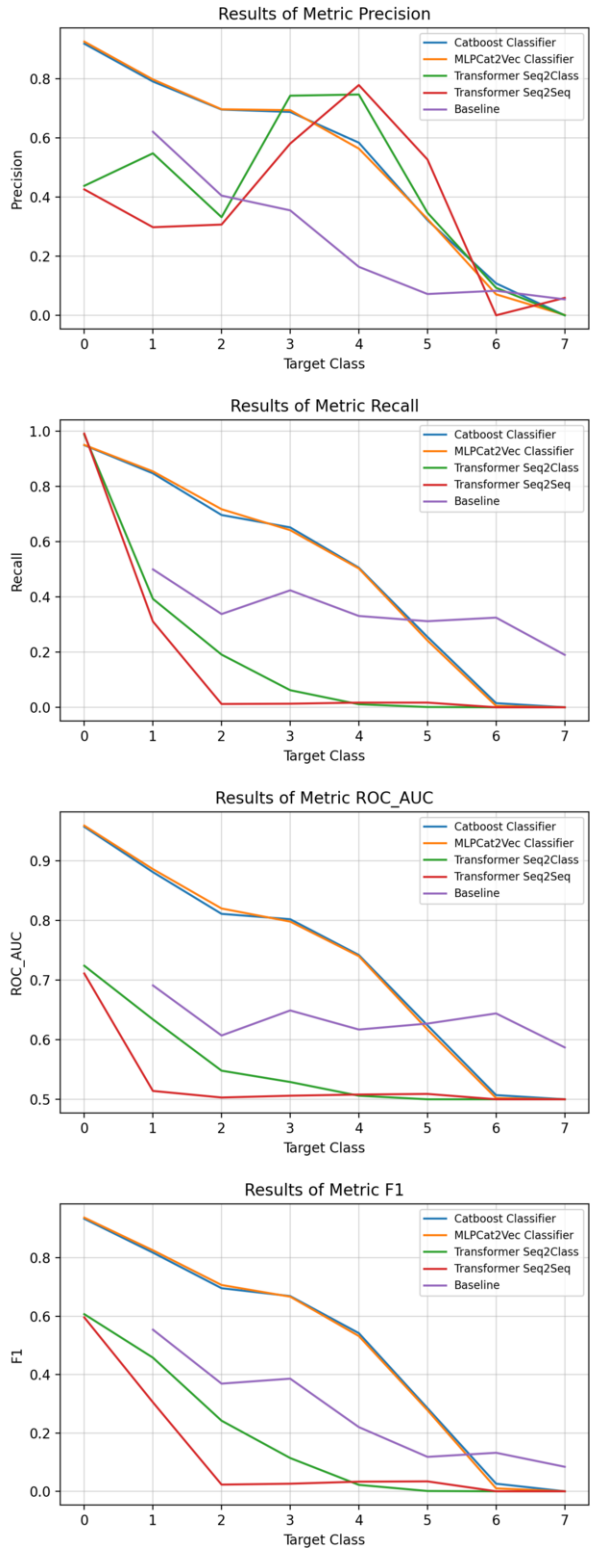

Figure 5. Performance of the baseline and developed models across on different classes on all relevant metrics

5.1 Results

Above described experiments show how models capable of predicting data in tabular format are superior to sequence based models in this problem. In particular both CatBoost and MLP reach a very similar performance above 78% accuracy. Table 3 summarizes the performance of the different models tested.

When using a regression problem setting (rather than classification) the obtained accuracy on predicting the delivery date is inferior. This is explainable due to the discrete nature of business context, in which events (such as package deliveries) happen in a defined time window rather than continuously. When using aggregate features an accuracy gain is obtained.

Figure 5 shows the performance of different models on each class of the problem. We can notice how CatBoost is capable of predicting more accurately than MLP in the most under-represented classes, while MLP is better at predicting dates closer to delivery. Due to the nature of the business problem classes are highly unbalanced, in particular classes '4 days' and above are mainly made of 'outlier' shipments that are either (i) traveling between very poorly connected locations (e.g. remote islands) or (ii) experience some unexpected delay in their transportation. This is why performance on these classes is less relevant than for the [0;3] classes.

Due to the very close accuracy performance these models have, a statistical comparison using the Wilcoxon statistical test (as described in [12]) was performed on these two best performing models (CatBoost and MLPCat2Vec). The test was executed on a dataset of accuracy scores obtained by predicting 500 samples of 20 thousand shipments taken from the test dataset. The statistical test results in a very low p-value indicating a significant difference in the performance of the obtained models on the data samples. This indicates that it could be possible to combine these two ML models into a superior one with techniques such as ensemble learning. Last, Table 4 shows the training and inference speed of each algorithm on Azure Kubernetes Service pods with 64GB of RAM and 4 CPU cores, GPU compute was not available on the side of the business partner due to cost constraints. The results show that CatBoost, thanks to the usage of BDTs, has a very quick training time and inference speed even on CPU, whereas, ANNs have a slow training time and their inference time is several magnitude slower than CatBoost. Albeit it is true that ANNs inference and training time can be improved when using GPUs, the cost associated with constantly running GPUs would not be justified in face of the small performance improvement with respect to ETA that ANNs bring with respect to CatBoost.

Model	Train (min)	Inference Time (sec)
Catboost	39	0.06
MLPCat2Vec	2971	10.49
Transformer Seq2Class	3392	39.67
Transformer Seq2Seq	2998	42.84

Table 4. Summary of model training & inference times, training is measured on a 2 million data-points training set and, inference on a 10 thousand data-points test set.

6 Discussion

We observed that when representing the problem in tabular format, both ANNs and BDTs achieve a comparable performance, but fail to produce an accurate prediction beyond five days. This seems to suggest that it is possible to calculate a distribution of the behaviour the shipment per product and facility, but also that shipments taking more than five days are more rare, and most of these shipments could potentially have hidden issues (e.g. missing the customs paperwork) not easily included in the model. Transformer models, despite using the additional information concerning the checkpoints and sequence of locations visited by the shipment, do not improve over the performance of ANNs or BTDs. This suggests that the nature of the problem at hand is not completely autoregressive, or at least it is not autoregressive with respect to the previous events a shipment received. Hence the journey of a package may have less relevance if compared to independently scheduled (future) events, such as available flights or trucks at a certain location. The result also seems to suggest that the sequence of locations visited by the shipments is only part of the information needed by the transformer. To completely represent the route of the shipment, it would be necessary to consider the times in which a shipment reaches each of the facilities on its route, in addition to information concerning the status of the current and future locations, requiring therefore a deep learning model with an attention mechanism that can tackle spatio-temporal dependencies.

We observe that the sequence representation should be multimodal in order to enhance the prediction model such that it can predict the timing of arrivals to intermediate steps in its route, besides to which location to move next. We believe that designing such a model will be highly challenging and a potential good venue for future research.

The approach used is capable of learning patterns from the provided dataset despite the high complexity of the underlying system which is not being explicitly modeled.

The fact that treating the problem as a regression task produces worse results than treating it as a classification task, suggests that the task at hand has a strongly discrete nature. This aspect seems to be supported by how the network operates. As a matter of fact, the network has a quite synchronous behaviour in its operation, defining specific time slots in which each shipments can depart and origin to leave towards the next step of its travel towards the destination. For example, referring to figure 1 consider a shipment that has to travel between Amsterdam and Recife. Its first flight leg to Miami might only be available every other day, forcing the shipment to wait for at least 24 hours before moving towards its first transit location. Similarly in Miami the shipment might arrive too late to catch a daily flight towards Recife, having to wait yet another day before being moved.

7 Conclusions and Future Work

In this work we developed advanced Machine Learning models to predict the ETA of an express shipment. The problem has several challenging aspects. The most interesting challenges are (i) a very large unbalanced dataset for which the correct sampling size had to be found and, (ii) high cardinality categorical features for which the correct representation strategy had to be used. Several types of ML models were benchmarked, in particular the performance of Multi Layer Perceptron ANNs with category to vector encoding was shown to be slightly superior to that of Boosted Decision Trees (CatBoost) in terms of evaluation metrics but inferior when it comes to training and inference time. Transformers were shown to have an inferior performance to their tabular counterparts on this predictive task when framed as a sequence to class or sequence to sequence task.

Future work on this problem will try to explicitly model the underlying transportation network, ensuring that the ML model predictions are feasible and realistic, especially in transportation cases taking longer than 3 days. Different ways to do this can be explored such as Graph Neural Networks as used in Google Maps [4], Sequence to Sequence models such as [16] or, transport network based modeling such as presented in [2].

References

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902, 2019. URL http://arxiv.org/abs/1907.10902.

[2] A. Balster, O. Hansen, H. Friedrich, and A. Ludwig. An ETA prediction model for intermodal transport networks based on machine learning. *Business & Information Systems Engineering*, 62(5):403–416, Oct. 2020.

[3] T. Cai, H. Wan, F. Wu, H. Wen, S. Guo, L. Wu, H. Hu, and Y. Lin. M2g4rtp: A multi-level and multi-task graph model for instant-logistics route and time joint prediction. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 3296–3308, 2023. doi: 10.1109/ICDE55515.2023.00253.

[4] A. Derrow-Pinion, J. She, D. Wong, O. Lange, T. Hester, L. Perez, M. Nunkesser, S. Lee, X. Guo, B. Wiltshire, P. W. Battaglia, V. Gupta, A. Li, Z. Xu, A. Sanchez-Gonzalez, Y. Li, and P. Velickovic. Eta prediction with graph neural networks in google maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 3767–3776, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3481916. URL https://doi.org/10.1145/3459637.3481916.

[5] A. V. Dorogush, A. Gulin, G. Gusev, N. Kazeev, L. O. Prokhorenkova, and A. Vorobev. Fighting biases with dynamic boosting. *CoRR*, abs/1706.09516, 2017. URL http://arxiv.org/abs/1706.09516.

[6] L. Ferrer. Analysis and comparison of classification metrics, 2023. URL https://arxiv.org/abs/2209.05355.

[7] A. Gulc. Determinants of courier service quality in e-commerce from customers' perspective. *Quality Innovation Prosperity*, 24(2):137–152, Jul. 2020. doi: 10.12776/qip.v24i2.1438. URL https://qip-journal.eu/index.php/QIP/article/view/1438.

[8] J. T. Hancock and T. M. Khoshgoftaar. Survey on categorical data for neural networks. *Journal of Big Data*, 7(1):28, Apr 2020. ISSN 2196-1115. doi: 10.1186/s40537-020-00305-w. URL https://doi.org/10.1186/s40537-020-00305-w.

[9] P. C. Mahajan, A. W. Kiwelekar, L. D. Netak, and A. B. Ghodake. Predicting expected time of arrival of shipments through multiple linear regression. In A. Kumar, S. Senatore, and V. K. Gunjan, editors, *ICDSMLA 2020*, pages 343–350, Singapore, 2022. Springer Singapore. ISBN 978-981-16-3690-5.

[10] N. Marković, S. Milinković, K. S. Tikhonov, and P. Schonfeld. Analyzing passenger train arrival delays with support vector regression. *Transportation Research Part C: Emerging Technologies*, 56: 251–262, 2015. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2015.04.004. URL https://www.sciencedirect.com/science/article/pii/S0968090X1500145X.

[11] D. McElfresh, S. Khandagale, J. Valverde, V. Prasad C, G. Ramakrishnan, M. Goldblum, and C. White. When do neural nets outperform boosted trees on tabular data? In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 76336–76369. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f06d5ebd4ff40b40dd97e30cee632123-Paper-Datasets_and_Benchmarks.pdf.

[12] O. Rainio, J. Teuho, and R. Klén. Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1):6086, Mar 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-56706-x. URL https://doi.org/10.1038/s41598-024-56706-x.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[15] L. Wang, J. Mao, L. Li, X. Li, and Y. Tu. Prediction of estimated time of arrival for multi-airport systems via "bubble" mechanism. *Transportation Research Part C: Emerging Technologies*, 149:104065, 2023. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2023.104065. URL https://www.sciencedirect.com/science/article/pii/S0968090X23000542.

[16] H. Wen, Y. Lin, F. Wu, H. Wan, Z. Sun, T. Cai, H. Liu, S. Guo, J. Zheng, C. Song, and L. Wu. Enough waiting for the couriers: Learning to estimate package pick-up arrival time from couriers' spatial-temporal behaviors. *ACM Trans. Intell. Syst. Technol.*, 14(3), apr 2023. ISSN 2157-6904. doi: 10.1145/3582561. URL https://doi.org/10.1145/3582561.

[17] Y. Wen, T. Chen, J. Wang, and W. Zhang. Pairwise multi-layer nets for learning distributed representation of multi-field categorical data. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, DLP-KDD '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367837. doi: 10.1145/3326937.3341251. URL https://doi.org/10.1145/3326937.3341251.

[18] Y. Wen, T. Chen, J. Wang, and W. Zhang. Pairwise multi-layer nets for learning distributed representation of multi-field categorical data. In *Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data*, pages 1–8, 2019.

[19] F. Wu and L. Wu. Deepeta: A spatial-temporal sequential neural network model for estimating time of arrival in package delivery system. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):774–781, Jul. 2019. doi: 10.1609/aaai.v33i01.3301774. URL https://ojs.aaai.org/index.php/AAAI/article/view/3856.

[20] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136, 2023. doi: 10.1109/JAS.2023.123618.