Probabilistic Prediction of Estimated Time of Arrival

Identifying Sources of Variance and Out-of-Sample Issues

Stanley Förster and Hartmut Fricke
Chair of Air Transport Technology and Logistics
Technische Universität Dresden
Dresden, Germany
stanley.foerster@tu-dresden.de

***Abstract*—The field of trajectory and target time prediction in aviation has long been dominated by point estimation models. However, safety-critical applications using these predictions as it is true in air traffic management should always be in a position to consider such limitations precisely. Therefore, in this paper, three neural network (NN)-based approaches for uncertainty quantification in estimated time of arrival (ETA) prediction are investigated and compared to previous works on the exploration of Quantile Regression Forest (QRF) models. Results show that a mixture density network (MDN) performs slightly better than the QRF model on a dataset covering the Phoenix TRACON obtained from NASA's Sherlock Data Warehouse. The best-performing model is selected based on the continuous ranked probability score (CRPS), which represents a variant of the mean absolute error (MAE) specifically tailored towards probabilistic models. Afterwards, sources of variance in the predictive distributions are investigated. Finally, the often overlooked problem of out-of-sample (OOS) situations is discussed. It is demonstrated, how these can be detected and what the adverse implications on model performance are. The prediction interval coverage probability (PICP) metric indicates massive underestimation of uncertainty is such situations, which is not acceptable for the safety-critical ATM domain and thus will require further investigation.**

***Keywords*—air traffic control, air traffic management, estimated time of arrival, probabilistic prediction, bayesian neural network, mixture density network**

I. Introduction

A scenario ranked as *most likely* in a study conducted by EUROCONTROL's *Challenges of Growth* [1] envisages an increase of air traffic volume to just over 16 million flights until 2040 within the ECAC region. This corresponds to a growth by nearly 53 % compared to the year 2017. Therefore, capacity-enhancing measures will become increasingly important as the forecasting shows that approximately 1.5 million flights cannot be accommodated in 2040, which corresponds to 8 % of the predicted future demand. This capacity gap is equivalent to approximately eight fully booked runways spread across 17 different states in Europe. Even in the most pessimistic scenario (from a growth point of view), a capacity gap of 0.4 million flights or 3 % of the demand is identified as excessive.

Due to the widespread deficiencies, local infrastructure enhancements (like building a new runway) are no suitable measures to at least tactically close this gap. Additionally, societal support or even acceptance is generally very limited for infrastructure expansion projects. Therefore, solutions are required that primarily target the improvement of procedures and operations, which are perceived as much less invasive measures. Furthermore, operational measures are mostly location-agnostic and thus can easily be implemented in various places, which in turn allows spreading improvements in capacity and its utilization more evenly across the whole Air Traffic Management (ATM) system. Such measures include new or adapted traffic management concepts, like free flight or dynamic airspace sectorization, refinement of procedures (e.g. reduced separation minima), but also the development of related sophisticated decision support tools for air traffic controllers (ATCos) and the flight deck crew for improved utilization of the given capacity.

However, this often requires anticipatory capabilities regarding the short- and mid-term evolution of at least parts of the ATM system, like flight trajectories. Hence, support tools need to employ algorithms for making predictions of this evolution and to actually provide additional benefit. A reliable prediction of future system states allows for early intervention and counteracts the need of inefficient and costly last minute actions like holdings, path stretching and shortening, or speed up and slow down, etc. One of the most prominent information that is required in all stages of the flight cycle for efficient aircraft and resource management is the time of arrival at a certain point, for instance the threshold, which is therefore referred to as the *ETA*.

The escalating availability of big amounts of data led to an increased interest in techniques from the field of Machine Learning (ML) to exploit potential that comes therewith.

Consequently, new and improved data-based methods for solving classification and regression tasks evolved in various fields [2]. This inevitable leads to the conclusion that the application of algorithms employing such techniques should be evaluated also in the context of ATM — like for predicting aircraft trajectories or arrival times — especially in the light of current and upcoming challenges when tackling capacity shortcomings.

Observations can only ever cover a fraction of all possibilities regarding inputs, outputs, and influencing factors of a process. Hence, also a data-based model of the process under consideration is subjected to some kind of simplification that originates from the limited availability of data samples (i.e. $N < \infty$). Consequently, any model can only give an estimate of the data the original process would generate, not the real output. This results in uncertainties about the predictions made by the model, which are often neglected when employing deterministic methods that provide so-called *point estimates* only. Therefore, the goal and contribution of this paper is to develop and present a model for probabilistically estimating the time of arrival that generates predictive distributions, which allow indicating uncertainty in the outputs.

II. Literature Review

As outlined previously, uncertainty quantification is gaining attention in the machine learning community, especially in the context of safety-critical domains such as aviation [3]. Most research in trajectory prediction focuses on point estimates, not considering uncertainties inherent to the flight process and other parts of the ATM system. However, awareness for the importance of uncertainty quantification is increasing also in the aviation community.

Over a decade ago, Ren and Clarke acknowledged the need for probabilistic description of aircraft spacing due to uncertainties along the approach segment [4, 5]. Glina, Jordan, and Ishutkina proposed a Quantile Regression Forest (QRF) model for predicting estimated time of arrival (ETA) [6]. Though, the model is still evaluated based primarily on the mean absolute error (MAE) metric, which is suitable for point estimates only and thus gives no information about quality of probabilistic prediction. Similarly, in [7], a model for average daily departure delay prediction is presented. Even though, authors use Dropout [8] for providing prediction intervals with each estimation, the model's performance is evaluated primarily using deterministic metrics like root mean squared error (RMSE) and MAE. Dropout is also used in [9], but no model evaluation is provided. Another work aiming at probabilistic prediction of take-off weight and speed intent (Mach and calibrated air speed (CAS)) is presented in [10]. Here again, the performance of all three methods presented is evaluated based on deterministic RMSE metric, only.

Zhang and Mahadevan propose to train, per state parameter (latitude, longitude, altitude, and velocity) multiple neural networks [11]. Model evaluation is performed based on deterministic metrics MAE, and RMSE, only.

In [12], a Gaussian Mixture Model (GMM) is presented that allows predicting arrival times for a series of upcoming waypoints along a specific air route concurrently. Model performance is evaluated solely on the deterministic part of the prediction (i.e. the mean) using the RMSE metric. The probabilistic part of the prediction is not evaluated.

Rivas, Vazquez, and Franco present a probabilistic trajectory prediction model with focus on estimating fuel consumption [13]. Since authors fix the final mass of an aircraft and calculate the path in reverse, this method is applicable in ex-post analyzes only. In [14], the same group refines the approach, introducing variants of the Beta distribution in addition to the previously employed uniform distribution for characterizing wind uncertainty.

Instead of directly predicting a trajectory, there also exist some indirect methods, in which specific aircraft performance parameters get predicted, which are then used in a physics model. In [15], the author proposes an ensemble model that calculates values for the mean and variance parameters of a Gaussian distribution to obtain information about uncertainty of aircraft mass and speed (Mach and CAS). For evaluation of the model, besides the deterministic RMSE metric, the prediction interval coverage probability (PICP) is employed for individual prediction intervals. Subsequently, a GMM for predicting mass and speed parameters with focus on the climb phase is proposed [16]. The predicted distributions are not analyzed in detail.

Zoutendijk and Mitici recognize the need for probabilistic prediction of departure and arrival delay and compare two models (mixture density network (MDN) and Random Forest (RF)) on a strategic level [17]. In the model evaluation phase, authors provide deterministic RMSE, and MAE metrics, but also investigate the continuous ranked probability score (CRPS) metric, which is tailored to probabilistic models. Results show that, overall, the RF model performs slightly better than the MDN model when predicting delay several days in advance.

Assuming an arrival time prediction method is already available that provides ETAs, Tielrooij et al. propose a method to complement this with information about the uncertainty of the prediction obtained from historic data [18].

Another approach for uncertainty quantification in flight time prediction is presented in [19, 20]. The variance of the flight time is modelled as function of variance of ground speed, which in turn is expressed as a function of cross- and tailwind, Mach number, as well as temperature and their respective variances. In [21] the authors refine the model to account for correlations between wind and temperature data. It could be shown that this adaption improved model performance for long-range flights (above 200 km) significantly.

III. Methodology

This paper extends on the work published in [22, 23], where the initial idea of employing QRF for probabilistic prediction of time-to-fly was presented. Here, alternative probabilistic prediction models are evaluated that target disadvantages of the previous model. The main drawback of QRF is its rather poor scalability in terms of memory, since for computing the quantiles, it is required to store all the training data within the tree's leaf nodes. Furthermore, the output is always an empirical distribution whose resolution depends on the training data. To tackle these disadvantages, three alternative neural network (NN) based model architectures are investigated. These have been selected based on their capabilities to support further ideas that will be investigated in future work but which are hardly achievable using decision-tree based model architectures (e.g. convolutional techniques and handling of time-series).

A. Bayesian Neural Networks

Bayesian neural networks (BNNs) integrate Bayesian inference techniques with neural network architectures to provide a way for uncertainty estimation. Unlike traditional neural networks that output point estimates, Bayesian neural networks (BNNs) allow for probabilistic predictions by treating network parameters θ (i.e. weights and biases) as random variables and leveraging Bayesian methods.

Following Bayes' rule, exact Bayesian inference refers to the process of deriving a posterior distribution $p(\theta|\mathbf{x})$ from a prior distribution $p(\theta)$ under some evidence (i.e. data) $\mathbf{x}$ and likelihood $p(\mathbf{x}|\theta)$:

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\theta)p(\theta)}{\int_\Theta p(\mathbf{x}|\theta)p(\theta)\mathrm{d}\theta} \tag{1}$$

Bayes rule can now be employed to calculate a predictive distribution, which represents the actual probabilistic prediction output of the BNN, as follows:

$$p(\hat{y}(x)|\mathbf{x}) = \int_\Theta p(\hat{y}(x)|\theta)p(\theta|\mathbf{x})\mathrm{d}\theta = \mathbb{E}_{p(\theta|\mathbf{x})}[p(\hat{y}(x)|\theta)] \tag{2}$$

Analytically solving these equations – especially the integral in equation (1) – is intractable, wherefore numeric approximation techniques have been developed, of which two are outlined in the following.

The predictive distribution in equation (2) can be interpreted as an infinite ensemble of networks covering the full parameter space over θ [24]. Approximating this set by a finite sum over m Monte Carlo samples leads to (with $q(\theta) = \frac{1}{m}$):

$$p(\hat{y}(x)|\mathbf{x}) = \sum_\Theta p(\hat{y}(x)|\theta)q(\theta) \tag{3}$$

Instead of aggregating the results into a scalar value as depicted in equation (3), an empirical distribution can be compiled, from which more information, like variance and thus uncertainty of the prediction, can be derived. Alternatively, a proper probability distribution may be fitted to the data.

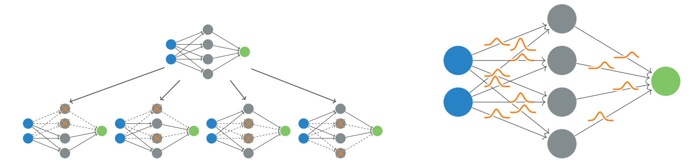
1) Monte Carlo Dropout: One way of generating the model ensemble has been found in the Dropout technique. Initially introduced as a method for regularization that prevents neural networks from overfitting by randomly disabling a set of neurons or connections within the hidden part of a network during training (cf. figure 1a) [25], Dropout can also be employed as a Bayesian approximation technique by enabling this mechanism during inference time [8]. Performing multiple forward passes on the same input leads to an empirical distribution over the output, since in every pass, another set of neurons gets disabled randomly. Dropout can be applied to any NN architecture by introducing an additional hyperparameter that gives the probability of disabling a neuron during a forward pass. In the following, we therefore refer to such a network as Monte Carlo Dropout Network (MCDN). The process of training and optimization is not different from the one used on the classic variant of the underlying NNs.

2) Variational Inference: Another way of creating a virtual ensemble of infinitely many NNs is to represent internal parameters by probability distributions instead of scalars. This idea is depicted in figure 1b. Before every forward pass, the actual value of each parameter is drawn from the corresponding distribution. Consequently, on each inference run, the network's weights and biases differ, leading to a different output. Again, performing Monte Carlo sampling leads to an empirical distribution over the model's outputs.

Since trainable parameters of the NN are represented by probability distributions, the training process has to be adopted accordingly. For the model presented in this paper, the Bayes by Backprob (BBB) method has been implemented [24].

B. Mixture Density Network

Approximate Bayesian inference methods suffer from a significant drawback, as they are computationally expensive since they require multiple runs of a network with varying configurations, i.e. Monte Carlo sampling, to generate enough samples to derive meaningful statistics about the target variable. To be analytically tractable, the resulting



(a) A neural network with Dropout applied in the hidden layer, resulting in a virtual ensemble of networks.

(b) A Bayesian neural network with weights represented as probability distributions.

Figure 1: Schematics of two neural network architectures for Bayesian inference to generate probabilistic predictions.

empirical distribution needs to be fitted with a continuous function. However, an alternative approach called MDN offers a sampling-free solution to these shortcomings, since a single forward pass is sufficient for generating a predictive distribution in analytical representation [17]. Figure 2 visualizes the general concept.

It is important to highlight that MDN represents a special kind of artificial neural network (ANN) [26]. Instead of producing a point estimate output, an MDN maps an input vector $\mathbf{x}$ onto the parameters of a GMM. These parameters are the mixture weights π_k, the means μ_k, and the variances σ_k of the K GMM components. The resulting probability function is then given as:

$$p(y|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) \mathcal{N}\left(y; \mu_k(\mathbf{x}), \sigma_k(\mathbf{x})\right) \tag{4}$$

To obtain a valid GMM from the model's output, special care has to be taken regarding weights and variance parameters. Since the weights of the mixture model have to satisfy the condition $\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$, the output layer of the respective part of the network usually resembles a *soft-max* layer [27], which enforces this condition via:

$$\pi_k = \frac{e^{w_k}}{e^{\sum_{k=1}^{K} w_k}} \tag{5}$$

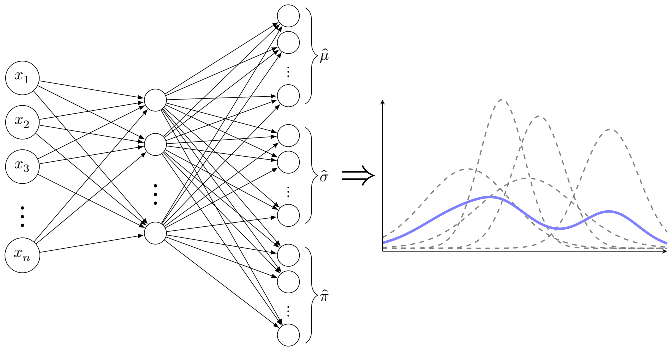


Figure 2: General architecture of a mixture density network. The network's output nodes provide mean μ and standard deviation σ parameters of Gaussian distributions (depicted in dashed gray lines). Using weights π, these are aggregated into a weighted sum (solid blue line).

Furthermore, the variance parameters have to satisfy $\sigma_k > 0$. To ensure this, a common practice is to represent those as the exponential of the raw outputs v_k: $\sigma_k = e^{v_k}$.

Training of the network is guided by a negative log-likelihood (NLL) loss function, which has to be adapted accordingly for a GMM. The loss of a single forward pass is computed as:

$$\mathcal{L}_i(\theta|\mathbf{x}) = -\log\left(\sum_{k=1}^{K} \pi_k(\mathbf{x}_i) \mathcal{N}\left(y_i; \mu_k(\mathbf{x}_i), \sigma_k(\mathbf{x}_i)\right)\right) \tag{6}$$

Based on this, the loss for a full batch of N samples is computed as

$$\mathcal{L}(\theta|D) = -\frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i(\theta|D) \tag{7}$$

C. Model Evaluation Scores

1) Continuous Ranked Probability Score: The CRPS allows assessing how well a probability distribution mimics a discrete value by measuring the area between the curves of the cumulative density function (CDF) for a probability distribution $\hat{F}_i$ and the CDF for a delta distribution (denoted by the Heaviside step function H) located at y_i [28]. It can be expressed as the integral over the Brier score for infinitely many groups [29]:

$$CRPS(\hat{F}, y) = \int_{-\infty}^{\infty} (\hat{F}(u) - H(u-y))^2 \, \mathrm{d}u \tag{8}$$

with

$$H(v) = \mathbb{1}_{v \geq 0} = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{else.} \end{cases} \tag{9}$$

For point estimates, where $\hat{F}_i$ is essentially also a delta distribution located at $\hat{y}_i$, it collapses to the MAE. The CRPS has the same dimension as the response variable, which makes it easy to interpret.

2) Prediction Interval Coverage Probability: A common metric to assess the overall performance of a model in terms of its uncertainty estimation of outputs is the PICP. It is only applicable on a dataset containing multiple samples and is not suitable for scoring individual outputs. The PICP measures the share of samples for which a given prediction interval covers the true value of the response variable. For a prediction interval with lower bound L_i and upper bound U_i, the PICP is defined as [30]:

$$PICP = \frac{1}{N}\sum_{i=1}^{N} c_i \tag{10}$$

with

$$c_i = \begin{cases} 1 & \text{if} \quad y_i \in [L_i, U_i] \\ 0 & \text{else.} \end{cases} \tag{11}$$

Ideally, the value of PICP would be equal to the associated interval width $\alpha_i = U_i - L_i$. However, from equation (10) follows that a high score can be achieved with wide intervals, which would in turn result in the uncertainty being overestimated regularly.

D. Data Description

The flight track data used in this study has been obtained from NASA's Sherlock open data warehouse [31]. Sherlock provides historic air traffic flight, weather, and traffic flow-related data for research in the ATM domain. The data stock comprises integrated National Air Space (NAS)-wide data of 20 Centers, 26 Terminal Radar Approach Controls (TRACONs), and 30 airports. Besides, data from individual facilities can be downloaded. There are three types of flight-related data available: Track points (IFF files), Flight Events (EV files), as well as Flight Summary (RD files). The latter contains meta information about each flight, like origin, destination, departure runway, arrival runway, aircraft type, departure and arrival runway threshold crossing time, and much more.

For the presented research, daily track data (IFF) and flight summary data (RD) has been downloaded for Phoenix TRACON (P50) for the months June, July, and August 2022. Furthermore, data for December 2022 has been downloaded to demonstrate and investigate out-of-sample (OOS) issues. Weather information has been obtained in form of Meteorological Aerodrome Report (METAR) reports for Phoenix Sky Harbor International Airport (KPHX).

During the first step of preprocessing the data, flight track and summary files have been merged. Subsequently, all flights that arrived at KPHX have been selected. Then, each trajectory has first been resampled to a 5-second update interval and then trimmed so that it extends from entering the TRACON to crossing the landing threshold. Figure 3 visualizes a sample of trajectories within the selected airspace.

For each track point, the most recent METAR report has been identified and attached to the corresponding trajectory record. Additionally, minor feature transformations have been applied, like transforming aircraft and wind velocity information from speed / track tuple into corresponding (u, v) vectors. Finally, to handle the issue of cyclic time information, this has been transformed into vector representation, as well. The full list of features used for model development and evaluation is depicted in table I..

As usually, the full data set has been partitioned into a train, validation, and test subset, respectively. Training data is used for optimizing internal model parameters (i.e. weights and biases for NN models, and splits for the QRF model) by minimizing a corresponding training loss functions. The validation data then helps to identify the best model configuration in terms of hyperparameters. Since these two data sets have been used for finding the optimal parameter setting for a certain model variant, these cannot serve in the process of identifying the best model, overall. Here, the test data set is applied. Additionally, this will be used for model evaluation as presented in section III-C.

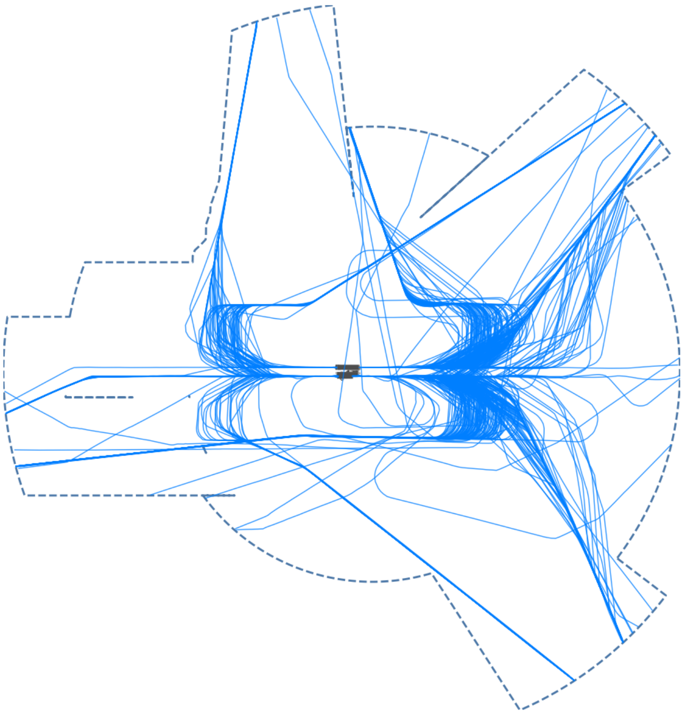


Figure 3: Sample of tracks from June 5, 2022 that depict arrivals to Phoenix Sky Harbor International Airport and the surrounding PHX Class B airspace area.

Table I.: List of features extracted from data obtained from NASA's Sherlock Data Warehouse and which are used for model training and evaluation.

Feature	Description
Day of the week	vector on a circle with $2\pi \mathrel{\widehat{=}} 7\,\text{d}$
Hour of the day	vector on a circle with $2\pi \mathrel{\widehat{=}} 24\,\text{h}$
METAR decay	seconds since last METAR update
x / y / altitude	position of aircraft
u_{AC} / v_{AC}	velocity vector of aircraft
u_{Wind} / v_{Wind}	velocity vector of wind
temp	temperature from the most recent METAR
press	pressure from the most recent METAR
WC	weight class (H - Heavy, F - 757, L -Large, S - Small, T - Small+, U - Unknown)
PC	performance category (J - Jet, T - Turbo Prop, P - Prop, U - Unknown)

Table II.: Summary of the data used for model development and evaluation.

Dataset	Date Range	Sample Count
Training	2022-08-01 – 2022-08-16	954,056
Validation	2022-08-17 – 2022-08-19	170,259
Test	2022-08-22 – 2022-08-28	426,962
Out-of-Sample	2022-12-01 – 2022-12-07	514,957

To prevent any leak of information from train and validation datasets into the test dataset, first the splitting has been performed in temporal order, and second, a 2-day (one weekend) purging gap has been introduced. A summary of the data sets is given in table II..

IV. Results

A. Model Evaluation

After training and tuning each model variation on its own and selecting the best-performing configuration using the validation data set, the independent test data set has been employed to identify the best performing model. The results are summarized in table III.. As can be seen from the third column, the QRF and MDN models show comparable performance over the test data set. This finding is in line with the results presented in [17], where authors compared an RF and an MDN model for predicting delay several days in advance. While in their work the decision-tree model performed slightly better than the MDN, here the latter is superior to all other variants investigated. Hence, this one will be used in further investigations presented below.

Finally, the PICP metric for probabilistic model evaluation will be presented briefly. Figure 4 depicts a plot of the PICP curve for the test data set in blue. It falls little below the 45-degree bisector line ($y = x$), which indicates that the model is underestimating the uncertainty of the outputs slightly.

B. Sources of Variance

After having a look into the results of the specific approaches for probabilistic predication, here it is investigated, if there is a correlation between certain input features and variance in the prediction. For this analysis, Spearman's rank correlation coefficient is calculated between the input features of the test data set, and the variance associated with the predictive distribution outputs generated by the selected MDN model. The results are depicted in figure 5.

Table III.: Results of the model evaluation. Validation data has been used to identify best in class, the test data allows identifying the best model over all variants (emphasized) independent of data seen during training and hyperparameter optimization.

Model	Validation CRPS	Test CRPS
QRF	18.81	16.59
MDN	18.21	**16.29**
BNN	24.05	21.86
MCDN	19.83	17.38

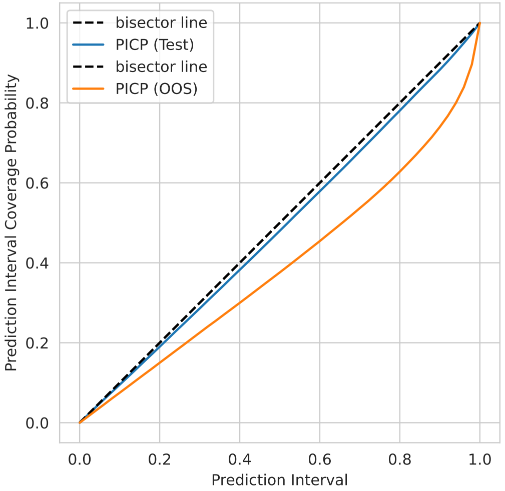


Figure 4: Plot of the prediction interval coverage probability (PICP) curve for the selected MDN model over test, and OOS data set, respectively.

The highest correlation with variance can be found with altitude and speed of the aircraft. This is expected, since aircraft move higher and faster the further away from the threshold they are, which implies a comparatively long remaining flight path and time to landing and thus a high look-ahead time (LAT), which usually is associated with higher uncertainty. No other input variable shows a significant correlation.

Specifically, there seems to be no correlation between atmospheric features and variance. Concluding, strong wind conditions or other off-nominal conditions do not appear to influence the uncertainty of the prediction. However, some values for weight classes and performance categories hint that these might also be worth to investigate further.

C. The Out-of-Sample Issue

A phenomenon that is often overlooked but crucial in safety-critical domains is the out-of-sample (OOS) issue[1], which results from bad generalization capabilities of data-based models [35].

Since any ML model can be conditioned only on a limited set of data during their development, data-based prediction methods are subjected to the effect of receiving inputs

[1] Other terms are out-of-distribution (OOD) [32], dataset shift [33], and domain shift [34]

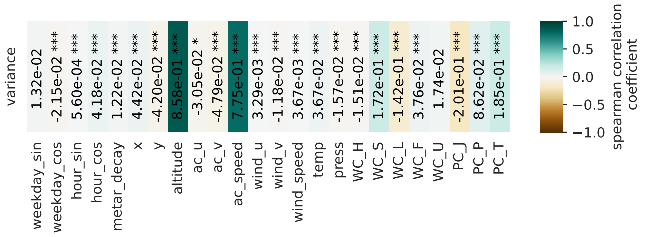


Figure 5: Spearman's rank correlation coefficients for all input features. Weight class and performance category are one-hot encoded, so there is one column for each possible value of the corresponding categorical variable.

that are not reflected in the training data set, particularly during inference time. For instance, a model that has been trained on data obtained through summer season is biased to some extent and may therefore perform much worse when confronted with inputs from a data set primarily covering winter season.

While point estimation models are especially incapable of indicating such cases, also probabilistic models are susceptible to underestimation of uncertainty for OOS inputs [16, 33, 36]. Instead, reliable, robust, and trustworthy models need to be able to detect such issues and consequently output a reasonably high uncertainty.

Quantification of the difference between two distributions P and Q can be achieved using the Kullback-Leibler (KL) divergence, an asymmetric distance measure, i.e. $KL(P \parallel Q) \neq KL(Q \parallel P)$ [37]. The Jensen-Shannon (JS) divergence [38] is a symmetrization of the KL divergence. Furthermore, while the latter has no upper bound, an important property of the JS divergence is that it is bounded in the interval $0 \leq JS(P \parallel Q) \leq \log 2$ [39]. It is defined as follows:

$$JS(P \parallel Q) = \frac{1}{2}\left(KL\left(P \middle\| \frac{P+Q}{2}\right) + KL\left(Q \middle\| \frac{P+Q}{2}\right)\right) \tag{12}$$

with

$$KL(P \parallel Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \tag{13}$$

The following figure visualizes the JS divergence between the data set used for training (P), as well as the test Q_1 and an intentionally selected OOS data set (Q_2), respectively:

Applying the model selected in the previous section to this data, leads to a massive underestimation of uncertainty in the predictive distribution output. This effect is indicated by the orange PICP curve depicted in figure 4.

However, when deploying a model in an operational environment, it is often not practical to first collect a bunch of data that is suitable to derive a distribution usable for comparison with the training data. Instead, the model itself has to become instantaneously aware of OOS situations for every single inference it performs and give an appropriate indication, whether it is processing in- or out-of-distribution data and thus, when an output can be trusted and when it cannot [33].

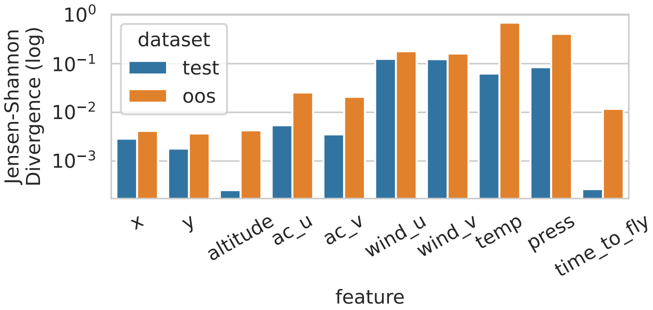


Figure 6: Jensen-Shannon (JS) divergence for the test and OOS datasets with respect to the training data set.

V. Conclusion

In this work, four different ML algorithms have been evaluated against predictive quality, all aiming to perform probabilistic prediction of the estimated time of arrival, this continuously from entry into the TRACON until touch down at a large international airport.

The PICP curve shown in figure 4 indicates a slight underestimation of uncertainty in the predictive distribution outputs, which is usually not acceptable for a safety-critical application as focuses in this paper. Here, a conservative model that rather overestimates uncertainty and thus may lead to increased safety margins better fits that purpose. Therefore, this issue should to be addressed explicitly in the system design process, for instance by developing and employing training loss functions that penalize uncertainty underestimation much higher than overestimation to guide the model into an appropriate direction. For applications in other areas, like for aircraft turn-around scheduling activities, the current behavior can still be acceptable.

Furthermore, sources of variance in the predictive distribution have been investigated. It could be shown that correlations exist between high variance and features that indicate high LAT, like speed and altitude. No other significant correlation could be found between input features and variance, except for weight class and performance category, where a weak correlation might exist.

Finally, the often neglected OOS issue has been discussed. Especially in safety-critical domains, it is crucial to have these effects in mind and invest effort into the identification of such situations. In this work, the issue has been demonstrated. Ongoing work will aim at enhancing the model to add the capability for indicating these.

References

[1] EUROCONTROL. “European Aviation in 2040 - Challenges of Growth.” 2018.

[2] C. L. Philip Chen and C.-Y. Zhang. “Data-Intensive Applications, Challenges, Techniques and Technologies: A Survey on Big Data.” In: *Information Sciences* 275 (Aug. 10, 2014). doi: 10.1016/j.ins.2014.01.015.

[3] E. Hüllermeier and W. Waegeman. “Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods.” In: *Machine Learning* 110.3 (2021). Ed. by H. Blockeel. doi: 10.1007/s10994-021-05946-3.

[4] L. Ren and J.-P. B. Clarke. “Separation Analysis Methodology for Designing Area Navigation Arrival Procedures.” In: *Journal of Guidance, Control, and Dynamics* 30.5 (Sept. 2007). doi: 10.2514/1.27067.

[5] L. Ren and J.-P. B. Clarke. “Flight-Test Evaluation of the Tool for Analysis of Separation and Throughput.” In: *Journal of Aircraft* 45.1 (Jan. 2008). doi: 10.2514/1.30198.

[6] Y. Glina, R. Jordan, and M. Ishutkina. “A Tree-Based Ensemble Method for the Prediction and Uncertainty Quantification of Aircraft Landing Times.” In: *10th Conference on Artificial Intelligence Applications to Environmental Science*. New Orleans, LA, USA, Jan. 2012.

[7] T. Vandal et al. “Prediction and Uncertainty Quantification of Daily Airport Flight Delays.” In: *4th International Conference on Predictive Applications and APIs (PAPIs)*. Boston, MA, USA, Oct. 2017.

[8] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.” In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML’16. New York City, New York, USA: JMLR.org, June 19, 2016.

[9] Y. Pang and Y. Liu. “Probabilistic Aircraft Trajectory Prediction Considering Weather Uncertainties Using Dropout As Bayesian Approximate Variational Inference.” In: *AIAA SciTech 2020 Forum*. Orlando, Florida, USA, Jan. 2020. doi: 10.2514/6.2020-1413.

[10] M. Uzun and E. Koyuncu. “Data-Driven Trajectory Uncertainty Quantification For Climbing Aircraft To Improve Ground-Based Trajectory Prediction.” In: *Anadolu University Journal of Science and Technology A: Applied Sciences and Engineering* 18.2 (2017). doi: 10.18038/aubtda.270074.

[11] X. Zhang and S. Mahadevan. “Bayesian Neural Networks for Flight Trajectory Prediction and Safety Assessment.” In: *Decision Support Systems* 131 (Apr. 2020). doi: 10.1016/j.dss.2020.113246.

[12] H. Paek, K. Lee, and A. E. Vela. “En-Route Arrival Time Prediction Using Gaussian Mixture Model.” In: *9th International Conference on Research in Air Transportation (ICRAT)*. Sept. 2020.

[13] D. Rivas, R. Vazquez, and A. Franco. “Probabilistic Analysis of Aircraft Fuel Consumption Using Ensemble Weather Forecasts.” In: *7th International Conference on Research in Air Transportation (ICRAT)*. Philadelphia, Pennsylvania, USA, June 2016.

[14] R. Vazquez, D. Rivas, and A. Franco. “Stochastic Analysis of Fuel Consumption in Aircraft Cruise Subject to Along-Track Wind Uncertainty.” In: *Aerospace Science and Technology* 66 (July 2017). doi: 10.1016/j.ast.2017.03.027.

[15] R. Alligier. “Predictive Distribution of the Mass and Speed Profile to Improve Aircraft Climb Prediction.” In: *13th USA/Europe Air Traffic Management Research and Development Seminar (ATM Seminar)*. Vienna, Austria, June 2019.

[16] R. Alligier. “Predictive Joint Distribution of the Mass and Speed Profile to Improve Aircraft Climb Prediction.” In: *1st Conference on Artificial Intelligence and Data Analytics in Air Transportation (AIDA-AT)*. Singapore, Feb. 2020. doi: 10.1109/aida-at48540.2020.9049196.

[17] M. Zoutendijk and M. Mitici. “Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem.” In: *Aerospace* 8.6 (June 2021). doi: 10.3390/aerospace8060152.

[18] M. Tielrooij et al. “Predicting Arrival Time Uncertainty from Actual Flight Information.” In: *11th USA/Europe Air Traffic Management Research and Development Seminar (ATM Seminar)*. Lisbon, Portugal, June 2015.

[19] N. Takeichi. “An Adaptive Model of Flight Time Uncertainty and Its Application to Time-Based Air Traffic Operations.” In: *Aviation, Technology, Integration, and Operations Conference (ATIO)*. AIAA. Atlanta, Georgia, USA, June 2018. doi: 10.2514/6.2018-0423.

[20] N. Takeichi. “Adaptive Prediction of Flight Time Uncertainty for Ground-Based 4D Trajectory Management.” In: *Transportation Research Part C: Emerging Technologies* 95 (Oct. 2018). doi: 10.1016/j.trc.2018.07.028.

[21] N. Takeichi et al. “Development of a Flight Time Uncertainty Model for Four-Dimensional Trajectory Management.” In: *Journal of Air Transportation* 28.3 (July 2020). doi: 10.2514/1.d0185.

[22] S. Förster, M. Schultz, and H. Fricke. “Probabilistic Prediction of Time To Fly Using Quantile Regression Forest.” In: *Proceedings of the 9th International Conference on Research in Air Transportation (ICRAT)*. Sept. 2020.

[23] S. Förster, M. Schultz, and H. Fricke. “Probabilistic Prediction of Separation Buffer to Compensate for the Closing Effect on Final Approach.” In: *Aerospace* 8.29 (Jan. 26, 2021). doi: 10.3390/aerospace8020029.

[24] C. Blundell et al. “Weight Uncertainty in Neural Networks.” May 21, 2015. arXiv: 1505.05424.

[25] N. Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” In: *Journal of Machine Learning Research* 15.56 (2014).

[26] C. M. Bishop. “Mixture Density Networks.” NCRG/94/004. Birmingham, UK: Neural Computing Research Group, Dept. of Computer Science and Applied Mathematics, Aston University, Feb. 1994.

[27] J. Bridle. “Training Stochastic Model Recognition Algorithms as Networks Can Lead to Maximum Mutual Information Estimation of Parameters.” In: *Advances in Neural Information Processing Systems*. Ed. by D. Touretzky. Vol. 2. Morgan-Kaufmann, 1989.

[28] J. E. Matheson and R. L. Winkler. “Scoring Rules for Continuous Probability Distributions.” In: *Management Science* 22.10 (1976). doi: 10/cwwt4g.

[29] H. Hersbach. “Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems.” In: *Weather and Forecasting* 15.5 (Oct. 1, 2000). doi: 10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2.

[30] A. Khosravi, S. Nahavandi, and D. Creighton. “A Prediction Interval-Based Approach to Determine Optimal Structures of Neural Network Metamodels.” In: *Expert Systems with Applications* 37.3 (Mar. 15, 2010). doi: 10.1016/j.eswa.2009.07.059.

[31] M. M. Eshow, M. Lui, and S. Ranjan. “Architecture and Capabilities of a Data Warehouse for ATM Research.” In: *33rd Digital Avionics Systems Conference (DASC)*. Colorado Springs, CO, USA: IEEE, Oct. 2014. doi: 10.1109/DASC.2014.6979418.

[32] D. Hendrycks and K. Gimpel. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks.” In: *5th International Conference on Learning Representations (ICLR)*. Toulon, France, Apr. 2017.

[33] Y. Ovadia et al. “Can You Trust Your Model’s Uncertainty? Evaluating Predictive Uncertainty under Dataset Shift.” In: *33rd International Conference on Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, Dec. 2019.

[34] B. Lakshminarayanan, A. Pritzel, and C. Blundell. “Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles.” In: *31st International Conference on Neural Information Processing Systems (NIPS)*. Long Beach, California, USA, Dec. 2017. doi: 10.5555/3295222.3295387.

[35] M. Uzun, M. U. Demirezen, and G. Inalhan. “Physics Guided Deep Learning for Data-Driven Aircraft Fuel Consumption Modeling.” In: *Aerospace* 8.2 (Feb. 2021). doi: 10.3390/aerospace8020044.

[36] M. Hein, M. Andriushchenko, and J. Bitterwolf. “Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem.” In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, California, USA: IEEE, June 2019. doi: 10.1109/CVPR.2019.00013.

[37] T. M. Cover and J. A. Thomas. “Elements of Information Theory.” Wiley Series in Telecommunications. New York: Wiley, 1991.

[38] J. Lin. “Divergence Measures Based on the Shannon Entropy.” In: *IEEE Transactions on Information Theory* 37.1 (Jan. 1991). doi: 10.1109/18.61115.

[39] F. Nielsen. “On the Jensen–Shannon Symmetrization of Distances Relying on Abstract Means.” In: *Entropy* 21.5 (5 May 2019). doi: 10.3390/e21050485.