aerospace

MDPI

Article

On the Exploration of Temporal Fusion Transformers for Anomaly Detection with Multivariate Aviation Time-Series Data

Bulent Ayhan *, Erik P. Vargo and Huang Tang

The MITRE Corporation, McLean, VA 22102, USA; evargo@mitre.org (E.P.V.); htang@mitre.org (H.T.)
* Correspondence: bayhan@mitre.org

Abstract: In this work, we explored the feasibility of using a transformer-based time-series forecasting architecture, known as the Temporal Fusion Transformer (TFT), for anomaly detection using threaded track data from the MITRE Corporation's Transportation Data Platform (TDP) and digital flight data. The TFT architecture has the flexibility to include both time-varying multivariate data and categorical data from multimodal data sources and conduct single-output or multi-output predictions. For anomaly detection, rather than training a TFT model to predict the outcomes of specific aviation safety events, we train a TFT model to learn nominal behavior. Any significant deviation of the TFT model's future horizon forecast for the output flight parameters of interest from the observed time-series data is considered an anomaly when conducting evaluations. For proof-of-concept demonstrations, we used an unstable approach (UA) as the anomaly event. This type of anomaly detection approach with nominal behavior learning can be used to develop flight analytics to identify emerging safety hazards in historical flight data and has the potential to be used as an on-board early warning system to assist pilots during flight.

Keywords: aviation; flight; time-series; forecasting; anomaly detection; transformers

Citation: Ayhan, B.; Vargo, E.P.; Tang, H. On the Exploration of Temporal Fusion Transformers for Anomaly Detection with Multivariate Aviation Time-Series Data. *Aerospace* **2024**, *11*, 646. https://doi.org/10.3390/aerospace11080646

Academic Editor: Jules Simo

Received: 13 June 2024
Revised: 25 July 2024
Accepted: 3 August 2024
Published: 9 August 2024

1. Introduction

In the last two decades, safety risk management in civil aviation has shifted from post-accident investigations and analyses to proactively identifying emerging safety hazards and incorporating the analyses of these proactive findings to supplement post-accident investigations. This type of shift requires applying new approaches that can process a large amount of multivariate time-series aviation data from various data sources that can be both time-varying and categorical. Identifying anomalous events within historical flight data is crucial for the extraction of various safety hazards. These hazards can be associated with various factors, including adverse weather conditions (such as heavy rain and strong winds), mechanical failures, human error (pilot, air traffic controller), airspace congestion, inadequate flight planning, ground operations, difficult terrain, and bird strikes, among others. The standard anomaly detection technique applied to aviation data is to use exceedance detection methods [1]. These methods require domain knowledge and involve comparing specific flight parameters with respect to aircraft-dependent thresholds pre-defined by aviation subject matter experts. Because exceedance-based methods are based on rules with strict thresholds [2], they face limitations when identifying new safety risks and capturing useful information from the acquired multivariate and multimodal flight and environmental data with complex and nonlinear relationships that take place at different temporal scales. Recently, machine learning techniques have been used to fill this emergent gap, and they have been investigated for their potential to automatically identify anomalies in multivariate flight data.

Exploring machine learning techniques to be used with flight sensor data for improving aviation safety is an active research field. Li et al. [3] introduced a cluster-based anomaly detection approach to detect abnormal flights, which can support domain experts in detecting anomalies and associated risks from routine airline operations. Their

approach, "ClusterAD-Flight", used data from the flight data recorder and applied the density-based spatial clustering of applications with noise (DBSCAN) algorithm to perform the cluster analysis to detect abnormal flights of unique data patterns. The authors, pointing out the need for predefined criteria or domain knowledge as a shortcoming of existing anomaly detection techniques, stated that ClusterAD-Flight no longer required these. Li et al. further extended their approach in [4] and named their extended approach "ClusterAD-DataSample". In this extended approach, a Gaussian Mixture Model (GMM)-based clustering is applied to digital flight data to detect flights with unusual data patterns, with the assumption that normal flights share common patterns while anomalies do not. The authors stated that, in comparison to ClusterAD-Flight, which can make decisions about whether the take-off or approach phase as a whole is abnormal or not, ClusterAD-DataSample can detect instantaneous abnormal data samples during flight. The authors noted that with their approach, airline safety experts can identify latent risks from daily operations without specifying what to look for in advance.

L. Basora, X. Olive, and T. Dubot provided a survey of data-driven anomaly detection approaches and their application to the aviation domain in [5]. Some of these approaches included machine learning techniques such as clustering-based approaches and advanced autoencoders. Following this survey, two authors, X. Olive and L. Basora, introduced a reconstruction-based anomaly detection technique using autoencoders [6]. Their technique was to detect and identify significant events in historical aircraft trajectory data. For flight data, the authors used Automatic Dependent Surveillance–Broadcast (ADS-B) trajectory data since it is often more accessible than aircraft data. The authors investigated the trajectory anomaly scores computed by autoencoders for significant operational events such as re-routings or deconfliction measures and found that the highest anomaly scores corresponded to poor weather conditions, while anomalies with a lower score related to Air Traffic Control (ATC) tactical actions.

The National Aeronautics and Space Administration's (NASA) Ames Research Center has generated tools for data mining and machine learning methods for aviation safety, such as Multiple Kernel Anomaly Detection [7]. Another software tool from NASA, the Automatic Discovery of Precursors in Time-Series, is based on finding precursors using multidimensional time-series data and has been applied to flight anomalies such as missed approach [8] and take-off stall [9].

Gavrilovski et al. [10] surveyed data-mining techniques in the aviation domain and provided a review of published work. Janakiraman [11] introduced an approach that combines multiple-instance learning and deep recurrent neural networks for weakly supervised learning problems that involve time-series flight data. Martinez et al. [12] introduced a methodology that performs a precursor analysis and a binary classification using Gradient Boosting frameworks and analyzes Flight Data Monitoring (FDM) temporal series with Long Short-Term Memory (LSTM) deep learning techniques. The authors stated that the aircraft speed, flap positions, altitude, rate of descent, and meteorological conditions of the destination airport were the most relevant precursors, and the investigated deep learning technique provided better forecasting performance.

Wang et al. [13,14] used surveillance track data and wind data to build and improve a forecasting model based on Logistic Regression for predicting unstable approach (UA). The authors demonstrated that by adding more features, the prediction performance can be improved. Ackley et al. [15], developed a methodology that is based on supervised machine learning techniques to train a model for classifying time-series flight data into safety events and non-safety events in the approach and landing phases. Time-series digital flight data obtained from historical commercial aviation operations is used to train a model and identify critical feature subsets and event precursors directly related to elevated levels of flight risk for commercial aircraft. Bleu-Laine et al. [16] introduced a methodology that leverages high-dimensional aviation data to predict multiple adverse events and discover their precursors. Their methodology used a deep learning model that consists of Convolutional Neural Networks (CNN) for each sensor data type to predict adverse events

and determine the precursors to the predicted adverse events. Recently, an autoencoder architecture has been used for anomaly detection with time-series flight sensor data, which utilizes the reconstruction error as the anomaly score [17,18]. Variational autoencoders (VAE) have also been investigated for their potential in anomaly detection [1].

To better characterize flight behavior and emerging safety risks, in addition to time-series data from various aircraft flight sensors, there is a need to consider several other environmental and operational parameters in the time-series data model, such as runway identifier, runway status, airport traffic, weather, wind, visibility, temperature, etc. These parameters can be time-varying and can also consist of categorical data. In the case of cascading aircraft failures, such as sensor readout differences, it is difficult to detect and characterize these precursor events through threshold-exceedance monitoring alone before a catastrophic failure happens. A novel detection paradigm is therefore needed that can monitor the states of multiple aircraft variables and correlate these states with the nominal or anomalous conditions of the aircraft through mathematical models trained with multivariate and multimodal flight data.

In this paper, we introduce a forecasting-based anomaly detection approach that uses multivariate aviation time-series data with the Temporal Fusion Transformer (TFT) architecture [19]. We show how a TFT model trained on nominal multivariate time-series data can be used for anomaly detection. For anomalies, we used flights that experienced a UA. We used Fisher's discriminant classifier [20] to demonstrate that the TFT model trained with nominal flight data is sensitive to UA flight data and can predict the temporal locations of UA during the approach phase of the landing. The contributions of this paper are as follows:

(a) Explored the feasibility of the TFT architecture with multivariate aviation time-series data for anomaly detection via nominal behavior learning.
(b) Demonstrated that the trained TFT forecasting models for nominal behavior are sensitive enough to detect anomalous flight time-series sequences, such as UA, and indicate the temporal locations of the anomaly.
(c) Showed the feasibility of training a single TFT model to forecast multiple outputs for anomaly detection.

The paper is organized as follows: In Section 2, we describe the anomaly detection approach and summarize background information about the TFT. The multivariate time-series flight data used in this research and sourced from The MITRE Corporation's TDP threaded track [21] and digital flight data [22] are also introduced in this section. In Section 3, we present the conducted investigations and summarize the results. In Section 4, we discuss our findings and address potential future work. Finally, in Section 5, we state our conclusions.

2. Materials and Methods

2.1. The TFT Architecture

The TFT [19] forms the backbone of the forecasting model used to learn nominal flight behavior in this work. The TFT architecture consists of neural networks that enrich learned temporal representations with static covariate information and combine self-attention with recurrent connections to capture both local temporal patterns and long-term temporal dependencies. The TFT is known for its multi-horizon forecasting capability on multivariate time-series data. The inputs into TFT can be time-varying features and time-independent static covariates (which provide contextual metadata about features). The TFT architecture is shown in Figure 1.

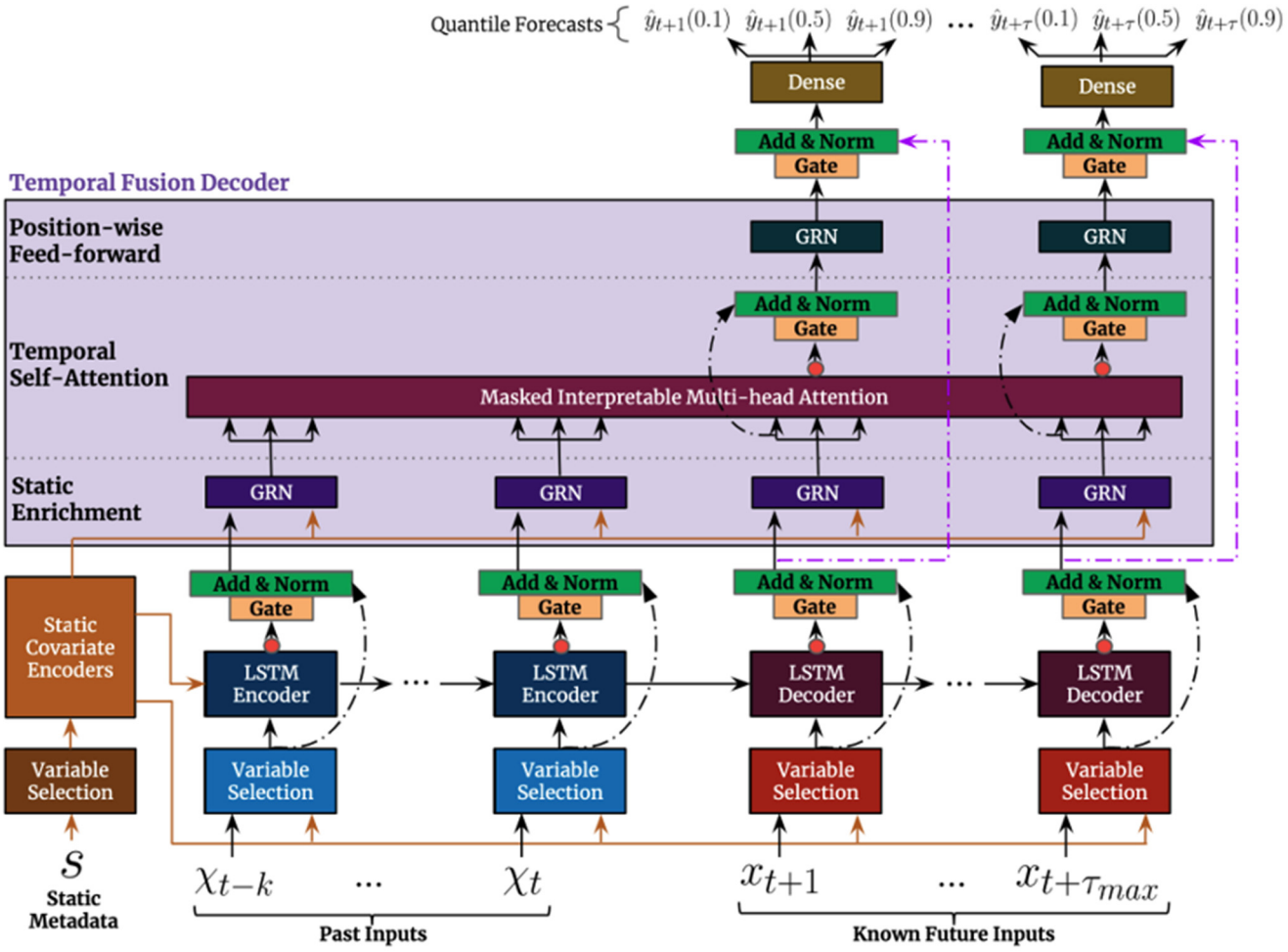

Figure 1. Temporal Fusion Transformer (TFT) architecture. Reproduced from [19].

Gating mechanisms are extensively used in the TFT architecture to skip unnecessary processing layers. These mechanisms are named Gated Residual Networks in the architecture. Their use provides flexibility in suppressing nonlinear contributions that are not needed. Variable Selection Networks (VSNs), which are used to select relevant input features at each time step, are considered for each type of input (known and unknown past inputs, known future inputs, and static covariates). The VSNs help with the interpretability of the TFT since the global importance weights of the input features are explicitly learned through these blocks. The context vectors from the static covariate encoders are injected at several parts of the TFT architecture to integrate static information and conditional temporal dynamics. The TFT enables short-term temporal relationship learning using sequence-to-sequence layers through LSTM blocks. The long-term dependencies are captured using a modified multi-head attention block by employing different heads for different representation spaces. This enables TFT to learn different temporal patterns while attending to a common set of input features to enhance explainability. In addition to point forecasts, the TFT generates quantile forecasts at each time step. A quantile loss function is used, which consists of terms for both upper and lower quantiles. The TFT is trained by minimizing the quantile loss across all quantile forecasts. For technical details of the TFT architecture and its inner blocks, refer to [19].

2.2. Forecasting-Based Anomaly Detection Approach Using the TFT Architecture

In the introduced anomaly detection approach, rather than training the TFT model to predict outcomes of a specific aviation safety event, such as UA or go-around, we train the TFT to model nominal behavior using nominal flight data. In this way, we

avoid the challenging problem of learning from a limited set of off-nominal samples and better leverage the capacity of the TFT. Because the introduced approach makes predictions into the future about the expected nominal flight behavior given past flight data and examines the differences between the model's flight parameter predictions and the observed flight parameters at future time points to highlight anomalies, it differs conceptually from autoencoder-based anomaly detection, which examines reconstruction error to detect anomalies. A block diagram of the introduced approach can be seen in Figure 2. In this proof-of-concept study, we used only some of the inputs shown in the block diagram in Figure 2 (such as flight track data, wind data) and considered speed and altitude as the two outputs for forecasting. However, the number and diversity of inputs can be extended along with the number of outputs.

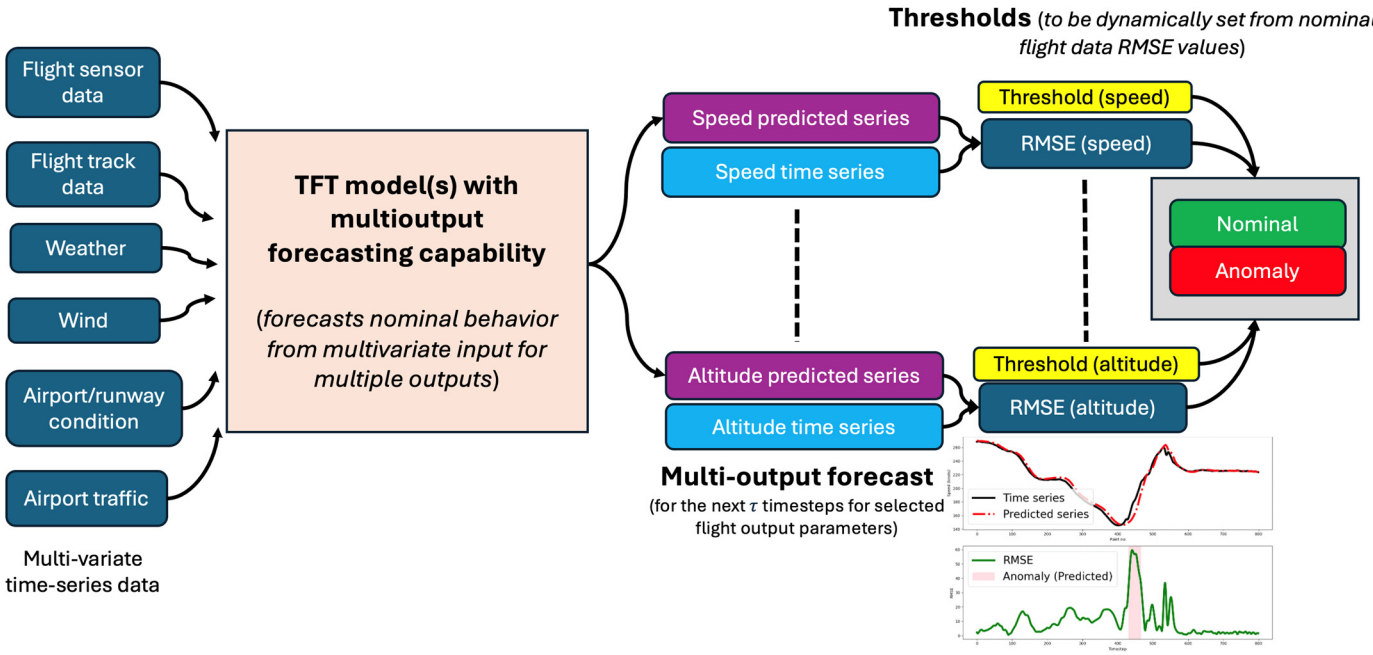

Figure 2. Forecasting-based anomaly detection via nominal behavior learning with the TFTs.

2.3. Data Description

TDP [21] and digital flight data [22] platforms are used to form the multivariate time-series dataset used in this work. The information in the two data sources is merged through a geospatial join using available flight metadata in both. The flight data in the dataset consists of a single wide-body aircraft type arriving at one of the major airports in the United States and belongs to a pre-COVID time frame to capture standard volume and operations. The data are limited to this single airport and wide-body aircraft type to decrease the computation times and to increase the robustness of the trained model in this proof-of-concept study.

Due to data privacy restrictions, we cannot disclose the exact flight counts for the nominal and UA flight data in the generated data splits (train, validation, and test). Instead, we provided the normalized flight counts, where the normalization process corresponds to scaling the raw flight counts in each data split to the flight counts in the train split. A breakdown of the normalized flight counts of the data splits can be found in Table 1. Similarly, we cannot disclose the exact flight data labeling methodology for determining nominal and UA operations; however, many of the energy and configuration criteria commonly applied within the domain [23] were evaluated. The UA labeling process in this work identifies several UA criteria, and flights exceeding these criteria up to an identified percentage are considered as UA. Considering this threshold exceedance-based UA event labeling, some of the UA-labeled flights in this work may not actually correspond to anomalous flights. It is also worth mentioning that the UA event criteria used in this work

may not match the UA criteria used by many flight operators since these criteria vary from one operator to another and are specific to their operations and the risk they have accepted through their safety management system processes.

Table 1. Normalized flight counts after the split generation for nominal flights and UA flights in the data splits.

	Train Split	Validation Split	Test Split
Nominal Flights	1	0.2316	0.0965
UA Flights	-	-	0.0927

The flight data inputs used in this proof-of-concept study consist of "flight" and "weather context" components, as seen in Table 2. The "flight" component can be further divided into time-varying features and static metadata. As an example, a runway identifier is considered static "flight" metadata, whereas altitude is a time-varying "flight" feature. The time-varying "weather context" features in Table 2 pertain to the weather impact at the arrival airport.

Table 2. Flight and weather context inputs.

Index	Flight	Weather Context
1	Latitude	Wind direction
2	Longitude	Wind speed
3	Altitude	Visibility
4	Speed	Wind runway difference
5	Course	Headwind
6	Curvature	Crosswind
7	Acceleration	
8	Climb Rate	
9	Runway Identifier	

To generate our time-series flight data samples, we consider only the final 240 timesteps prior to the wheels down time due to the consideration of UA as the anomaly type. Each timestep is separated by 5 s, for a total duration of 20 min. Each TFT sample consists of 64 timesteps (320 s) of input (look-back input window), followed by 8 timesteps (40 s) of output (look-ahead output window). As such, the first TFT sample's forecast time occurs 64 timesteps into the arrival phase, and the final TFT sample's forecast time occurs 8 timesteps prior to the wheels down time. Consecutive samples are overlapped by 5 s (i.e., one timestep), which results in a total of 169 look-back input window and look-forward output window pairs per flight. Some of these time parameters could certainly be considered as hyperparameters and can be selected through a systematic approach. In this work, the look-back and look-ahead window size selections were ad-hoc, with the preference to have more data points within the look-back window in contrast to a smaller data size in the look-ahead window.

3. Results

Our first two sets of investigations involved analyzing the impact of varying the input features in the TFT model training and output targets (i.e., using single output versus multi-output for forecasting) on the Root Mean Square Error (RMSE) of the prediction. The third set of investigations consisted of generating a quantitative metric to demonstrate that the trained TFT model can differentiate anomalous time sequences from nominal and can identify temporal sections for anomalous behavior. For all considered TFT models, the training split was used to train the TFT via gradient descent using the PyTorch-Forecasting Python package [24]. The validation split was used to decide when to stop training (i.e., when the validation error establishes a local minimum). The test split was reserved

to assess the performance of the TFT on data that the model hadn't seen before. The UA test split was used to assess the feasibility of anomaly detection based on RMSE thresholding. The hyperparameters used for the TFT model training are in Table 3. For technical information about these hyperparameters, refer to [24].

Table 3. Hyperparameters used in the TFT model training.

Hyperparameter	Value
learning_rate	0.03
hidden_size	16
attention_head_size	1
dropout	0.1
hidden_continuous_size	8
output_quantile_size	7
loss	Quantile Loss
reduce_on_plateau_patience	4
epoch_number	200
batch_size	128
gradient_clip_val	0.1

3.1. Using Different Input Features in the TFT Model Training

A total of three different subsets of input feature combinations are considered when training TFT models, as shown in Table 4. The first input feature combination (TFT-1) contains time (time before touchdown), a runway identifier, and eight different flight track features (latitude, longitude, altitude, speed, course, curvature, acceleration, and climb rate). The second input feature combination (TFT-2) contains what's available within TFT-1 plus two additional wind-related features (headwind and crosswind). The third input feature combination contains what's available within TFT-2 plus four additional weather/wind-related features (wind direction, wind speed, visibility, and wind runway difference). Speed is set as the output target. This investigation was mainly to observe which of the input feature combinations would provide lower forecasting errors and assess the impact of feature selection when training a TFT model. Intuitively, we assume that the TFT model with the lowest forecasting error on nominal test flight data would be the best candidate for detecting and discriminating flight data that contains anomalous behavior or events.

Table 4. TFT models were trained with different subsets of input features.

TFT-1	TFT-2	TFT-3
Time	Time	Time
Runway identifier	Runway identifier	Runway identifier
Latitude	Latitude	Latitude
Longitude	Longitude	Longitude
Altitude	Altitude	Altitude
Speed	Speed	Speed
Course	Course	Course
Curvature	Curvature	Curvature
Acceleration	Acceleration	Acceleration
Climb rate	Climb rate	Climb rate
	Headwind	Headwind
	Crosswind	Crosswind
		Wind direction
		Wind speed
		Visibility
		Wind runway difference

The RMSE metric is used to assess the forecasting performance of the TFT models trained with different subsets of input features. Suppose t_i corresponds to a forecast

timestep value for nominal flight data and speed is the targeted output for forecasting. At forecast timestep t_i, the TFT model makes predictions for speed at the look-ahead window timesteps (t_i, $t_{i+1}, \ldots, t_{i+7}$). For prediction, the TFT model uses the input data in the 64-timesteps look-back window (t_{i-64}, $t_{i-63}, \ldots, t_{i-1}$). The RMSE computation at t_i, RMSE(t_i), is mathematically expressed in (1), where $(\hat{y}(t_i), \hat{y}(t_{i+1}), \ldots, \hat{y}(t_{i+7}))$, are the TFT-predicted speed values and $(y(t_i), y(t_{i+1}), \ldots, y(t_{i+7}))$ are the actual observed speed values. In (1), τ corresponds to the size of the look-ahead window, which is set to 8 in this work.

$$\mathrm{RMSE}(t_i) = \sqrt{\sum_{k=i}^{i+\tau-1} \frac{(\hat{y}(t_k) - y(t_k))^2}{\tau}} \quad (1)$$

The RMSE profile of nominal flight data is formed by computing the RMSE values at each forecast timestep value of nominal flight data before touchdown (a total of 169 forecast timesteps). The RMSE profiles are generated for each nominal flight data in the test split. Even though the nominal flight data varies temporally from each other, making it impractical to compare the RMSE values of two separate flights, we averaged the resulting RMSE profiles of the nominal flight data in the test split. This was to identify the TFT model that provided lower RMSE values and to get a rough idea about the temporal locations where lower forecasting errors were observed. Figure 3 shows the averaged RMSE profiles for all the nominal flights in the test split with four TFT models trained to predict the speed of the aircraft. The time before touchdown on the x-axis is the forecast timesteps.

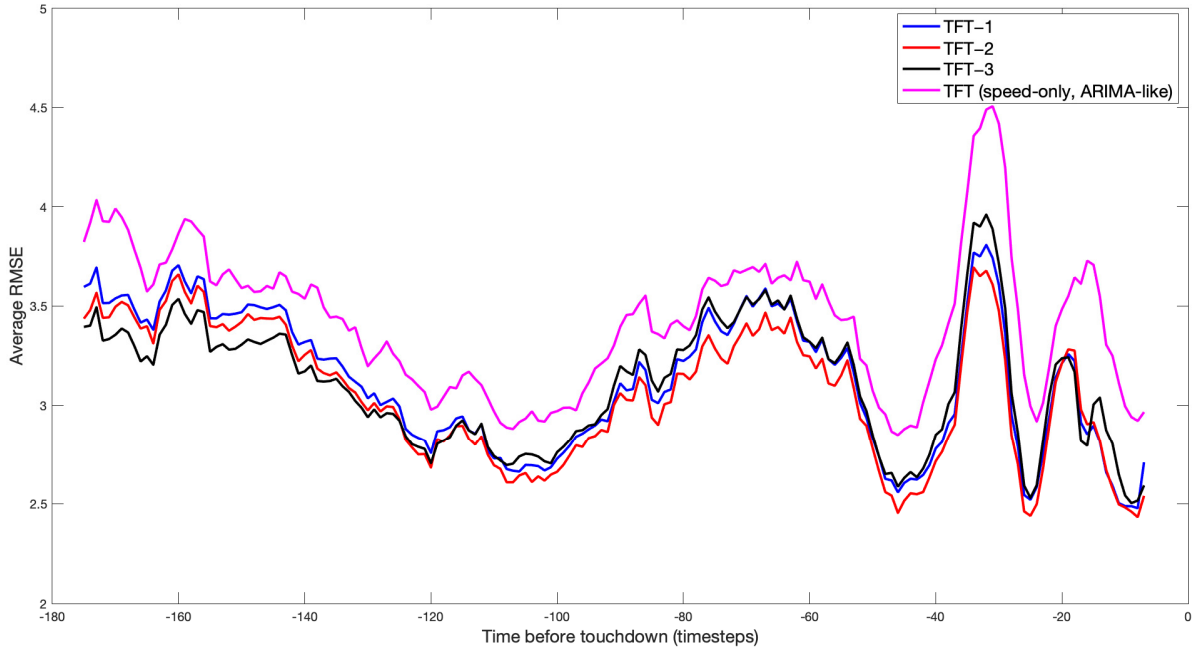

Figure 3. Averaged RMSE profiles as a function of "time before touchdown" resulting from various TFT models for the nominal flight data in the test split with speed as the target output.

Three of these TFT models (TFT-1, TFT-2, TFT-3) correspond to the three different input feature combinations. For the sake of a benchmark comparison, we also included the performance of another TFT model that is trained to predict speed solely as a function of the previously observed 64 timesteps of speed. This single-variate TFT model can be thought of as an Autoregressive Integrated Moving Average (ARIMA) [25] model. Notably, from Figure 3, it can be observed that the ARIMA-like TFT model is the worst performer at each forecast timestep yielding higher forecast errors along the x-axis, which highlights that the TFT's have learned patterns from the multivariate inputs. The mean value of the averaged RMSE profiles (along the whole forecast time-series) is 3.43 for the ARIMA-like TFT model, 3.13 for TFT-1, 3.06 for TFT-2, and 3.11 for TFT-3. Among the other TFT models, it can be

observed that, except for the first 55 timesteps, the TFT-2 model, which is trained with a runway identifier, flight tracks, headwind, and crosswind, provided lower forecasting errors in comparison to the TFT-1 and TFT-3 models, indicating the importance of feature selection in model training.

One interesting attribute of the TFT architecture is that it can learn the global importance weights of input features due to its use of VSNs and provide feature importance rankings. Figure 4 shows the resultant feature importance rankings with respect to the three TFT models when used with the nominal test data split.

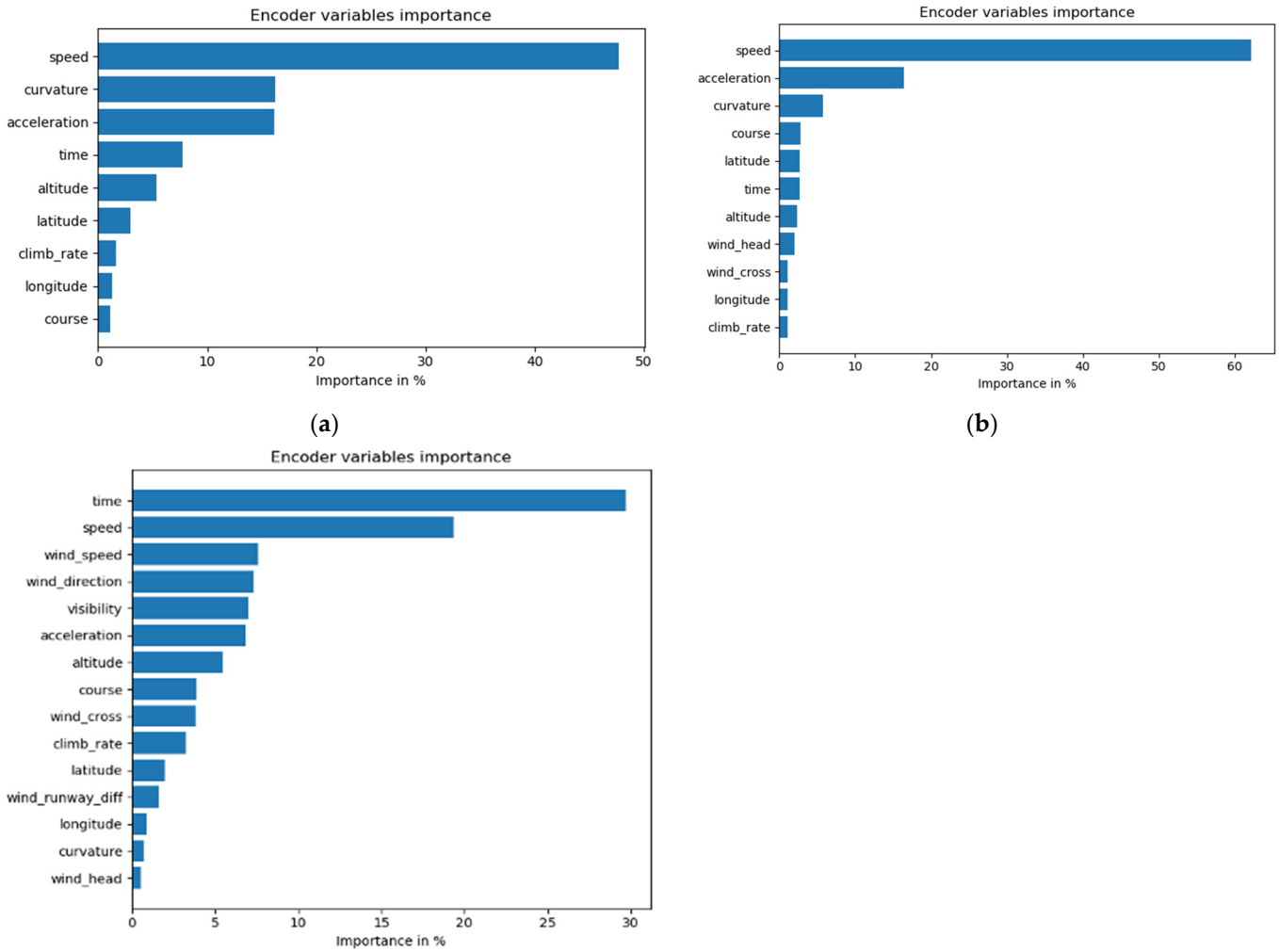

Figure 4. Feature importance rankings for the three TFT models trained with different input feature combinations with speed as the target (**a**) TFT-1, (**b**) TFT-2, and (**c**) TFT-3.

To explore whether TFT's feature importance ranking capability could be used for feature selection for training TFT models that yield lower forecasting errors, we used all but the final four features in the TFT-3 input feature combination (Figure 4c) in order of importance to train a new TFT model in which the number of input features is set the same as TFT-2 (since TFT-2 provided lower forecasting errors). We label this input feature combination as "TFT-select". Figure 5 shows the averaged RMSE profiles obtained with this new input feature combination (TFT-select) and TFT-2. From Figure 5, it is observed that overall, the two sets of averaged RMSE profiles are quite close to each other. The mean value of the averaged RMSE profiles (along the whole forecast time-series) is 3.02 for TFT-select and 3.06 for TFT-2 (it was 3.11 for TFT-3 from Figure 3). Considering the time duration after the first 55 timesteps, TFT-2 provides slightly lower forecasting errors in comparison to TFT-

select, and TFT-select significantly performs better within the first 55 timesteps. Overall, this result shows the feasibility of conducting feature selection using the information from TFT's feature importance rankings instead of a manual feature selection process.

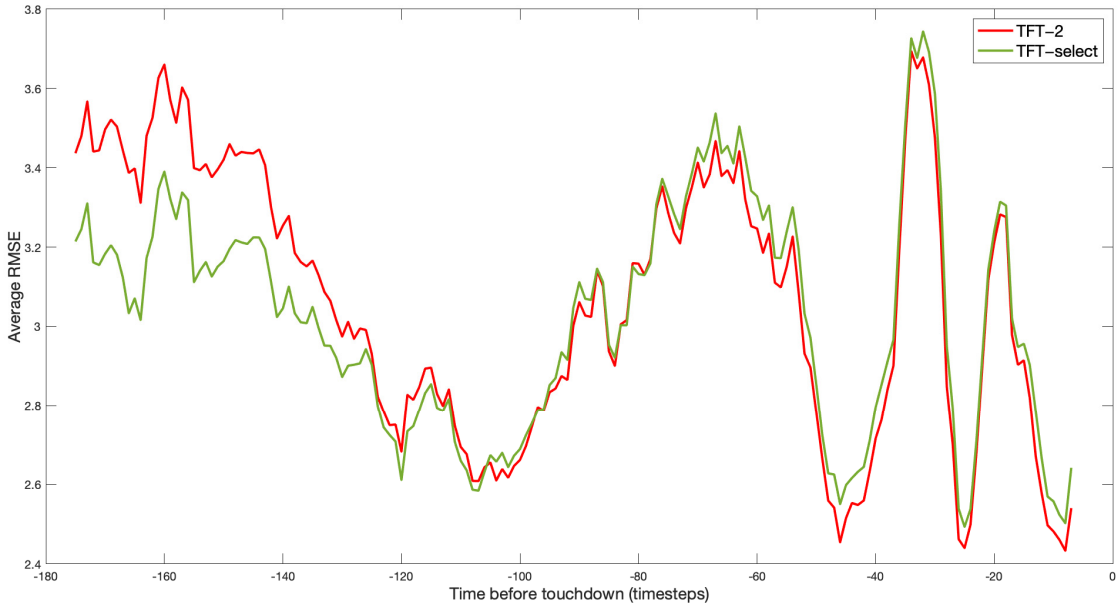

Figure 5. Averaged RMSE profiles for TFT-2 and TFT-select.

A potential automated feature selection process for the TFT model training could thus entail identifying all available features first, followed by a TFT model training using these features. This trained TFT model could then be applied to a nominal flight data set to identify the resultant feature rankings. Based on these TFT-based feature rankings, features that are not deemed as important could be excluded considering the computational constraints (for example, by dropping all features with importance values less than 5%). Finally, a new TFT model could be trained using the selected top-ranked features only, which is anticipated to decrease the TFT model training time while keeping the same forecasting power or perhaps providing even better forecasting performance due to excluding some of the redundant or less important features.

3.2. Single Output vs. Multi-Output Forecasting with the TFT

The TFT architecture allows for training a single TFT model that can forecast multiple outputs at once. This type of capability could reduce computation needs and simplify the data processing pipeline. In this part, speed and altitude were considered as the two output targets, due to their known correlation with UA events, to assess the TFT's ability to predict multiple targets at once. Some of the trained TFT models were considered to jointly predict both speed and altitude, and other models were considered to predict only speed or altitude (but not both). We compared the forecasting error of the single-output and multi-output TFT models on the nominal flight data test split.

Prior to training the multi-output models, the speed and altitude targets are standardized via "min-max" normalization to help balance the contribution of speed and altitude prediction errors in the loss function. Figures 6 and 7 compare the averaged RMSE profiles of the multi-output TFT models trained to jointly predict speed and altitude during the arrival phase for the nominal flight data in the test split. The multi-output TFT models are represented using one of the three subsets of input feature combinations that were introduced in Table 4.

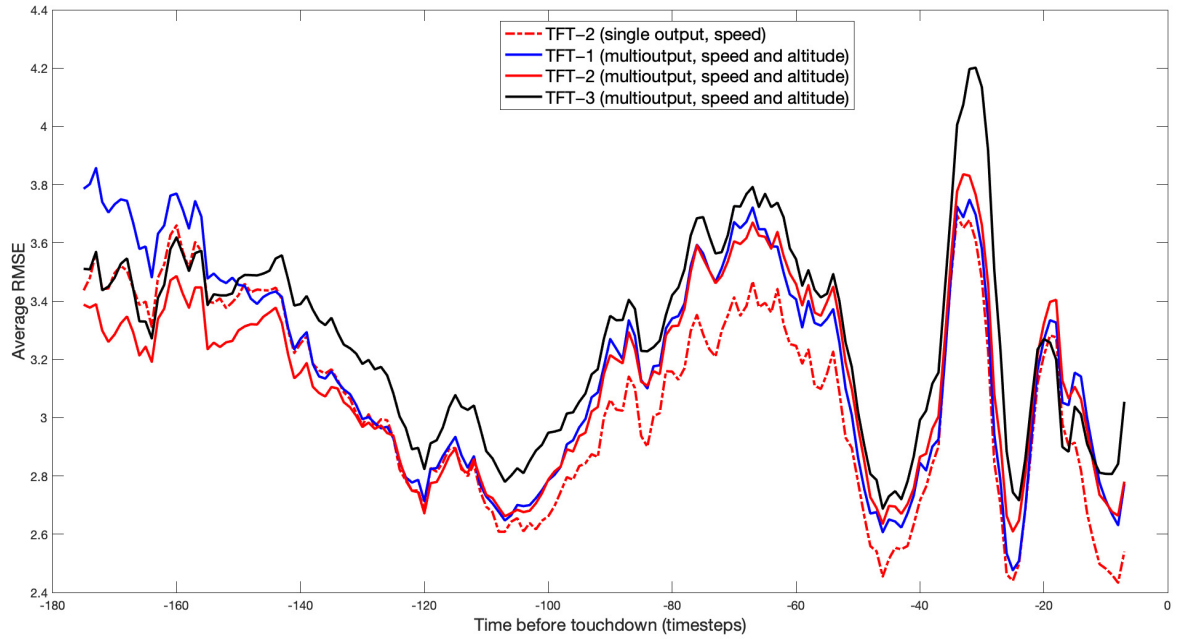

Figure 6. Averaged RMSE profiles from speed predictions for the TFTs trained to jointly predict speed and altitude (multi-output) in comparison to the single-output TFT model (speed).

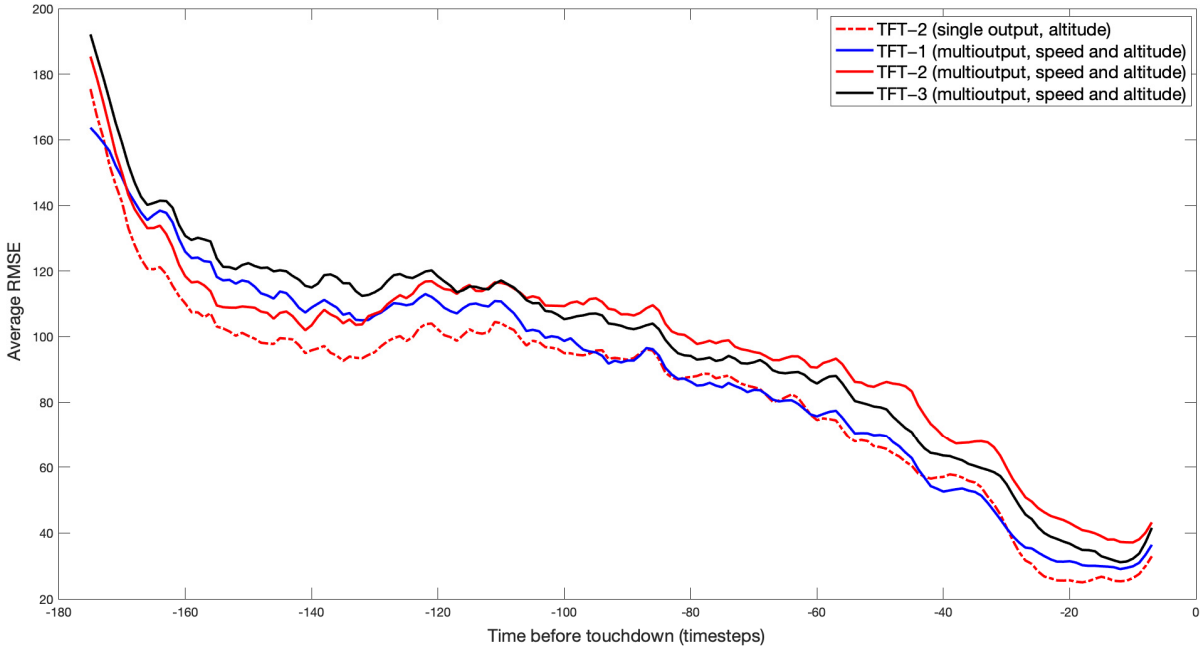

Figure 7. Averaged RMSE profiles from altitude predictions for the TFTs trained to jointly predict speed and altitude (multi-output) in comparison to the single-output TFT model (altitude).

For the sake of comparison, the performance of TFT models trained to predict a single output is provided in each plot as well. In Figure 6, the mean value of the averaged RMSE profiles is 3.06 for TFT-2 (single output, speed), 3.18 for TFT-1 (multi-output), 3.13 for TFT-2 (multi-output), and 3.27 for TFT-3 (multi-output). In Figure 7, the mean value of the averaged RMSE profiles is 83.48 for TFT-2 (single output, speed), 88.72 for TFT-1 (multi-output), 96.65 for TFT-2 (multi-output), and 96.77 for TFT-3 (multi-output). For each of the two targets (speed and altitude), the single-output TFT model is generally superior across the forecast time axis. This is understandable, as the single-output models are trained exclusively to predict a single target, and so we can dedicate the TFT model's entire

forecasting capacity to this one task. Yet, the plots in Figures 6 and 7 demonstrate the TFT's multi-output forecasting capability. With the adoption of an enhanced architecture (adding more layers or increasing the number of network parameters in each layer) and hyperparameter finetuning, improved performance for the multi-output TFT models could be possible.

3.3. Anomaly Detection via Nominal Behavior Learning with the TFT Forecasting Model

For demonstrating anomaly detection with the TFT forecasting model, we make use of both the nominal and UA data test splits to examine their corresponding RMSE values at each timestep and to identify the temporal locations where the UA test split's RMSE values differ from the nominal RMSE values. Even though the temporal locations of UA might differ from one UA-labeled flight data to another, we hypothesize that a significant portion of them should be taking place at a time close to landing. Figure 8 shows the averaged RMSE profiles for both speed and altitude using the TFT-2 single-output TFT models (for the nominal and UA flight data in the test split). The differences between the averaged RMSE profiles of the nominal and UA flight data can be visually noticed in Figure 8 and provide information about the temporal locations where UA events are possibly taking place.

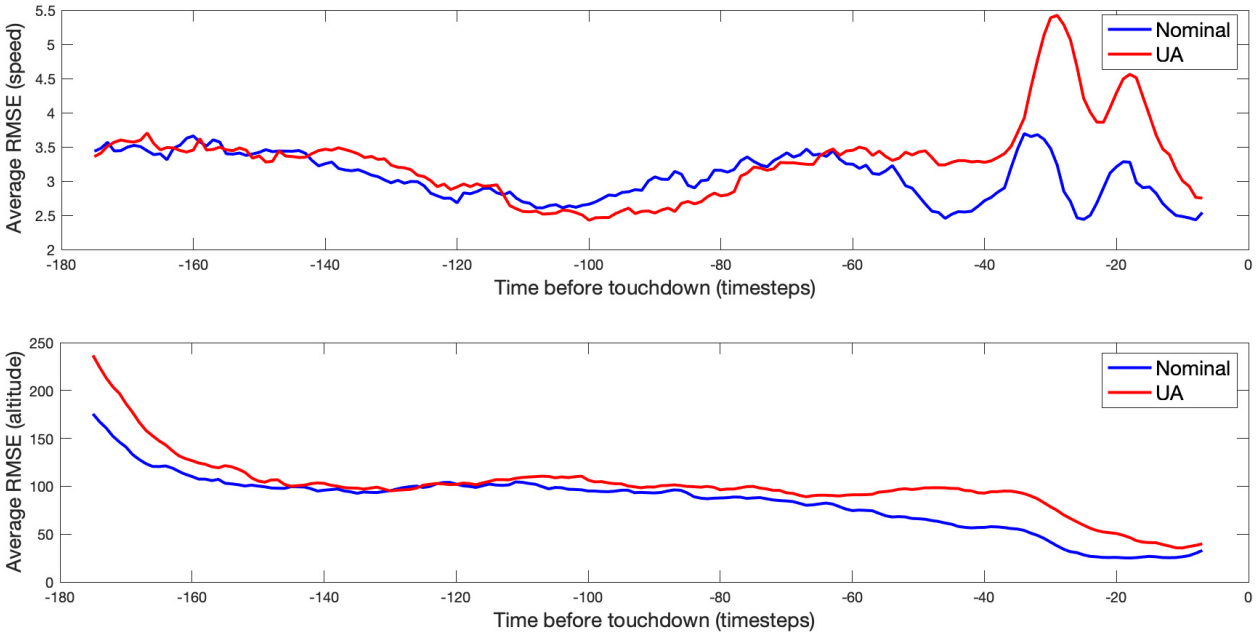

Figure 8. Averaged RMSE profiles for speed and altitude using the TFT-2 single-output models (speed and altitude modeled separately).

We utilized Fisher's linear discriminant [20] to quantitatively show how separable the nominal flight data are from the UA flight data with respect to their RMSE values and to locate the temporal locations where the RMSE values of the nominal flight data differ from the UA flight data throughout the time-series. Fisher's linear discriminant is not used as a classifier here but rather as an auxiliary analysis tool to identify a potential time point candidate along the time before touchdown axis, where the separation between the RMSE values of the nominal and UA flight data is relatively higher. The RMSE values at this identified time point for both the nominal and UA flight data samples are then used in an RMSE-threshold-based anomaly detection setting to demonstrate the feasibility of the forecasting-based anomaly detection.

Fisher's criterion function is mathematically described in (2). In (2), $\mathbf{S}_B$ corresponds to the between-class scatter matrix, $\mathbf{S}_W$ is the within-class scatter matrix, and $\mathbf{W}$ is a transformation matrix that maximizes the ratio of the between-class scatter to the within-class scatter. One of the two classes corresponds to the RMSE values from the nominal

flight data (speed and altitude forecasting errors) and the other class corresponds to the RMSE values for the UA flight data. With $\mathbf{W}$ that maximizes the ratio of the between-class scatter to the within-class scatter, the resulting Fisher's criterion is utilized as a metric to visualize the time instances where the separation between the two classes (nominal and UA) starts to become apparent within the time axis and to examine which of the TFT models (single-output alone, merged single-output, or multi-output) provides higher separation through time.

$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|} \tag{2}$$

Due to the two-class nature of the problem, Fisher's method is not strictly applicable in the single-output case. In the multi-output case where two RMSEs are available, Fisher's method first computes a scalar projection of the joint speed and altitude RMSEs to simultaneously minimize within-class variance and maximize between-class variance. Both in the single- and multi-output cases, anomaly detection can then be conducted by setting a threshold on RMSE that balances true positives against false-positives on the test set. Figure 9 plots the optimal Fisher's discriminant score as a function of "time before touchdown". All test samples available at each timestep were used to compute Fisher's score and optimize the scalar projection in the multi-output cases at each time step. Larger Fisher's scores indicate greater class separation at that timestep and, presumably, greater classification potential when combined with an appropriate RMSE threshold.

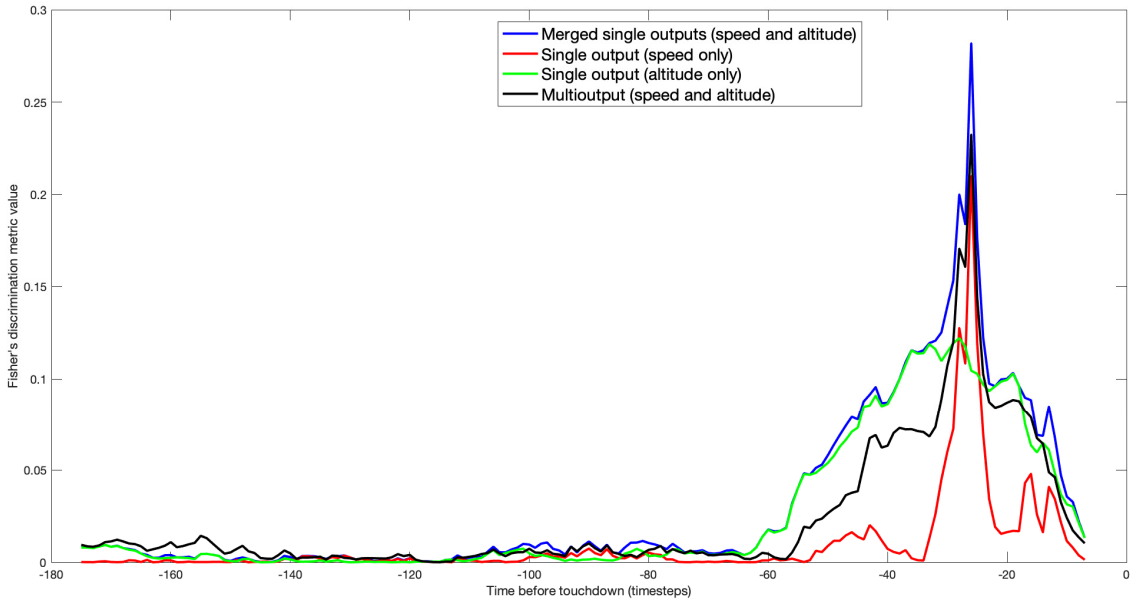

Figure 9. Fisher's score for various TFT models as a function of time before touchdown.

Based on Figure 9, all four TFT models have a maximum Fisher's score around the 26 timesteps before touchdown. Fisher's score is highest when the RMSEs for the single-output speed and altitude TFT models are merged, followed by the multi-output TFT model at that timestep. Figure 9 shows the impact of considering multiple outputs in comparison to a single output for anomaly detection. With speed output alone, the temporal range to differentiate UA from nominal flights is found to be narrow but has a higher separation potential, whereas, with altitude output alone, there is a wider temporal range for differentiation but with a lower separation potential. Merging both outputs yields the best of both worlds, achieving high-value separation and a wide temporal range. The RMSE values of the test split (nominal and UA) from the single output TFT models for altitude and speed prediction (using TFT-2 input feature combination) at 26 timesteps before touchdown can be seen in Figure 10. Higher RMSE values for both speed and

altitude predictions with wide scattering can be observed for the UA flight data in contrast to a more clustered set of RMSE values with smaller magnitudes for the nominal flight data at this timestep.

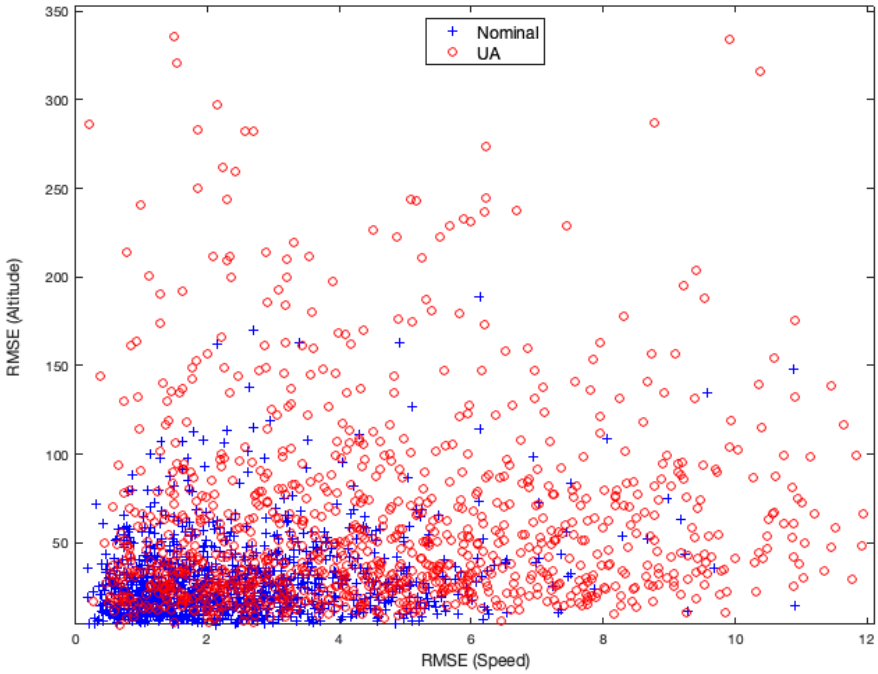

Figure 10. RMSE values of the test split (nominal and UA flight data) for the single output TFT models (speed and altitude outputs) at the 26 timesteps before touchdown—zoomed in for better visualization.

After identifying the time point that yielded a good separation between the RMSE values of the nominal and UA flight data, which corresponded to the 26 timesteps before touchdown in the time-axis, we set RMSE thresholds for both speed and altitude output at this time point to conduct a threshold-based anomaly detection. In this threshold-based anomaly detection, to determine whether test flight data is nominal or UA, the resultant RMSE values of the test flight data for the two target outputs are compared with the set RMSE thresholds. If the test flight data's RMSE value is below the RMSE threshold for the two target outputs, the test flight data is considered as nominal. However, if the test flight data's RMSE value is above the set RMSE threshold for either of the two target outputs, the test flight data are considered as anomalous.

Regarding the threshold settings, suppose an RMSE threshold is set to 5.15 for the speed output and an RMSE threshold is set to 64 for the altitude output at the 26 timesteps before touchdown after examining the RMSE scatter plot in Figure 10 for the two outputs. In doing this, our goal is to get a sense of how the anomaly detection results would look when the RMSE values from the two outputs are used alone and when they are used together. With these thresholds, the resultant confusion matrix (normalized) for speed and altitude alone is shown in Table 5a,b. It can be noticed from the two confusion matrices that with using speed output alone and with using altitude output alone, the false positive (FP) rates are the same at 6.99% with these two thresholds, whereas the True Positive (TP) rate using speed is 38.63% and the TP rate using altitude is 33.33%, indicating that speed is a better output for separating UA from nominal flight data, as was also observed from the Fisher discrimination scores in Figure 9.

When the two outputs are used together for classification such that the class label is assigned to nominal only when the RMSE values for the two outputs are both lower than their assigned thresholds and are assigned to UA for all other cases, the resultant confusion matrix can be seen in Table 5c. It is observed that the TP rate jumps significantly to 56.96% and the FP rate increases to 12.52%. By relaxing the two thresholds slightly (speed threshold increased to 5.85 and altitude threshold increased to 85.0) to get the same

FP rate of 6.99% (to have a fair TP rate comparison), we get the confusion matrix shown in Table 5d. From Table 5d, it is seen that the TP rate becomes 45.46% (while FP is 6.99% like the single output FP values), which is significantly higher than the TP rate of using speed, which was 38.63%.

Table 5. Confusion matrices (normalized) with single- and multi-output RMSE thresholds: (**a**) Speed RMSE alone (speed RMSE threshold = 5.15), (**b**) Altitude RMSE alone (altitude RMSE threshold = 64.0), (**c**) Speed and altitude RMSE together (speed RMSE threshold = 5.15 and altitude RMSE threshold = 64.0), (**d**) Speed and altitude RMSE together (speed RMSE threshold = 5.85 and altitude RMSE threshold = 85.0).

(**a**)			
		Predicted Condition	
		Nominal	UA
Actual Condition	Nominal	0.9301	0.0699
	UA	0.6137	0.3863
(**b**)			
		Predicted Condition	
		Nominal	UA
Actual Condition	Nominal	0.9301	0.0699
	UA	0.6667	0.3333
(**c**)			
		Predicted Condition	
		Nominal	UA
Actual Condition	Nominal	0.8748	0.1252
	UA	0.4304	0.5696
(**d**)			
		Predicted Condition	
		Nominal	UA
Actual Condition	Nominal	0.9301	0.0699
	UA	0.5454	0.4546

We used the Receiver Operating Characteristics (ROC) curve to visualize the detection performance when the set RMSE thresholds for both speed and altitude are changed incrementally. The speed RMSE threshold range is set between 0.25 and 6.55, and the altitude RMSE threshold range is set between 1 and 106, with 37 threshold points in each range with equal intervals. It is assumed that the two identified threshold pairs (5.15, 64) and (5.85, 85) for speed and altitude, which are used in the confusion matrices above, are included as two sets of threshold pairs in the two identified ranges. Figure 11 shows the resultant ROC curve, which indicates that with the use of speed and altitude outputs together, higher detection performance can be achieved in comparison to using a single output, which supports the findings of the confusion matrices.

This result demonstrates the impact of examining multi-output flight parameters for anomaly detection with the TFT nominal behavior forecasting models. It is worth mentioning that while we used a simple RMSE threshold-based classifier for anomaly detection in the proof-of-concept demonstrations, other ML algorithms (e.g., regression trees, support vector machines, neural nets, etc.) could be utilized for higher accuracy.

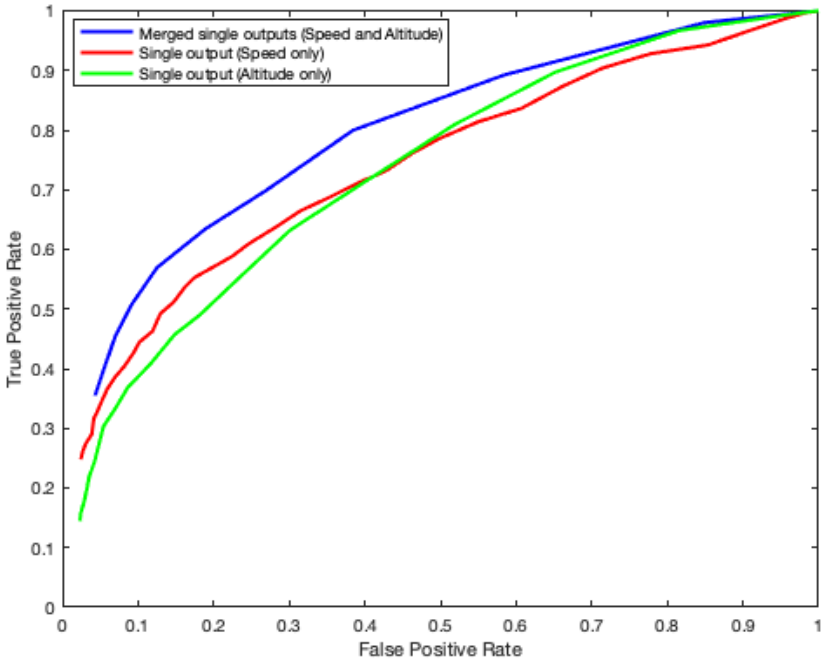

Figure 11. ROC curve for visualizing the detection performance of the RMSE-threshold-based anomaly detection at the identified time point when used with single outputs and two outputs together.

4. Discussion

The multivariate time-series dataset used in this work contains a small subset of digital flight data features and lacks on-board sensor features that would otherwise enable the model to capture more nuanced relationships between inputs and outputs and provide greater insights into safety event precursors. Despite these caveats, our preliminary results suggest that the TFT is an effective way to summarize multivariate time-series aviation data.

In this initial proof of concept effort, we did not perform a hyperparameter optimization for the TFT model training, and we didn't conduct performance benchmarking. In a follow-up future work, we are considering these as promising future directions for improving the forecasting accuracy of the TFT models and examining how the TFT models compare with respect to other forecasting techniques in the literature when used with aviation data for anomaly detection. Augmenting the existing dataset with a more complete set of digital flight data features and incorporating additional modes of data is another promising future direction, such as gridded weather (e.g., convective weather, wind fields), voice, and textual information. These modes need not be raw inputs to the TFT but can be vectorized data representations derived from other upstream capabilities, such as large language models and speech recognition software.

One more promising future direction would be to add additional layers of analysis to the anomaly detection framework. As it stands, our approach flags anomalies based on the magnitude of error between the nominal TFT's prediction in a future forecast window and observed behavior at that time window. When a large prediction error results, we assume that the error is due to the presence of variable settings in the observed inputs that the model had limited exposure to during training, i.e., off-nominal precursors. The automated identification of such precursors and the anomaly type is a natural next step, which we anticipate could be explored via analysis of the TFT's latent representation of model inputs and temporal attention weights and the RMSE profiles from the multi-output predictions.

5. Conclusions

In this paper, we explore the feasibility of an anomaly detection approach via nominal behavior learning that uses the TFT model's forecasting capability with multivariate time-series flight data. UA is used as an anomalous event to evaluate anomaly detection

performance. Our anomaly detection approach does not require labels of historical anomalies per se, only a way to identify nominal flights. The results were found to be promising despite the limitations of a threshold exceedance-based UA event labeling process, which can lead to a lack of correspondence between anomalous flights and UA-labeled flights. The results indicated that the TFT models that learn nominal flight behavior can detect abnormal behavior and indicate their temporal locations in the time-series. By monitoring multiple flight output parameters (speed and altitude in this work) through the TFT model, we showed that the anomaly detection performance can be further improved.

Author Contributions: Conceptualization, B.A.; methodology, B.A.; implementation, B.A. and E.P.V.; validation, B.A.; formal analysis, B.A.; investigation, B.A.; data curation, E.P.V.; writing—original draft preparation, B.A. and E.P.V.; writing—review and editing, B.A., E.P.V. and H.T.; project administration, H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was produced for the U. S. Government under Contract Number 693KA8-22-C-00001, and is subject to Federal Aviation Administration Acquisition Management System Clause 3.5-13, Rights In Data-General (October 2014), Alt. III and Alt. IV (October 2009).

Data Availability Statement: The datasets presented in this article are not readily available due to data privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript; or in the decision to publish the results.

References

1. Memarzadeh, M.; Matthews, B.; Avrekh, I. Unsupervised anomaly detection in flight data using convolutional variational auto-encoder. *Aerospace* **2020**, *7*, 115. [CrossRef]
2. Lee, H.; Li, G.; Rai, A.; Chattopadhyay, A. Real-time anomaly detection framework using a support vector regression for the safety monitoring of commercial aircraft. *Adv. Eng. Inform.* **2020**, *44*, 101071. [CrossRef]
3. Li, L.; Das, S.; John Hansman, R.; Palacios, R.; Srivastava, A.N. Analysis of flight data using clustering techniques for detecting abnormal operations. *J. Aerosp. Inf. Syst.* **2015**, *12*, 587–598. [CrossRef]
4. Li, L.; Hansman, R.J.; Palacios, R.; Welsch, R. Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring. *Transp. Res. Part C Emerg. Technol.* **2016**, *64*, 45–57. [CrossRef]
5. Basora, L.; Olive, X.; Dubot, T. Recent advances in anomaly detection methods applied to aviation. *Aerospace* **2019**, *6*, 117. [CrossRef]
6. Olive, X.; Basora, L. Detection and identification of significant events in historical aircraft trajectory data. *Transp. Res. Part C Emerg. Technol.* **2020**, *119*, 102737. [CrossRef]
7. Das, S.; Matthews, B.L.; Srivastava, A.N.; Oza, N.C. Multiple kernel learning for heterogeneous anomaly detection: Algorithm and aviation safety case study. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–28 July 2010; ACM: New York, NY, USA, 2010; pp. 47–56.
8. Janakiraman, V.M.; Matthews, B.; Oza, N. Discovery of precursors to adverse events using time series data. In Proceedings of the 2016 SIAM International Conference on Data Mining, Miami, FL, USA, 5–7 May 2016; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2016; pp. 639–647.
9. Janakiraman, V.M.; Matthews, B.; Oza, N. Finding precursors to anomalous drop in airspeed during a flight's takeoff. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, USA, 13–17 August 2017; pp. 1843–1852.

10. Gavrilovski, A.; Jimenez, H.; Mavris, D.N.; Rao, A.H.; Shin, S.; Hwang, I.; Marais, K. Challenges and Opportunities in Flight Data Mining: A Review of the State of the Art. AIAA Infotech@Aerospace. 2016, p. 0923. Available online: https://arc.aiaa.org/doi/abs/10.2514/6.2016-0923 (accessed on 3 August 2024).
11. Janakiraman, V.M. Explaining aviation safety incidents using deep learned precursors. *arXiv* **2017**, arXiv:1710.04749.
12. Martinez, D.; Fernández, A.; Hernández, P.; Cristóbal, S.; Schwaiger, F.; Nunez, J.M.; Ruiz, J.M. Forecasting Unstable Approaches with Boosting Frameworks and LSTM Networks. In Proceedings of the 9th SESAR Innovation Days, Athens, Greece, 2–6 December 2019.
13. Wang, Z.; Sherry, L.; Shortle, J. Airspace Risk Management using Surveillance Track Data: Stabilized Approaches. In Proceedings of the 8th Integrated Communications, Navigation, Surveillance (ICNS) Conference, Dulles, VA, USA, 21–23 April 2015.
14. Wang, Z.; Sherry, L.; Shortle, J. Improving the nowcast of unstable approaches. In Proceedings of the 8th International Conference on Research in Air Transportation, Barcelona, Spain, 26–29 June 2016.
15. Ackley, J.L.; Puranik, T.G.; Mavris, D. A supervised learning approach for safety event precursor identification in commercial aviation. In Proceedings of the AIAA Aviation 2020 Forum, Online, 15–19 June 2020; p. 2880.
16. Bleu-Laine, M.H.; Puranik, T.G.; Mavris, D.N.; Matthews, B. Predicting adverse events and their precursors in aviation using multi-class multiple-instance learning. In Proceedings of the AIAA Scitech 2021 Forum, Online, 19–21 January 2021; p. 0776.
17. Reddy, K.K.; Sarkar, S.; Venugopalan, V.; Giering, M. Anomaly Detection and Fault Disambiguation in Large Flight Data: A Multi-modal Deep Auto-encoder Approach. In Proceedings of the Annual Conference of the Prognostics and Health Management Society, Denver, CO, USA, 3 October 2016; p. 7.
18. Zhou, C.; Paffenroth, R.C. Anomaly Detection with Robust Deep Autoencoders. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17), Halifax, NS, Canada, 13–17 August 2017; pp. 665–674.
19. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]
20. Hart, P.E.; Stork, D.G.; Duda, R.O. *Pattern Classification*; Wiley: Hoboken, NJ, USA, 2000.
21. Eckstein, A.; Kurcz, C.; Silva, M. *Threaded Track: Geospatial Data Fusion for Aircraft Flight Trajectories*; The MITRE Corporation: McLean, VA, USA, 2012.
22. Lowe, S.E.; Pfleiderer, E.M.; Chidester, T.R. *Perceptions and Efficacy of Flight Operational Quality Assurance (FOQA) Programs Among Small-Scale Operators*; Research Task Report; Office of Aerospace Medicine: Washington, DC, USA, 2012.
23. Stabilised Approach. Available online: https://skybrary.aero/tutorials/stabilised-approach (accessed on 24 July 2024).
24. PyTorch Forecasting Documentation. Available online: https://pytorch-forecasting.readthedocs.io/en/stable/ (accessed on 26 September 2023).
25. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.