THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 2 - Jean Jaurès

Présentée et soutenue par

Tianyi LI

Le 4 septembre 2023

Application des techniques d'apprentissage automatique pour la détection et la prédiction de turbulence d'aéronefs en vol

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Josiane MOTHE et Olivier TESTE

Jury
M. Nicolas LACHICHE, Rapporteur
Mme Karine ZEITOUNI, Rapporteure
Mme Maguelonne TEISSEIRE, Examinatrice
Mme Josiane MOTHE, Directrice de thèse
M. Olivier TESTE, Co-directeur de thèse
M. Philippe GOUPIL, Co-encadrant de thèse

Contents

List of Figures

List of Tables

Liste des Algorithmes

Acknowledgements

I wish to extend my sincere acknowledgments to the colleagues and friends that have contributed to the successful completion of this doctoral thesis.

First and foremost, I express my gratitude to my advisors Prof.Josiane Mothe, Prof.Olivier Teste and Dr.Philippe Goupil, for their exceptional guidance, expertise, and unwavering support. Their mentorship has been instrumental in shaping this research and extended beyond the academic realm. Thanks to Prof.Josiane Mothe and Prof.Olivier Teste, I have been able to gain insights into the world of research and make meaningful contributions to my research topic. Thanks to Dr.Philippe Goupil, I have learned how to think critically and effectively address engineering challenges.

I express my gratitude to the members of my thesis committee, Prof.Nicolas Lachiche, Prof.Karine Zeitouni and Prof.Maguelonne Teisseire for their valuable feedback and rigorous examination of this work.

I would like to thank my family for their support and encouragement, which have been indispensable throughout this PhD thesis. The companionship of my family provides me with a constant sense of warmth and fuels my motivation for both life and work. Especially during the COVID-19 pandemic, their selfless assistance and encouragement helped me navigate through various challenges.

I acknowledge all the colleagues and friends from both IRIT and Airbus, which has not only enabled me to conduct this research but also built a pleasant working environment. Their collective support and contributions have been invaluable, and I am truly thankful for all the help in making this thesis a reality. Thanks to them, my work is filled with joy. Being able to see these lovely friends every day makes me eager to go to work.

Résumé

Les phénomènes de turbulence atmosphérique sévère sont la principale cause de blessures en vol dans le transport aérien civil. Leur fréquence et leur gravité augmente en raison des changements climatiques. Il n'y a pas de méthode embarquée de détection de turbulence précise et fiable à court terme.

Pour résoudre ce problème, nous présentons dans cette thèse FUTURA (FUnctional shape feature for real time TURbulence Alerting), une approche basée sur les données pour la détection de la turbulence en temps réel qui utilise uniquement les données de capteurs déjà existantes à bord de l'avion. FUTURA fusionne trois technologies : le filtre de Kalman, l'analyse des données fonctionnelles et la détection d'anomalies des séries temporelles pour une identification précoce de la turbulence. Dans la première étape, nous proposons d'utiliser le filtre de Kalman en régime stationnaire pour approximer les signaux bruts de manière récursive. L'approximation fonctionnelle ne permet pas seulement d'éliminer implicitement le bruit des données, mais également de tirer parti des propriétés mathématiques des fonctions approximées. Dans la deuxième étape, les caractéristiques géométriques fonctionnelles sont extraites de l'approximation fonctionnelle obtenue lors de la première étape. Les caractéristiques géométriques extraites capturent la relation entre plusieurs variables et fournissent une nouvelle représentation des données originales qui peut nous aider à mieux identifier les comportements anormaux dans les séries temporelles multivariées. Pour finir, une technique de détection d'anomalies de séries temporelles est utilisée sur les caractéristiques géométriques extraites afin d'attribuer un score d'anomalie à chaque instance. En considérant les turbulences comme des anomalies et en établissant un seuil sur les scores d'anomalie, nous pouvons distinguer parmi les vols lesquels rencontreront prochainement des turbulences.

Afin d'examiner les performances de FUTURA, une série d'expériences est réalisée. La méthode est évaluée en utilisant des données des séries temporelles multivariées provenant de vols ayant rencontré différents niveaux de perturbation d'air, y compris des turbulences sévères, et des comparaisons sont effectuées avec des méthodes basées sur Gradient Boosting utilisées dans la littérature récente pour la prédiction de la turbulence. Les expérimentations démontrent que FUTURA est capable d'identifier 40% des turbulences sévères (taux de vrais positifs) 30 secondes à l'avance tout en maintenant un taux de faux positifs nul, ce qui surpasse non seulement les méthodes basées sur Gradient Boosting, mais répond également à l'exigence d'absence de fausses alarmes pour optimiser l'expérience passager et la fiabilité opérationnelle de l'avion.

Abstract

Severe atmospheric turbulence phenomena are the leading causes of in-flight injuries in civil air transport in the context of the increasing frequency and severity of turbulence due to climate changes. There is currently no precise and reliable short term on-board turbulence prediction method.

To address this problem, in this thesis, we propose the new FUTURA (FUnctional shape feature for real time TURbulence Alerting) function, a data-driven approach for real time turbulence prediction that utilizes only the sensor data already existing on-board of the aircraft. FUTURA fuses three technologies, namely Steady State Kalman Filter (SSKF), Functional Data Analysis (FDA) and time series anomaly detection for turbulence early identification.

In the first step, we put forward Steady State Kalman Filter (SSKF) to approximate raw signals in a recursive manner. The functional approximation can not only implicitly remove the data noise, but also allow us to make use of the mathematical properties of the approximated functions. In the second step, functional shape features are extracted from the functional approximation obtained in the first step. The extracted shape features capture the relationship among multiple variables and provide a new representation of original data that can help us better identify the abnormal behavior in multivariate time series data. Finally, a time series anomaly detection method is applied on the extracted shape features to give an anomaly score to each instance. Turbulence is considered as an anomaly and by setting a threshold on anomaly scores, we can distinguish which flights will shortly encounter turbulence and which flights will not.

To investigate the performance of FUTURA, a sequence of experiments are conducted. The method is evaluated using multivariate time series sensor data coming from flights that encountered different levels of air disturbance including severe turbulence and comparisons are made with Gradient Boosting based methods that are utilized in the recent literature for turbulence prediction task. Experimental results show that FUTURA can distinguish 40% of the severe turbulence cases (true positive rate) 30 seconds in advance while keeping a zero false positive rate, which not only outperforms the gradient boosting based methods, but also meets the zero false alarm requirement for optimizing the passenger experience and the aircraft operational reliability.

Chapter 1

Introduction

Table of contents

Abstract.

Atmospheric turbulence phenomena are the main causes of injuries in civil air transport and due to climate change, the frequency and severity of turbulence is increasing. There is currently no precise and reliable short term on-board turbulence prediction method. With the aim of designing a real time atmospheric turbulence early identification and alerting function on board of commercial aircraft, the dedicated algorithm should be as 'light' as possible in terms of computation efforts and memory usage to meet the requirement of the embedded computer and provide turbulence predictions in a real time manner. On board aircraft warning functions should also deliver robust outcomes thus keep the false alarm rate as low as possible. In response to these practical needs, we propose FUTURA (FUnctional shape feature for real time TURbulence Alerting), an on-board turbulence alerting function that combines three principles, namely Steady State Kalman Filter, Functional Data Analysis and Time Series Anomaly Detection.

1.1 Introduction

In this chapter, we present the industrial context, scientific motivation, and contributions of our work.

We first introduce aeronautical systems with particular emphasis on the onboard sensors and flight control systems. Aircraft onboard sensors are the data source of the proposed turbulence alerting algorithm and flight control computer is where the algorithm is planned to be embedded. Atmospheric turbulence including its classification and intensity metrics are presented. We then summarize and clarify the industrial constraints as well as objectives of our research. We transform an industrial problem to a scientific problem by taking all the industrial constraints into consideration and present each scientific field involved in our research.

1.2 Industrial context

1.2.1 Aeronautical system

1.2.1.1 Aeronautical systems and onboard sensors

A civil aircraft is composed of the engines, structure and systems. According to Aerospace Recommended Practice (ARP) 4754 (SAE 2010), a system is a combination of inter-related items arranged to perform a specific function(s)'. A civil aircraft encompasses about 100 different systems, which include, for example, the flight control system, the landing gear system and the electrical system, just to name a few.

Coming from different aircraft systems, on board sensors play an important role in the process of automating aircraft control, piloting and monitoring aircraft systems. They reflect the evolution of aeronautical systems state in a changing and disturbed environment. Depending on the type of aircraft, the available sensors and their adopted technology can be very different. The same anemometric sensors (i.e. total pressure, static pressure and total temperature sensors, allowing the relative wind speed to be measured for an aircraft) and clinometric sensors (i.e. angle of attack and sideslip, indicating the inclination of the airplane in relation to the air mass), however, can be found on different airplanes (Alcalay 2018), namely:

- 3 to 4 angle of attack sensors;
- 3 Pitot tubes generating 3 total pressure measurements;
- 6 static pressure taps generating 3 static pressure measurements;
- 2 or 3 total temperature sensors;
- 1 total temperature sensor per motor;
- 2 static sockets per motor, allowing the reconstruction of 1 pressure measurement;

The aircraft sensors are also valuable resources for meteorological analysis and forecast. Especially on atmospheric turbulent flows, this phenomenon can be detected using aircraft sensor responses. In this thesis, we develop a machine learning based turbulence early identification and alerting function leveraging the aircraft on-board sensor data.

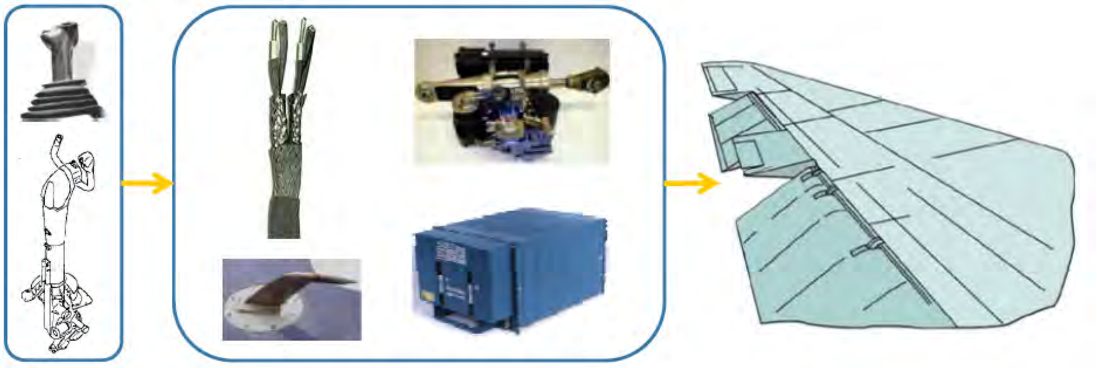


Figure 1.1: The Flight Control System (FCS) contains all the elements between pilot inputs and control surfaces.

1.2.1.2 Flight control system

The Flight Control System (FCS) contains all the elements between pilot inputs (in the cockpit) and control surfaces (i.e. movable parts that allow to change the aircraft aerodynamic configuration), dedicated to control the aircraft altitude, trajectory and speed. More specifically, it consists mainly of 5 different components: the pilots control inputs, the FCS computers, the power sources, the actuators and the control surfaces. The pilot inputs are first linked to a FCS computer. Considering both pilot orders and flight sensor parameters, the FCS computer calculates the command and sends it to the actuator that moves the control surface.

In place of mechanical FCS, Airbus initially introduced Electrical Flight Control System (EFCS) on the Airbus A310 (1982), on the spoilers, slats and flaps only, followed by a complete generalization on all control surfaces on the A320 (1987) (Goupil 2011). The EFCS system is now recognized as the industrial standard in commercial applications and it provides more sophisticated control of the aircraft and flight envelope protection functions (Traverse 2004). The most important requirement and objective of the flight control system is to ensure the safety of the flight. Based on the safety requirement, the flight control system also needs to be robust to failures and enhance aircraft's performance. Transitioning from mechanical FCS to electrical FCS has also allowed to contribute to significantly reduce the pilot workload.

As the brain of FCS, the flight control computer is a high integrity, low size, weight and power airborne computer that has strict memory and computation limits

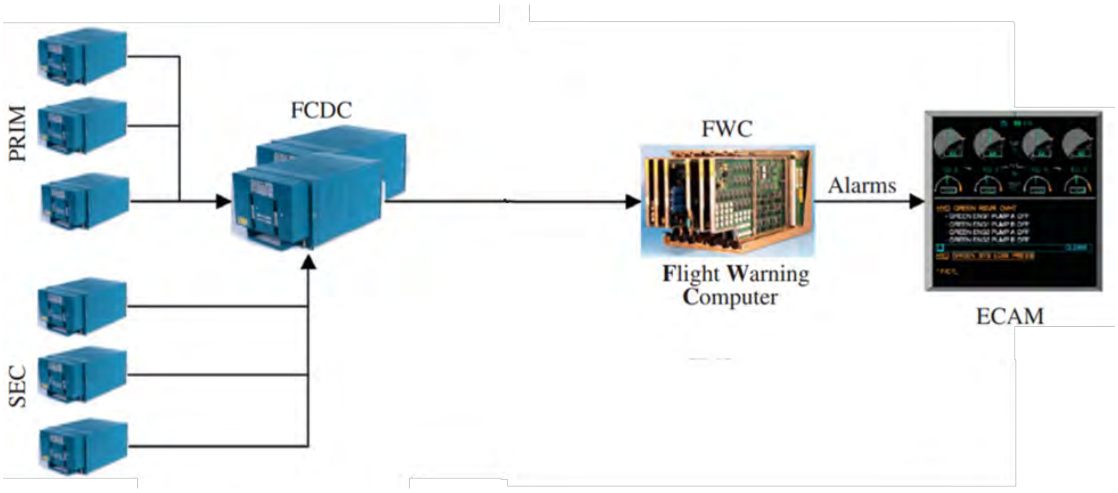


Figure 1.2: The turbulence early identification function is dedicated to be integrated in the flight control system.

(Goupil 2011). It processes the flight control laws and provides commands for flight control surfaces. The turbulence early identification function designed in this thesis is dedicated to be integrated in the flight control computer. After receiving the flight parameters, flight control computer calculate the turbulence prediction result and send it to Flight Control Data Concentrator (FCDC) computers (see figure 1.2). FCDC computers are in charge of the system maintenance and warnings, cautions and indications. The FCDCs compute some logics to generate warning messages towards the Flight Warning Computer (FWC). These messages are displayed in the cockpit through a dedicated screen (ECAM) (Goupil 2011). As shown in figure 1.2, there are the 6 EFCS computers on an A380 including 3 primary (PRIM) computers and 3 secondary (SEC) computers that have different hardware and software than PRIM (Goupil 2011). The reason for the hardware redundancy as well as dissimilarity between PRIM and SEC is to be compliant with Airworthiness requirements for aircraft certification and to design a fault-tolerant aircraft (Traverse 2004).

As the turbulence alerting function aims to be embedded function in flight control computer, several constraints need to be taken into consideration (Goupil 2011):

- Considering the limited computing capacities of flight control computer, a complex algorithm must be developed with as many simplifications as possible for effective implementation.

- Flight control computers operate on a multi-rate time-triggered basis, resulting in varying sampling periods for data processing, even within a single unit. For example, if an algorithm computes every 10 ms, this algorithm must adapt the data that are produced every 40 ms to its faster sampling time.

- Data measured by different sensors may be sent to flight control computers with different sampling frequency.

- Maintaining a low false alarm rate is crucial to avoid degrading the operational reliability. A false alarm may leads to a system reconfiguration and potentially to degrade the flight envelope protection level.

It is worth noticing that the proposed turbulence warning function does not seek to control the aircraft nor change the trajectory of the aircraft, but to warn the crews and passengers for the upcoming turbulence. As part of the flight control computer, it is imperative that the turbulence warning function complies with the safety and robustness requirements of the flight control system.

1.2.2 Atmospheric turbulence

Atmospheric turbulence phenomena are not only major concerns for the safety and comfort of passengers and crew, but also for safe and cost-effective aircraft operations (Sharman 2016). According to International Air Transport Association (IATA 2018), turbulence is responsible for 25% of injuries in flight and causes 14% of flight diversions. 13% of turbulence events lead to aircraft damage and the most serious events can trigger more than 100 insurance claims. In addition, the frequency and severity of turbulence are increasing due to the climate change (IATA 2018).

1.2.2.1 Generation mechanism

According to different generation mechanisms, atmospheric turbulence can be classified as Convective induced turbulence, Low-level turbulence, Mountain wave turbulence, Clear-air turbulence and Aircraft-induced turbulence (Sharman 2016). Ideally, the proposed method should handle all types of turbulence except the Aircraft-induced turbulence.

1. **Convective induced turbulence.** Turbulence associated with convective clouds is called convective induced turbulence and includes in-cloud or near-cloud turbulence. Since convectively induced turbulence are directly related to convective clouds, it tends to have a relatively brief lifetime of just a few minutes (Sharman 2016).

2. **Low-level turbulence.** Caused by strong winds related to a surface frontal passage, dry hot air flows over a hot surface or flows over obstacles like mountains, trees and buildings, low-level turbulence is a safety hazard during takeoffs and landings, especially for small aircraft (Sharman 2016).

3. **Mountain wave turbulence.** Mountain Wave Turbulence is linked to the significant amplitude and breaking of gravity waves above and behind mountainous terrain (Sharman 2016). Wave breaking may occur above the mountain or at lower altitudes downstream from the mountain.

4. **Clear-air turbulence.** Caused by enhanced wind shears and reduced stability near jet streams, Clear-Air Turbulence (CAT) often occurs in clear air. CAT typically occurs in the upper troposphere and lower stratosphere.

5. **Aircraft-induced turbulence.** Aircraft-induced turbulence typically happens near airports when a lighter aircraft trails a heavier aircraft and impacted by its trailing vortex wakes.

1.2.2.2 Different turbulence levels

Despite the different nature and generation mechanism, all kinds of turbulence can be categorized as “light,” “moderate,” “severe,” and “extreme”, depending on the size and intensity of the encountered atmospheric turbulent eddies and the corresponding aircraft’s response (FAA 2014) (see table 1.1 for a detailed description of different atmospheric turbulence levels). The current work mainly concentrates on severe and extreme turbulence that pose a higher risk of injury.

1.2.2.3 Turbulence intensity metrics

In-flight turbulence intensity can be quantified by vertical load factor or Eddy (or Energy) Dissipation Rate (EDR) (Lee 2022). Load factor is the ratio of the aircraft’s lift

Table 1.1: Atmospheric turbulence levels.(FAA 2014)

Description	Aircraft Reaction	Occupants Reaction
Light	Slight changes in altitude and/or attitude.	Slight strain against seat belts or shoulder straps. Unsafe objects may be moved slightly. Food service may be conducted and little or no difficulty is encountered in walking.
Moderate	Changes in altitude and/or attitude occur. The aircraft remains in control. Variations in indicated airspeed.	Strains against seat belts or shoulder straps. Unsecured objects are dislodged. Food service and walking are difficult.
Severe	Large, abrupt changes in altitude and/or attitude. Large variations in indicated airspeed. Aircraft may temporarily lose control.	Occupants are forced violently against seat belts or shoulder straps. Unsecured objects are tossed about. Food Service and walking are impossible.
Extreme	Aircraft is violently tossed about and is practically impossible to control. It may cause structural damage.	

to its weight (see equation 1.1) (Clancy 1975).

$$N_z = \frac{L}{W} \tag{1.1}$$

Where L is the lift of an aircraft, W is the weight and N_z the vertical load factor.

Load factor is an aircraft dependent metric that describes the stress on aircraft structures and reflects the real bumpiness and passenger feelings during the turbulence. A vertical load factor equals to 1 indicates a straight level flight and a load factor greater or less than 1 is the consequence of maneuvers or turbulence.

EDR, on the contrary, is an objectively turbulence intensity metric independent of the aircraft type (Meymaris 2019). Owing to its objectivity, EDR is used to deliver the encountered turbulence information among aircraft. The aircraft that receives EDR interprets the turbulence intensity level based on its appropriate aircraft type, altitude, airspeed, weight and flight conditions to infer its response to turbulence. EDR has been adopted by both International Civil Aviation Organization and World Meteorological Organization as the official turbulence reporting metric (Lee 2022). Both National Aerospace Laboratory and National Center for Atmospheric Research (Meymaris 2019) developed their corresponding algorithms for EDR calculations.

The advantages and inconveniences of EDR and Load factor as turbulence intensity metrics can be summarized as follows (Meymaris 2019):

1. EDR
    - Advantage
        - Independent on aircraft models and thus EDR message can be delivered and easily interpreted by other aircraft.
    - Inconvenience
        - Can not reflect the passenger's real feeling compared to the load factor.
        - Cannot be measured directly by the sensor. Embedded software is required to calculate from sensor data.
2. Load factor
    - Advantage
        - Reflect the real bumpiness and passenger feeling during turbulence.
        - Can be measured directly by aircraft sensors.
    - Inconvenience
        - Aircraft dependent. Not a universal measurement thus may not be used directly on other aircraft models.
        - Influenced by the aircraft operations and maneuvers.

The objective of this thesis is to design a turbulence alerting function on board of each commercial aircraft. Such function should identify in advance severe atmospheric turbulence that result in passenger injuries ahead of each aircraft in a real time manner. For this purpose, we use load factor to describe the turbulence intensity as it reflects the aircraft bumpiness and passengers' feelings.

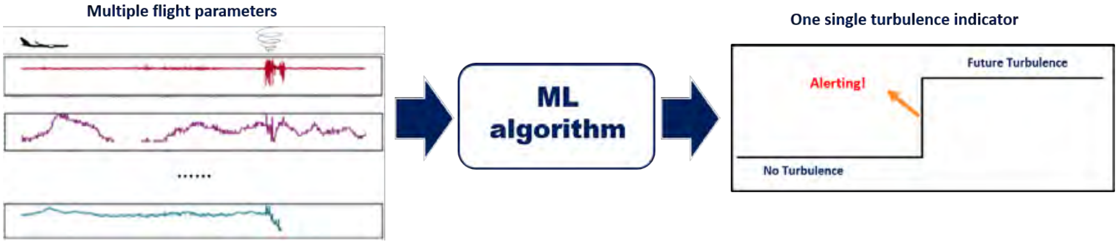


Figure 1.3: The machine learning based turbulence alerting function transforms on-board sensor signals into a turbulence indicator.

1.2.3 Objective and industrial constraints

In this thesis, we aim to design a Machine Learning based turbulence alerting function on board of commercial aircraft using sensor data. More specifically, we transform the on-board sensor signals into a turbulence indicator (see figure 1.3).

The main industry constraints are summarized as follows:

- The aircraft flight control computer has strict computational limit and memory requirement for the embedded functions. The designed turbulence early identification algorithm should be as 'light' as possible in terms of computation efforts and memory usage to meet the requirement of the embedded computer.

- The turbulence identification function should provide turbulence predictions and alert in a real time manner for timely turbulence warnings which means sufficiently in advance. The time required for alerting is still to be defined w.r.t operational constraints. Related to the first constraint, the meaning of 'real time' has two folds:1. The recursive (or iterative, incremental) nature of the algorithm to reduce the memory requirement. The algorithm should update its prediction according to the new coming data point without needing to store and reuse all historical data every time. 2. The low computation cost at each iteration.

- On board aircraft warning functions should deliver robust outcomes thus keep the false alarm rate as low as possible. Although the false alarm of such an alerting function does not harm in flight operation, its accumulation will make pilot lose confidence in the turbulence alerting function and make passengers lose trust in airlines.

1.3 Scientific motivation

1.3.1 From an industrial problem to a scientific problem

According to its functionality and system requirements, a data-driven on-board turbulence identification and alerting function has the following characteristics:

- Data collected from on-board sensors are composed of multivariate time series where each variable represents a different sensor parameter.
- The data are highly imbalanced: there are much less severe turbulence flights than normal flights which is consistent with the reality.
- The turbulence should be identified early enough before the aircraft enters the turbulence zone to allow sufficient time for passengers and crew to secure.
- The turbulence identification algorithm should meet the computational limit and memory requirements of the on-board aircraft control computer and provide real time turbulence warnings.
- Sensor measurements contain noise and different sensors have different sampling frequencies.

1.3.2 Multivariate time series

A time series is a sequence of values that has temporal dependency among data points (Gupta 2020a) (Faouzi 2022). Time series data can be found in various fields such as engineering, chemometrics, econometrics, finance, healthcare and internet activity.

Depending on the dimension of the measurements, time series can be classified into univariate and multivariate time series. A time series is univariate if its observation value at each timestamp returns a scalar and multivariate if its observation value returns a m dimensional vector where m is the number of variables.

Although multivariate time series can be seen as a stack of multiple univariate time series and one can treat each variable independently, such a way of analyzing multivariate time series data results in omitting the relationships between the multiple variables (Blázquez-García 2021). In multivariate time series the variables can be correlated between each other along time and the relationships between variables contain important information. Multivariate time series has not only temporal variabilities along each variable but also correlation variabilities across them.

On board sensors coming from different aircraft systems (e.g. avionic systems, engines) generate multivariate time series, which allow data driven approaches to become possible options. Figure 1.4 shows the variation of on-board sensors' measurements while encountering severe turbulence. Flight variables exhibit strong fluctuation in the turbulence area. Atmospheric turbulence can be detected if these specific behaviors are early identified.

The relationship between flight parameters can be deterministic. For example the relation between total air temperature and static air temperature can be given by (Trenkle 1973):

$$\frac{TAT}{SAT} = 1 + \frac{\gamma - 1}{2} M_a^2 \tag{1.2}$$

where TAT is the total air temperature, SAT is the static air temperature, M_a^2 is the square Mach number and γ is the ratio of specific heats.

The relationship between flight parameters can also be non-deterministic, e.g. the relation between total air temperature and motor temperature. Figure 1.4 illustrate the behavior of five flight parameters near a turbulent region. While parameters 1,2 and 4 seem to have synchronized changes, the relation between parameters 3 and 5, as well as their relationship with all other parameters can not be concluded intuitively from visualization.

1.3.3 Rare event

The data collected from a rare event are often highly imbalanced where one class has a higher percentage of data than the others (Lewis 1994). In addition to atmospheric turbulence detection, similar scenarios are observed in areas such as detecting fraud in banking operation, detecting fraudulent telephone calls, managing risk and medical diagnosis.

In the supervised case, classification performance of machine learning algorithms is affected with highly imbalanced dataset (Chawla 2010) (Buda 2018) (Thabtah 2020). The machine learning classification are trained with the aim of maximizing the overall accuracy of the derived classifiers with the assumption of a balanced number of instances per class (Ganganwar 2012), which can result in a high degree of accuracy on the

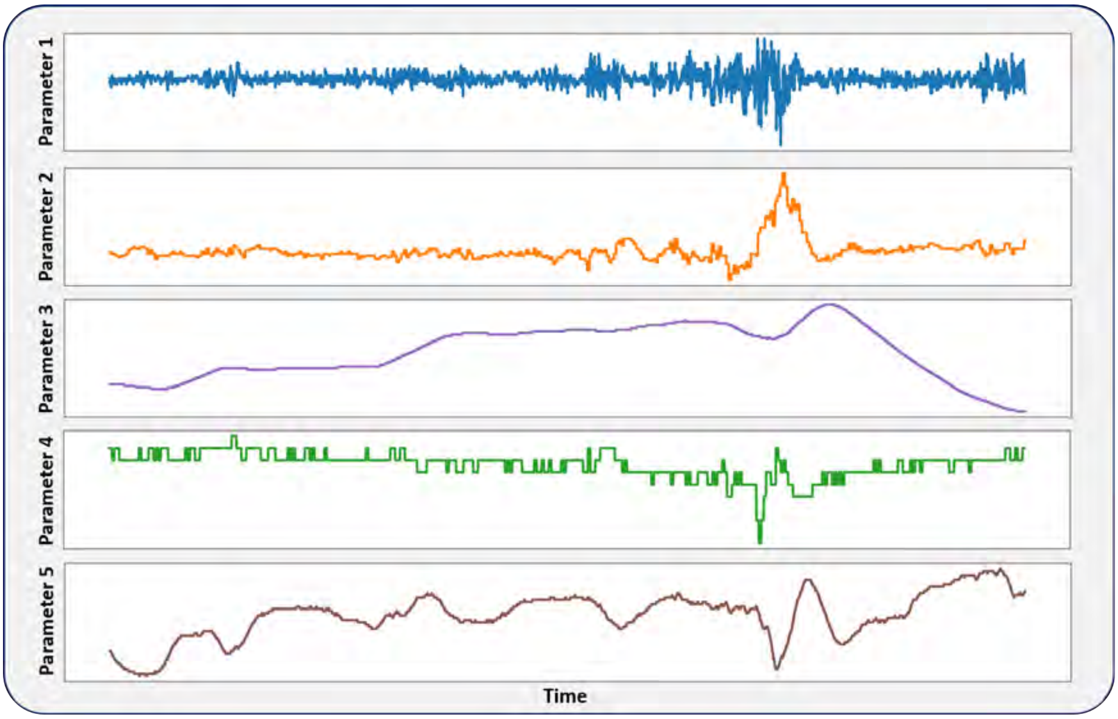


Figure 1.4: Signals of five flight parameters near a turbulent region.

majority class and poor performance on the minority class as the major class has the higher weight in the data (Thabtah 2020). As the minor class is often of greater interest in the real-world application, applying the machine learning classification model directly to highly imbalanced data may not lead to desired results. Techniques to tackle the class imbalance problem include sampling methods and algorithmic level solutions. Sampling methods resample the data set by oversampling the minority class or by under-sampling the majority class to have an approximately equal class distributions. Common used methods include Synthetic Minority Over-sampling Technique (SMOTE) (Chawla 2002), borderline-SMOTE1 and borderline-SMOTE2 (Han 2005). Algorithmic level solutions include cost sensitive learning which attaches the cost associated with misclassification during the learning task, thresholding methods that adjust threshold of the classifiers to reduce cost of misclassifications and ensemble learning approaches (Thabtah 2020).

In the unsupervised case, the anomaly detection or outlier detection algorithm has its advantage in detecting rare data samples that deviate remarkably from the general distribution of the data (Braei 2020). Anomalies occupy only a small portion of the data and their distribution is remarkably different from the major class of the dataset which are considered as the normal data (Braei 2020). Therefore, depending on the nature of the real application, rare events can be regarded as anomalies or outliers, which allow applying unsupervised anomaly detection algorithms to identify rare events.

As sensor signals exhibit particular behaviors in the turbulence zone and represent only a very small part of the data, in this thesis, turbulent signals are considered as outliers in magnitude and shape compared with normal sensor signals and multivariate functional outlier detection methods are applied to identify turbulence several seconds or minutes in advance.

1.3.4 Time series early classification

Compared with time series classification that uses complete time series to predict the class label, time series early classification aims to classify an incomplete time series as early as possible with a desired level of classification performance (Gupta 2020a). The earliness of classification is essential in many time sensitive domains such as gas leakage (Hatami 2013), bearing faults (Ahn 2020), earthquake (Fauvel 2020), early detection of infected people during pandemic period of COVID-19 (Gupta 2020a). Although using the complete time series can help achieve better classification performance, it is not applicable in streaming applications and waiting for more data also results in missing the right timing to identify hazard or important events. Thus, the main challenge of early classification approach lies in balancing between two conflicting objectives, accuracy and earliness, at the same time (Mori 2019).

1.3.5 Functional Data Analysis

Functional Data Analysis (FDA) aims to represent high dimensional discrete observations as functions (Ramsay 2008). It assumes that there exists a function generating the observed data and that the underlying function is smooth with one or more derivatives (Ramsay 2008). Observations from high rate sensors deployed in commercial aircraft are real continuous functions observed at discrete time points, thus can be treated as functional data (Lejeune 2020).

Distinguishing FDA from treating the data as just multivariate time series, the smoothness property is essential as it can not only implicitly remove noise from the raw signal but also allow us make use of the information contained in their derivatives.

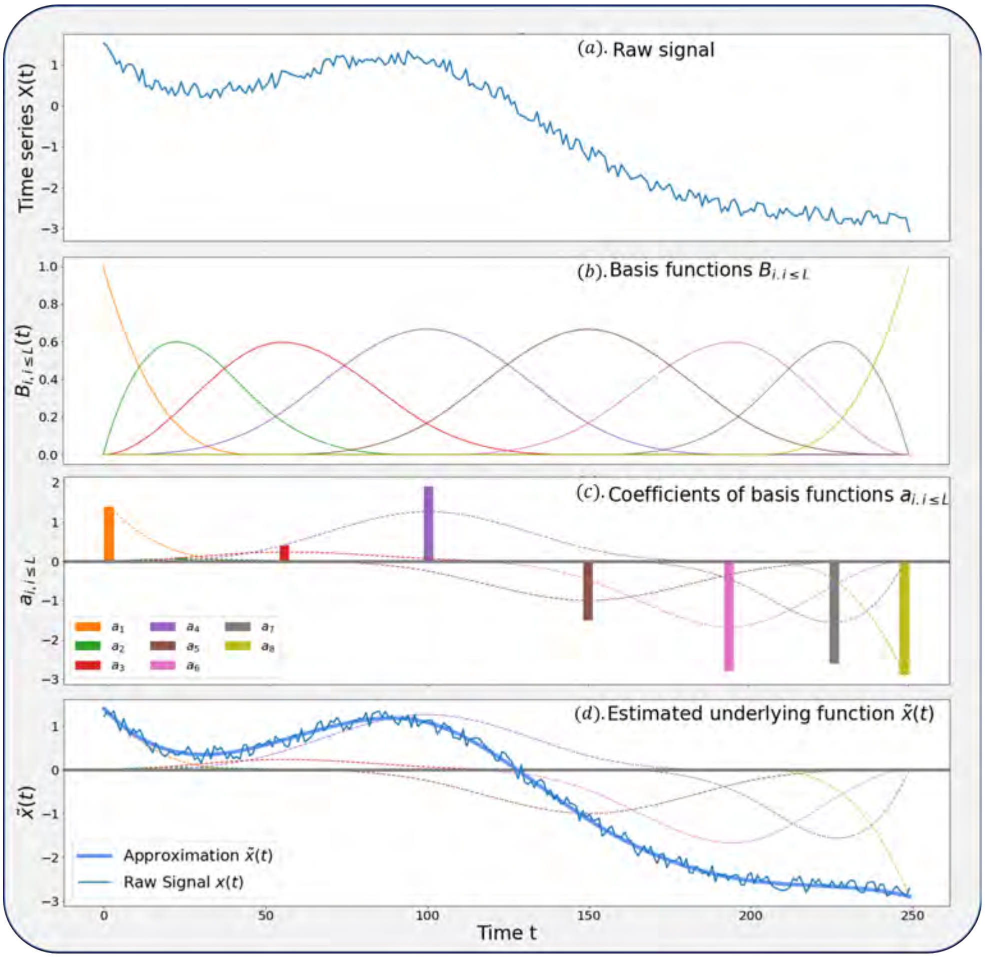


Figure 1.5: An illustration of approximating raw signals by b-spline basis functions. a) the raw signal. b) initialized basis functions. c) coefficients of basis functions obtained by least square estimations (colored vertical bars). d) the estimated underlying function.

Although one can calculate derivatives by forward difference or central difference, estimated derivatives can easily go wrong by differencing methods even if functions are analytically available (Press 1992) and the estimated derivatives can be rather noisy.

The underlying functions can be constructed by a weighted sum or linear combination of basis functions. Basis functions are a series of known and mathematically independent functions. For example, Fourier basis functions are commonly used to approximate periodic data and b-spline basis are commonly used for non-periodic data (See figure 1.5).

1.3.6 Kalman Filter

As the underlying function of a time series is constructed by a weighted sum of basis functions, the familiar technique of computing the amplitudes or coefficients of basis functions is by minimizing the sum of squared errors. Such a way of approximating raw signals demands the availability of the complete time series thus can not be directly utilized for real-time applications. Even though it is possible to apply the least square estimation within a sliding window, such methods require to store all the data in the sliding window which is not desirable considering the strict memory limit of aircraft embedded control computer and does not meet the real-time industry requirements. To tackle this problem, the Kalman filter (Kalman 1960) is applied to approximate raw sensors signals in real time.

Assume that there is a dynamic system of which measurements are available, and its current condition or behavior at a given time point is quantitatively described by the system states. For example, a car speed monitoring system where the system state is the car speed and the measurements can be wheel rotation speed measured by wheel speed sensors. In the case of b-spline approximation, the system we constructed is not a physical system. The measurements are the continuously-generated sensor data points and we regard the coefficients of underlying basis functions as system states. Different from smoothing, where measurements derived later than time t can be used in obtaining information about the system, and prediction, which aims to predict the system information of $t + \delta t$ at time t, filtering refers to the process of obtaining information about the system at a given time t by utilizing measurements collected up to that point in time (Anderson 2012). As one of the most famous filter to estimate the state of a dynamic system in the presence of noisy measurements, Kalman Filter (Kalman 1960) operates in a recursive manner, consistently updates its state estimates as it receives new data, making it especially appropriate for real-time applications and systems with time-varying parameters.

Based on the assumption that the system state follows a linear model over time, the Kalman Filter considers measurements are linearly related to the underlying state and corrupted by noise (Simon 2006). It works in two steps: prediction and update. The prediction step involves using the previous state estimate along with the system model which describes the evolution of the system state over time, to estimate the current state. Then, in the update stage, it integrates the most recent measurement to improve the state estimate, considering both the measurement noise and the system model uncertainties (Simon 2006). Kalman filter can also be viewed as a Bayesian estimation approach: in each time step, the a priori state estimate is first obtained after the prediction step and then get updated with the new observations to obtain the a posteriori state estimate (Welch 1995).

1.4 Contributions

Our main contribution lies in the proposal of FUTURA (FUnctional shape feature for real time TURbulence Alerting), an on-board turbulence early identification model utilizing only the sensor data on-board of aircraft.

The introduction of FUTURA is intended to tackle the subsequent requirements and yet-to-be-solved challenges:

- Although severe atmospheric turbulence phenomena are the leading causes of in-flight injuries in civil air transport (IATA 2018), there is no precise and reliable in-flight turbulence prediction method (Sharman 2016). The state-of-the-art turbulence detection methods used on board commercial aircraft include pilot reports and in situ turbulence algorithms only provide turbulence observations but not their prediction. On-board weather radar detects turbulence in wet air without any precise announcement about the timing. Turbulence forecasting methods are difficult to precisely predict turbulence due to limited spatial and temporal resolution (Kim 2018).

- Using high frequency multivariate time series generated by sensors for turbulence early identification, flight parameters can be correlated between each other along time and the relationship between variables contains important information. Most of the research work on time series classification as well as time series early classification is dedicated to exploring univariate time series classification. Most existing multivariate time series early classification methods do not consider these relationships (Ma 2017; Gupta 2020a).

- To implement a turbulence alerting function in an embedded system, the strict memory limits and computational requirements should be taken into consideration. We need a method that has a low memory requirement as well as a low computation complexity to make real time turbulence prediction become possible.

FUTURA combines three principles, namely Steady State Kalman Filter (SSKF) (Simon 2006), Functional Data Analysis (FDA) (Ramsay 2008) (Jauch 2017) and Time Series Anomaly Detection for turbulence early identification.

In the first step, we put forward Steady State Kalman Filter (SSKF) for functional approximation. SSKF is utilized to approximate raw signals by estimating the coefficients of b-spline basis functions in a recursive manner. The functional approximation can not only implicitly remove the data noise, but also allows us to make use of the mathematical properties of the approximated functions. For example, if b-spline basis functions are second-order derivable, the first and second derivatives of the approximation function can then be used for further analysis. In the second step, functional shape features (Srivastava 2016) are extracted from the functional approximation obtained in the first step. The extracted shape features capture the relationship among multiple variables and provide a new representation of the original data that can help us better identify the abnormal behavior in multivariate time series data. Finally, a time series anomaly detection method (such as Functional Isolation Forest (FIF) (Staerman 2019)) is applied on the extracted shape features to give an anomaly score to each instance. Turbulence is considered as an anomaly and by setting a threshold on anomaly scores, we can distinguish which flights will shortly encounter turbulence and which flights will not. Besides FUTURA, we also propose a feature selection method for multivariate time series data.

To investigate the performance and properties of FUTURA, a sequence of experiments is conducted. The method is evaluated using multivariate time series sensor data coming from flights that encountered different levels of air disturbance including severe turbulence and comparisons are made with gradient boosting based methods that are utilized in the recent literature (Emara 2021) for turbulence prediction task. Experimental results show that FUTURA can distinguish 40% of the severe turbulence cases (true positive rate) 30 seconds in advance while keeping a zero false positive rate, which not only outperforms the gradient boosting based methods, but also meets the zero false alarm requirement for optimizing the passenger experience and the aircraft operational reliability.

The thesis is structured as follows: Chapter 2 introduces related works for our proposed task in both turbulence detection field and multivariate time series early classification as well as anomaly detection field. Chapter 3 describes the method of functional shape features for turbulence detection in an offline manner where the real time constraints are not yet taken into consideration. Chapter 4 details the principle of FUTURA, our proposed real time turbulence identification approach. Various experiments including the preliminary data explorations, experiments for the offline methods and experiments for FUTURA are introduced in Chapter 5. Conclusions are discussed in Chapter 6.

CHAPTER 2

Related Works

Table of contents

Abstract.

There is no precise and reliable in-flight turbulence prediction method. The state-of-the-art turbulence detection methods used on board commercial aircraft include pilot reports and in situ turbulence algorithms to calculate energy dissipation rate. They provide turbulence observations but not their prediction. On-board weather radar on the other hand only detects turbulence in wet air without any precise announcement about the timing. Turbulence forecasting methods are difficult to precisely predict turbulence due to limited spatial and temporal resolution. Equipped with a large number of sensors coming from different aircraft systems, we believe that the flight variables (high frequency multivariate time series generated by sensors) as well as their relationships contain useful information indicating upcoming turbulence. Most existing multivariate time series early classification methods, however, do not consider the relationship among variables. Regarding time series anomaly detection methods, although many application scenarios require algorithms to operate in real time, no study has been done to analyze the computational cost of anomaly detection methods for their possibility to be used in real-time contexts.

2.1 Introduction

In this chapter, we highlight the most relevant works for our proposed task as well as the remaining gaps in both turbulence detection field and multivariate time series early classification and anomaly detection field.

In the first section, we summarize the turbulence detection and prediction state-of-the-art being applied or proposed in aeronautical industry and academia. We divide turbulence detection and prediction techniques into four categories: Turbulence forecasting methods, Turbulence detection on-board of commercial aircraft, Advanced sensing techniques for turbulence detection and AI-based turbulence prediction using sensor data.

A data driven turbulence identification problem can be tackled by supervised times series early classification methods or unsupervised times series anomaly detection approaches. In the second section of this chapter, we present the related research works of time series classification, early classification as well as anomaly detection.

2.2 Turbulence detection and prediction state-of-the-art

2.2.1 Turbulence forecasting methods

Established by the International Civil Aviation Organization (ICAO), the World Area Forecast Centers (WAFCs) provides 6-hourly forecasts of weather information such as wind, temperature and turbulence to global aviation users (Gill 2014). The current WAFCS forecasts utilize global Numerical Weather Prediction (NWP) model output to estimate turbulence on a 1.25° × 1.25° (equivalent to 138.75km × 138.75km) gridded. At the request of ICAO, WAFCS plans to provide calibrated turbulence forecasts on a 0.25° × 0.25° grid every 6 hours by 2024 (Kim 2018). Using NWP output data, Graphical Turbulence Guidance (GTG) developed by the National Center for Atmospheric Research (NCAR) provides turbulence forecasts by combining several different turbulence diagnostics (Sharman 2016). Nevertheless, due to the resolution of NWP models, the spatial and temporal resolution of current forecast system are still too large to capture the rapidly changing atmospheric turbulence that has only a few hundred meters to a few kilometers in size (Sharman 2016) (Kim 2018).

2.2.2 Turbulence detection on-board of commercial aircraft

2.2.2.1 Pilot report

Pilot report (often termed as PIREP or AIREP (air report) by aviation meteorologists) contains the manual reporting of weather conditions encountered during the flight. Most commercial aircraft are equipped with an on-board computer to calculate wind direction and speed. Since the early 1970s, PIREPs not only contained these measured variables, but also the encountered turbulence information (intensity, duration and cloud information) (Sharman 2016). Turbulence reports contained in PIREPs are based on the pilots' subjective assessment therefore cannot precisely and objectively reflect the intensity, location and appearance time of turbulence. It is a way of recording the turbulence events encountered during the flight instead of a predictive method.

2.2.2.2 In-situ Energy Dissipation Rate (EDR)

The Energy Dissipation Rate (EDR) (Sharman 2016) is a measure of turbulence intensity. EDR can not be measured directly but is calculated from measurable flight variables such as velocities and temperature. Several algorithms for estimating EDR for commercial airline flights are proposed in the literature (Huang 2019) (Chen 2019). As an objective intensity measure of the encountered turbulence, EDR can be used for warning following aircraft. Same as pilot reports, it is not a predictive method.

2.2.2.3 Airborne weather radar

As a necessary equipment for civil aircraft, weather radar was first utilized in civil aviation for autonomous navigation based on characteristic landmarks; e.g. cities and lakes (Yanovsky 2005). Today, the development of airborne weather radar is mainly related to the detection of hazardous weather phenomena such as severe turbulence. The weather radars on board of aircraft are able to detect clouds and precipitation, and display a radar image of the surface in map mode (Yanovsky 2005). The weather radars can only detect turbulence in wet air without the precise timing of upcoming turbulence. For non-rainy weather, they might not provide reliable turbulence detection result (Sharman 2016).

2.2.3 Advanced sensing techniques for turbulence detection

2.2.3.1 LIDAR

LIDAR (Light Detection and Ranging) emits pulsed laser and collects scattering light reflected from aerosol particles in the atmosphere. It has been used to measure wind gusts and turbulence in front of aircraft for feed-forward flight control (Schmitt 2007), as well as obstacle detection and avoidance for unmanned aerial vehicles (Ramasamy 2016). JAXA (Japan Aerospace Exploration Agency) developed on board Doppler LIDAR to detect clear-air turbulence (JAXA 2018). The developed clear-air turbulence detection system was tested on small jets in 2017 and then on Boeing 777 on 2018. JAXA will continue their research to realize the practical application of the developed turbulence detection system.

2.2.3.2 Infra-sonic sensor to detect turbulence

Infrasound consists of sound waves between 0.001 and 20 hertz, which is below human audibility. Based on the definite infrasound signature of clear-air turbulence, researchers at NASA (National Aeronautics and Space Administration) developed infrasonic microphones that capture low frequency sound waves for clear-air turbulence detection (NASA 2021). The microphones were first tested on the grounds by placing them in an equidistant triangular pattern and then, with the help of stratodynamics, implemented on an uncrewed stratospheric glider for both turbulence detection and aeronautical research. Stratodynamics will conduct more flight tests to further improve the infra-sonic microphone technique.

2.2.4 AI based turbulence prediction using sensor data

Gradient boosting The authors in (Emara 2021) trained machine learning models to estimate turbulence severity utilizing the time-series data collected from airline flights. The collected data is segmented into windows of fixed length and each window contains a single vector obtained by flattening all parameters values. EDR is calculated (Chen 2019) to label each flight. Gradient Boosting Classifier (GBC) and Gradient Boosting Regressor (GBR) are applied to predict turbulence severity 10 seconds in advance. Both models were trained on 55,885 flights (53,492 lights turbulence, 2170 moderate turbulence and 223 severe turbulence) and tested on 215,148 flights (213,968 lights turbulence, 1069 moderate turbulence and 111 severe turbulence). Both GBC and GBR reached the accuracy rate of 0.9937. The F1-score for GBC and GBR are 0.7336 and 0.6850, respectively. Benefiting from the use of gradient boosting methods, feature importance ranks are given for both methods. Vertical wind, acceleration, vertical speed, and fuel flow are recognized as the most important flight parameters for turbulence prediction.

One questionable aspect of this work is that the authors predict turbulence 10 seconds before it occurs. As an alerting function, an earliness of 10s is too short to notify and secure passengers. Furthermore, using overall accuracy and F1-score (obtained by averaging the F1-score of three classes) as evaluation metrics is not appropriate as the data used for both training and test are highly imbalanced (light turbulence: moderate turbulence: severe turbulence $= 0.9945 : 0.0050 : 0.0005$ in test set). As light turbulence cases account for 99.45%, the overall accuracy can not fairly represent the model capability of predicting moderate and severe turbulence. Likewise, adding the F1-score of light turbulence to the calculation of the overall F1 score will push up the average F1-score thus does not reflect the model performance for predicting turbulence.

2.3 Machine learning approaches for time series data

The problem of turbulence early identification can be categorized as a times series early classification problem or a times series anomaly detection problem, depending on the technical approaches to tackle this problem. Treating the turbulence flights as anomaly, the unsupervised time series anomaly detection techniques can be applied to identify anomalies in time series. Similarly, the supervised time series classification as well as early classification methods can also be utilized to classifying 'turbulence' and 'normal flights'.

2.3.1 Times series classification and early classification

Conventional machine learning classification techniques, designed to work with structured data, are not consistently compatible with unstructured data like time series, where successive time points exhibiting a high degree of correlation (Faouzi 2022). For example, logistic regression assumes independence between input features (Hosmer Jr 2013) (time points in time series data) which neglects the temporal correlation between time points.

Numerous research studies have been carried out to address the challenges of time series classification. The combination of the K-Nearest Neighbor (KNN) classifier and the Dynamic Time Warping (DTW) as distance metric (Xi 2006) is commonly viewed as the benchmark algorithm in time series classification, where the advantage of DTW compare to Euclidean distance lies in its capability of compares the values of both time series at different time points.

A time series shapelet (Ye 2011) is a subsequence within a time series that holds discriminative information for differentiating classification labels. It can be regarded as signatures that capture distinguishing features of different classes of time series. In the context of time series classification, shapelet transform (Lines 2012) (Hills 2014) (Bostrom 2017) extract the best shapelets from the training data set. Following their work, there have been efforts that enable learning shapelets directly without needing to extracting numerous shapelet candidates (Grabocka 2014). As shapelets are related directly to raw time series, a key strength of shapelet approaches lies in their interpretability.

Some research works aims to transform tree-based methods and applying them to time series data. Time series forest (Deng 2013) randomly segmented time series into intervals where the mean, the standard deviation and the slope are extracted. In this manner, the time series data is transformed into structured data, and then the random forest algorithm is employed for classification.

Dictionary-based approaches discretize time series (by Aggregation approXimation (SAX) in Bag of patterns (BOP) algorithm (Lin 2012)) or discretize Fourier coefficients (by Symbolic Fourier Approximation (SFA) in Bag-of-SFA-Symbols (BOSS) (Schäfer 2015)) into symbol sequences. Words are then extracted from these sequences through a sliding window and the frequency of each word is computed in the dictionary to form a word histogram of the time series, where a classifier is finally applied on.

Early classification of time series aims to predict the class label of incoming time series as early as possible (Mori 2019). It is applied in many domains such as earthquake warning (Fauvel 2020) and gaz leakage detection (Hatami 2013). To avoid serious consequences, atmospheric turbulence should be correctly identified with a certain amount of time in advance. As reviewed in (Gupta 2020a), several shapelet based and prefix based multivariate early classification methods have been proposed in recent years.

Shapelet based methods search for time series subsequences that are representative of a class and appear early in time (Xing 2011; He 2015). (Ghalwash 2012) developed a multivariate shapelets detection method for multivariate time series early classification. The proposed method extracted core shapelets that have the same start and end points for all variables. (Ghalwash 2013) proposed interpretable patterns for early diagnosis, which allowed multivariate shapelets with different start and end points. (He 2015) obtained key shapelets for each variable separately and dealt with within-class imbalance problems by clustering shapelet candidates. (He 2020) further proposed a method with interpretable rules which selected key shapelets by estimating the confidence of early classification for multivariate time series. To generate shapelets from categorical time series, (Lin 2015) proposed an early classification approach for multivariate time series that contained categorical variables along with numerical variables. To address class imbalance problem in a multivariate time series early classification scenario, (He 2013) combined an under-sampling method with shapelet based approaches and proposed an early prediction on imbalanced multivariate time series.

Prefix-based methods are dedicated to first learn minimum prediction length of time series and then utilize it for early classification(Xing 2009). (Ma 2017) transformed multivariate time series into univariate time series by center sequence and executed the early classification on piecewise aggregate approximation representations. (Gupta 2020b) focused on multivariate time series from sensors with different sampling rates and proposed a divide-and-conquer based method.

Equipped with thousands of sensors on board commercial aircraft, the evolution of flight variables as well as their relationship may provide valuable information indicating upcoming turbulence.

Most of the research work on time series classification as well as time series early classification is dedicated to exploring univariate time series classification (He 2015; Faouzi 2022). Most existing multivariate time series early classification methods do not consider the relationships among variables (Ma 2017; Gupta 2020a). An early classification method (He 2015) utilized the combination of core shapelets to reflect the connection between variables, but the relation among variables is not considered before shapelet extraction.

2.3.2 Times series anomaly detection

According to anomaly types and the number of variables, the times series anomaly detection algorithms are proposed in the literature to detect anomaly points, anomaly subsequences, or anomaly time series for univariate time series or multivariate time series data (Blázquez-García 2021). While most time series anomaly detection methods are dedicated to univariate time series data and anomaly point detection, fewer methods handle multivariate time series and anomaly subsequences (Blázquez-García 2021). Even though it is possible to apply univariate method on each variable independently, such a way will miss the information about dependencies between variables. To leverage univariate anomaly detection methods, some multivariate methods first find or represent multiple variables as a set of uncorrelated variables and then apply univariate techniques. The dimensionality reduction techniques include clustering based methods (Wang 2018), Functional Principal Component Analysis (Ramsay 2008) and functional shape features (Srivastava 2016). Multivariate time series anomaly detection methods include local outlier factor (LOF) (Breunig 2000), one-class support vector machine

(OCSVM) (Manevitz 2001) (Schölkopf 2001), isolation forest (Liu 2008), functional isolation forest (Staerman 2019) and deep learning based methods (Zenati 2018) (Su 2019), etc.

Although many application scenarios of time series anomaly detection require algorithms to operate in real time, no study has been done to analyze the computational cost of anomaly detection methods for their possibility to be used in real-time contexts (Blázquez-García 2021).

2.4 Conclusion

Even though many studies has been conducted in both industry and academia, there is still no precise and reliable in-flight turbulence prediction method. Pilot reports and EDR algorithms provide turbulence observations but not their real time prediction. On-board weather radar only detects turbulence in wet air without any precise announcement about the timing. Turbulence forecasting methods are difficult to precisely predict turbulence due to limited spatial and temporal resolution. Considering the growing severity of the atmospheric turbulence phenomenon and the lack of the precise prediction methods, we are committed to filling this gap and investigating new approaches for precise turbulence early identification. Equipped with a large number of sensors coming from different aircraft systems, we believe that the flight variables (high frequency multivariate time series generated by sensors) as well as their relationships contain useful information indicating upcoming turbulence. Most existing multivariate time series early classification methods, however, do not consider the relationship among variables. Regarding time series anomaly detection methods, although many application scenarios require algorithms to operate in real time, no study has been done to analyze the computational cost of anomaly detection methods for their possibility to be used in real-time contexts. To tackle the above-mentioned problems, we propose a machine-learning-based turbulence early identification method. The proposed method not only makes real-time turbulence prediction become possible due to its recursive nature and low computation complexity, but also captures the dynamic relation between multiple variables which help identify and predict turbulence.

Chapter 3

Offline Method: Shape features for turbulence detection

Table of contents

Abstract.

In this chapter, functional shape features combined with the state-of-the-art time series anomaly detection methods are utilized for turbulence early identification. We first represent raw time series as functions which enable not only discovering the underlying function behind raw measurements but also implicitly removing data noise. Functional geometry features, which can capture the dynamic relation between variables, are deduced from the multidimensional path in functional representation. Based on the transformed geometry features, a time series anomaly detection method is further deployed to detect specific behaviors indicating upcoming severe turbulence. We also proposed a method to select important variables for multivariate time series based on the functional shape feature arc length.

3.1 Introduction

As shown in Figure 1.4, flight variables exhibit strong fluctuation in the turbulence area. Atmospheric turbulence can be detected if these specific behaviors are early identified. In this work, the problem of turbulence detection is cast as a problem of rare event early classification task. We use time series data from sensors for this.

Both flight variables and their relationships change before encountering turbulence, which may provide valuable information indicating upcoming turbulence. To early detect atmospheric turbulence, we propose to utilize both flight variables and their relationships.

Most existing multivariate time series early classification methods do not consider the relationships among variables (Ma 2017; Gupta 2020a). To address this problem, we early classify upcoming turbulence based on functional shape features extracted from the multidimensional path in functional representation, where each shape feature exhibits a different relationship between variables.

3.2 Methodology

Functional Data Analysis (FDA) aims to represent high dimensional discrete observations as functions (Ramsay 2008). Observations from high rate sensors deployed in commercial aircraft are real continuous functions observed at discrete time points, thus can be treated as functional data. As sensor signals exhibit particular behaviors in the turbulence zone, turbulent signals are considered as outliers in magnitude and shape compared with normal sensor signals and multivariate functional outlier detection methods (Srivastava 2016; Staerman 2019; Lejeune 2020) are applied to identify turbulence several seconds or minutes in advance.

The overall workflow (See Figure 3.1) can be divided into four steps: functional approximation, multidimensional representation, functional shape features and anomaly detection with functional isolation forest. For the sake of simplicity, a bivariate time series data set from (Ramsay 2008) is employed to illustrate the framework.

1. Functional approximation.

The discretely recorded data collected from a sensor on board of commercial aircraft can be represented as (t_i, y_i), with y_i the observation value at time t_i. By assuming the existence of a 'smooth' function generating the discretely recorded data, the discretely recorded data can be treated as functional data. The smooth property of the underlying function is critical: it ensures that the underlying function is derivable and one can take advantage of the information contained in its derivatives. Besides, representing the raw data as functions can also implicitly filter out data noise.

Data coming from multiple sensors are observed over multiple dimensions of time thus can be regarded as multidimensional functional data. To find the underlying function, functional approximation is applied to high dimensional functional data and represents the latent function by linear combination of basis functions (see figure1.5). Basis functions are a group of known functions and the common used basis function systems includes monomial basis, Fourier basis and b-spline basis (Ramsay 2008). For non-periodic functional data, b-spline functions are the most common choices for functional approximation (Ramsay 2008).

As the approximation function is the weighted sum of basis functions, it can be determined once we obtain the coefficient of each basis function. To estimate the coefficients, a familiar technique is to minimize the weighted sum of squared errors (with roughness penalty) between estimated functions and functional data (Ramsay 2008).

Assume that we have an univariate raw time series with N sampling points. We approximate the time series data $\{y_t\}$, $t \in [1, N]$ through J b-spline basis functions h_j, $j \in [1, J]$. The coefficients x_j, $j \in [1, J]$ for basis functions h_j can be calculated by minimizing least square criteria (Ramsay 2008):

$$J(x) = \underset{x}{argmin} \sum_{t=1}^{N} (y_t - \sum_{j=1}^{J} x_j h_j(t))^2 \tag{3.1}$$

The estimate $\hat{x}$ can be computed as:

$$\begin{aligned} \hat{x} &= \underset{x}{argmin}(||y - HX||^2) \\ &= (H^T H)^{-1} H^T y \end{aligned} \tag{3.2}$$

Where the matrix $H = \{h_j(t)\}_{j \in [1,J], t \in [1,N]}$ contains the values of all basis functions at sampling points. $X = \{x_j\}$, $j \in [1, J]$ is the coefficient vector.

To avoid over fitting, roughness penalty is added to balance two conflicting goals: bias and variance, at the same time.

$$J(x) = \underset{x}{argmin} \sum_{t=1}^{N} (y_t - \sum_{j=1}^{J} x_j h_j(t))^2 + \lambda \int [D^q \hat{f}_m(t)]^2 dt \tag{3.3}$$

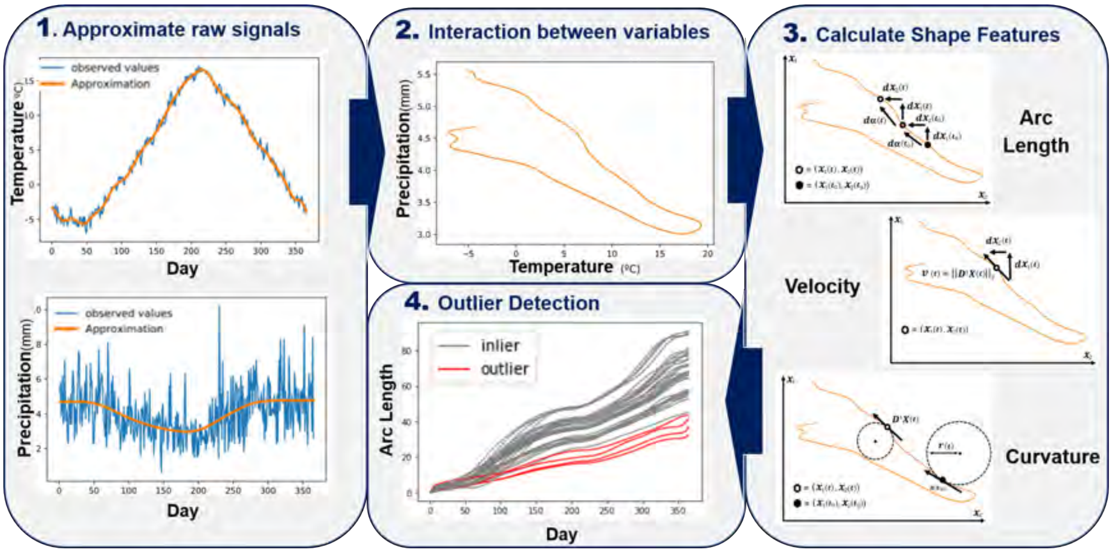


Figure 3.1: Geometry based functional outlier detection method framework. The images for the first, second and fourth steps come from (Bernard 2020) and the illustrations for the third step are inspired by (Lejeune 2020). In the first step, raw signals (blue lines) are approximated by b-spline basis functions (orange lines). Then, in the second step, two variables are represented as a path in a two-dimensional space. In the third step, geometry shape features are extracted from the bi-dimensional path curve. This operation allows the shapes of the curves to capture the important relationships between variables to classify the time series. Then finally, in the fourth step, functional isolation forest is applied based on a shape feature to detect outliers.

$\hat{f}_m(t) = \sum_{j=1}^{J} x_j h_j(t)$ is the approximation function and $D^q = \frac{d^q}{dt}$ is the q^{th} derivative. We referred to theorem 2.18. and theorem 3.15. in (Lyche 2008) for the calculation of b-spline function values h_j, $j \in [1, J]$ and their derivatives. λ is a hyper-parameter that controls the weight of the penalty.

$\hat{x}$ can be obtained by equating the first derivative of $J(x)$ to 0 with with respect to the vector X that contains all the basis functions' coefficients:

$$\begin{aligned} \hat{x} &= \underset{x}{argmin}((||y - HX||^2) + \lambda X^T R X) \\ &= (H^T H + \lambda R)^{-1} H^T y \end{aligned} \tag{3.4}$$

Where the matrix $H = \{h_j(t)\}_{j \in [1,J], t \in [1,N]}$ contains the values of all basis functions at sampling points. $X = \{x_j\}, j \in [1, J]$ is the coefficient vector. $R = \int D^q h_j(t) D^q h_l(t) dt$ with $j, l \in [1, J]$.

2. Path lying in M-dimensional space. Suppose that we have a multivariate time series with M variables (dimensions), each variable can be approximated by a linear combination of basis functions. Once the raw curves are approximated by smooth functions, a space is formed by combining all the variables, where each variable represents an axis of the multidimensional space. Such representation is adopted in (Lejeune 2020) to incorporate multivariate time series. In this space, the state evolving over time traces a path through the M-dimensional space thus each point represents a system's state. Figure 3.1 shows an example of a two-dimensional space plotted by combining two variables.

3. Functional shape features. As the path lying in the multidimensional space is the representation of all variables, its shape reflects the relationship between variables. Therefore, an underlying shape outlier in the path curve corresponds to a change of the relationship between variables. To capture shape features, various aggregation functions are applied, where each aggregation function extracts a shape feature and each shape feature exhibits a different aspect of relationship among variables. Aggregation functions are used in (Srivastava 2016) to analyze multidimensional curves and the following shape features are adapted in this work as they have been successfully applied in (Lejeune 2020) for multivariate time series outlier detection.

Arc length Suppose that $f(t)$ is a m-dimensional continuous function of variable t, the arc length $\alpha(t)$, which is the cumulative length of $f(t)$, can be calculated as follows:

$$\alpha(t) = \int_{t_0}^{t} \left\| D^1(f(u)\right\| du = \int_{t_0}^{t} \sqrt{\sum_{k=1}^{m} \frac{\mathrm{d}f(u)^2}{\mathrm{d}u}} du \tag{3.5}$$

where $\|.\|$ is the l2-norm in R^m. Arc length is a positive increasing function which enable detecting the change of curve length. Thus, whether it is a continuous fluctuation or a sudden overshoot, as long as an outlier causes changes in the curve's length, it can be effectively revealed by arc length.

Velocity Velocity shows the instantaneous change of the curve and can be regarded as the curve's slope at instant t

$$v(t) = \left\| D^1 f(t)) \right\| \tag{3.6}$$

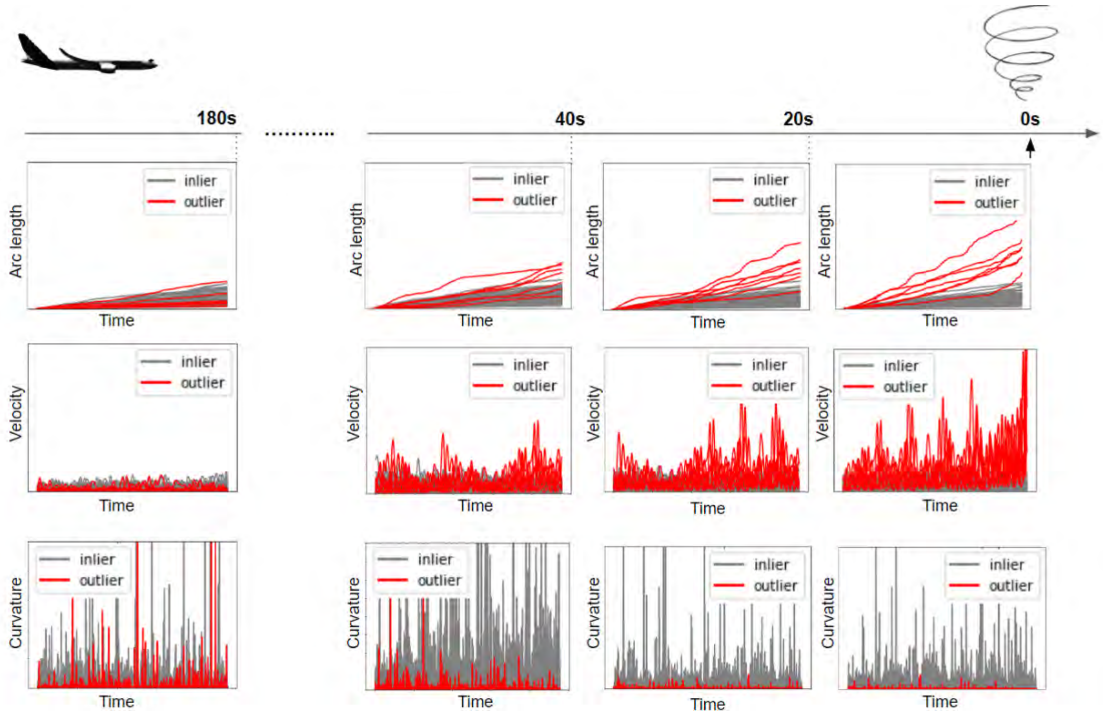


Figure 3.2: Evolution of arc length, velocity and curvature before encountering atmospheric turbulence. Red lines are flights that will encounter turbulence while grey lines are flights that will not. The closer to the turbulence, the bigger the difference between the red lines and the grey lines for all three shape features, which indicates the upcoming turbulence.

Velocity can also be computed from arc length: $v(t) = \frac{\mathrm{d}\alpha}{\mathrm{d}t}$ or $\alpha(t) = \int_{t_0}^{t} v(t)dt$. Velocity is an instantaneous variable that reflects the rate of change of arc length.

Curvature Curvature shows how quickly the tangent vector changes its direction. More intuitively, its inverse is the radius of the curve at instant *t* and can detect outliers in bended shapes. Curvature is defined by the following equation:

$$\kappa(t) = \frac{\left\| D^1(\frac{D^1(f(t))}{\|D^1(f(t)\|}) \right\|}{\|D^1(f(t)\|} \tag{3.7}$$

where $\frac{D^1(f(t))}{\|D^1(f(t)\|}$ is the normalized tangent vector and $\left\| D^1(\frac{D^1(f(t))}{\|D^1(f(t)\|}) \right\|$ can be regarded as its rate of change. Therefore, curvature is the rate of change of the tangent vector in relation to the normalized tangent vector.

Example. Figure 3.2 illustrates how these shape features are used for turbulence early identification. It provides an example of the evolution of arc length, velocity and curvature before encountering atmospheric turbulence. A 60 seconds sliding window with a step size of 10 seconds is applied. Red lines are flights that will encounter turbulence while grey lines are flights that will not. The closer to the turbulence, the bigger the difference between the red lines and the grey lines for all three shape features, which indicates the upcoming turbulence. Moreover, it also shows that arc length, velocity and curvature exhibit the information of incoming turbulence and the closer to turbulence, the easier to predict its occurrence.

Based on shape features, functional isolation forest is applied to further detect outliers.

4. Functional isolation forest. Functional isolation forest (FIF) (Staerman 2019) extends isolation forest (IF) (Liu 2008) from finite dimensional observations to functional data (Staerman 2019). Based on the same principle of constructing isolation trees as IF, FIF projects the functional data onto a finite dimensional function space by introducing a dictionary and a scalar product. The dictionary consists of a series of functions (deterministic functions, stochastic functions or observations themselves). The scalar product of the observation function and elements in dictionary provides a rich representation of the observation function. Different dictionaries (e.g. cosine dictionary, brownian bridge dictionary, mexican hat wavelet dictionary) describes different properties of original observations thus the choice of dictionary plays a key role (Staerman 2019). To split a node, FIF first samples an element from the predefined dictionary. Then, a split value is uniformly drawn from the interval defined by the maximum and minimum values of the projections of the dictionary element and observations. Same as IF, the instance that has a shorter average path length will have a larger anomaly score, as it is easier to isolate from all others.

Based on a shape feature, FIF returns an anomaly score for each instance; i.e., the bigger the score, the shorter the path, and the more likely the instance is an outlier (more likely that a flight will encounter severe turbulence in our case). How to set the contamination threshold is tricky; i.e., a low threshold will cause more false negative cases whereas a high threshold will lead to more false alarms (false positive cases). For the purpose of having a robust turbulence alerting system, a relative low threshold is desired to eliminate false alarm.

3.3 Parameter selection by arc length

Coming from different aircraft systems, hundreds of sensors are equipped on board of a commercial aircraft. Taking all the parameters into consideration for turbulence prediction will create a huge computational burden for on board aircraft control computer. We therefore propose to use the functional shape feature arc length to select important parameters for turbulence prediction. In addition, understanding the parameters importance can also provide insights to better understand atmospheric turbulence for both aviation practitioners and ML algorithm engineers.

Figure 3.3 shows the principle of feature selection by arc length. Assume that we have l time series and each time series has M different variables. We first compute the arc length using all M parameters. Then, we remove one parameter, recompute the arc length with the $M-1$ parameters left and calculate the euclidean distance between arc length with all the parameters and arc length with $M-1$ parameters. The variables that have larger distance gaps are considered as the important parameters as they make larger contribution to the formation of arc length. We then iterate this process over all M variables and obtained a parameter importance list where the distance difference obtained by removing one variable indicates its importance.

Algorithme 3.1 : Parameter importance by arc length

```
**Input:** A $M$-dimensional time series $\mathbf{y} = \{y_t^1, y_t^2, ..., y_t^M\}, t \in [1, N]$.
**Output:** Scores of parameters importance: $\mathbf{S} = [S^1, S^2, ..., S^M]$.
$\alpha^{\mathbf{y}} \leftarrow$ calculate arc length of $\mathbf{y}$ according to equations 3.4 and 3.5.
**for** *m* $\leftarrow$ *1 to* $M$ **do**
    $\mathbf{y}_m \leftarrow$ remove the $m$-th variable $y^m$ from $\mathbf{y}$
    $\alpha^{\mathbf{y}_m} \leftarrow$ calculate arc length of $\mathbf{y}_m$ according to equations 3.4 and 3.5.
    $S^m \leftarrow$ EuclideanDistance($\alpha^{\mathbf{y}}$,$\alpha^{\mathbf{y}_m}$)
**end**
```

Algorithm 3.1 illustrates how parameters importance is calculated for a multivariate time series $\mathbf{y}$. Assume that we have l multivariate time series in the dataset, the overall parameter importance of the m-th variable can be obtained by averaging S^m through all l time series.

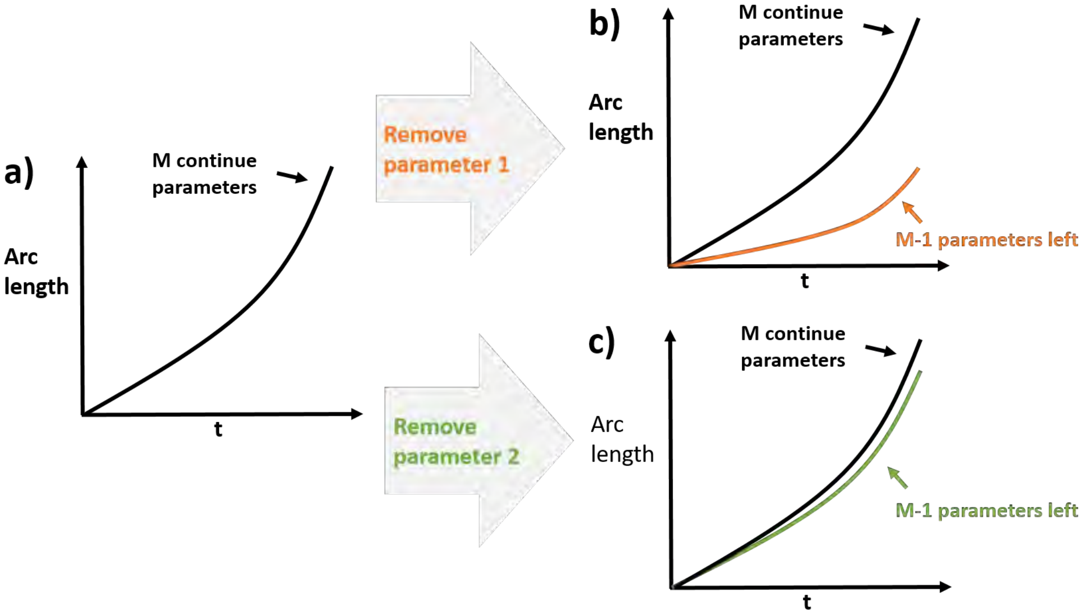


Figure 3.3: Parameter importance by functional shape feature arc length. The black curve in both a), b) and c) shows the arc length calculated using all M parameters. The orange curve in b) shows the arc length calculated using $M-1$ parameters after removing parameter 1. The green curve in c) shows the arc length calculated using $M-1$ parameters after removing parameter 2. The euclidean distances are calculated between arc length with all the parameters and arc length with $M-1$ parameters. The larger the distance, the more important this parameter is.

3.4 Conclusion

In this chapter, we use multivariate time series data coming from aircraft on board sensors and identify future turbulence by a functional shape-based outlier early detection method. This approach can not only capture the dynamic relation between variables, but also allows us to detect the upcoming turbulence information based on the shape of the transformed geometry features. Such a method, however, demands the availability of the entire time series to perform functional approximation by least squares, which limit its use in real-time applications. In the next chapter, we will present how to approximate time series and extract functional shape features in a recursive way for real time applications.

CHAPTER 4

FUTURA

Table of contents

Abstract.

In this chapter, we introduce FUTURA for FUnctional shape feature for real time TURbulence Alerting. Based on the offline method from chapter 3, FUTURA adapts a steady state Kalman Filter to approximate raw signals and extract functional shape features in a recursive manner. FUTURA not only makes real-time turbulence prediction become possible due to its incremental nature and low computation complexity, but also inherits the advantage of the offline method by capturing the dynamic relation between multiple variables which helps identify and predict turbulence.

4.1 Introduction

Offline method use least square to calculate the coefficients of basis function which demands the availability of the entire time series. Although it is possible to apply the offline method in real time by adding a sliding window, such a way of approximating raw signal needs to store all the data in the sliding window which is not desirable considering the strict memory limit of aircraft embedded control computer. To tackle this problem, we propose a method called FUTURA for FUnctional shape feature for real time TURbulence Alerting.

In this chapter, we show the principle of FUTURA and its improvements in both memory requirement and time complexity compared to the offline method presented in the chapter 3.

4.2 Methodology

FUTURA is a data-driven approach to predict severe turbulence 30 seconds in advance based on the aircraft on board sensor data. It first applied steady state Kalman Filter (Simon 2006) on functional data approximation (Ramsay 2008) (Jauch 2017) to extract functional shape features (Srivastava 2016) in an incremental way. Then, a time series anomaly detection method such as functional isolation forest (Staerman 2019) is applied on the generated shape features to identify future severe turbulence.

The overall workflow of the proposed method can be divided into 4 steps: functional approximation, multidimensional representation, shape feature extraction and anomaly detection with functional isolation forest. A visual illustration of the first three steps can be found in figure 4.3. Both FUTURA and the offline method share the second step of multidimensional representation, the third step of shape feature extraction and the fourth step of anomaly detection. The main difference of the two methods is how they approximate raw signals: the offline method applies least square on the entire time series and FUTURA uses steady state Kalman Filter to approximate raw signal in an recursive way.

4.2.1 Functional approximation by steady state Kalman Filter

In the first step, steady state Kalman Filter is applied to approximate raw signals by calculating corresponding basis functions coefficients in real time.

4.2.1.1 Functional approximation

According to chapter 3, the entire raw time series can be approximated by a linear combination of basis functions and the estimated basis function coefficients $\hat{x}$ can be obtained by minimizing the weighted sum of squared errors (with roughness penalty) between estimated functions and raw measurements.

Although it is possible to applied least square in real time by applying it to a sliding window, such technique is not suitable for real-time approximation of high frequency sensor data for two reasons:

1. high memory requirements. One needs to store sufficient raw data points in the embedded computer to calculate the basis functions coefficients and whenever a new data point comes, the stored data needs to be updated and then used again to calculate the basis functions coefficients by least squares.

2. high time complexity. The time complexity of estimating basis functions coefficients in each iteration is $O(C_{window}^2 N_{window} + C_{window}^3)$ where C_{window} and N_{window} are respectively the number of basis functions and the number of data points in the sliding window.

We therefore propose an incremental way of approximating functional data utilizing steady state Kalman Filter, based on the work of (Jauch 2017). The proposed method can significantly reduce the computation and storage requirements, thus better adapt to process high frequency sensor data in embedded system. The comparison of time and space complexity between our proposed method and least squares will also be illustrated in the following subsection.

4.2.1.2 Functional approximation with Kalman Filter

Kalman Filter estimates the system state from sensors measurements (See Simon's book (Simon 2006) for a detailed explanation of Kalman Filter). Based on the linear relation between the underlying function and b-spline basis functions, a discrete linear time-invariant dynamic system can be established where the coefficients of basis functions are the system states (Jauch 2017). Then, a linear Kalman Filter is utilized to estimate the coefficients of basis functions that are not observable, from the observable sensor measurement at every instant *t*.

Algorithme 4.1 : Kalman Filter (Simon 2006)

```
1 Input: $\hat{x}_{t-1}^+, P_{t-1}^+, u_t, y_t, F_t, G_t, H_t, Q_t, R_t$
2 /* Predict
3 $\hat{\boldsymbol{x}}_t^- \leftarrow \boldsymbol{F}_t \hat{\boldsymbol{x}}_{t-1}^+ + \boldsymbol{G}_t \boldsymbol{u}_t$
4 $\boldsymbol{P}_t^- \leftarrow \boldsymbol{F}_t \boldsymbol{P}_{t-1}^+ \boldsymbol{F}_t^T + \boldsymbol{Q}_t$
5 /* Update
6 $\boldsymbol{K}_t \leftarrow \boldsymbol{P}_t^- \boldsymbol{H}_t^T (\boldsymbol{H}_t \boldsymbol{P}_t^- \boldsymbol{H}_t^T + \boldsymbol{R}_t)^{-1}$
7 $\hat{\boldsymbol{x}}_t^+ \leftarrow \hat{\boldsymbol{x}}_t^- + \boldsymbol{K}_t(\boldsymbol{y}_t - \boldsymbol{H}_t \hat{\boldsymbol{x}}_t^-)$
8 $\boldsymbol{P}_t^+ \leftarrow (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t) \boldsymbol{P}_t^- (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t)^T + \boldsymbol{K}_t \boldsymbol{R}_t \boldsymbol{K}_t^T$
9 Output: $\hat{x}_t^+, P_t^+$
```

Algorithm 4.1 (Simon 2006) illustrates the five equations contained in the Kalman Filter. Compared with least square where the coefficients of all basis functions are calculated at once, Kalman Filter estimates the coefficients of the basis functions in an incremental way: algorithm 4.1 is applied on each new measurement coming from the sensor and updates only the coefficients of related basis functions (See figure 4.1).

For the two steps of Kalman Filter 'predict' and 'update', 'predict' uses the previous system state estimate to predict the current state and 'update' integrates the most recent measurement to improve the state estimate. Applying Kalman Filter to functional approximation, the 'predict' step serves to realize the change of basis functions so that a new coming data point can always be approximated by the corresponding basis function beneath it (see figure 4.1 and figure 4.2 for an example).

1. In the first equation (algorithm 4.1 line 3) of Kalman Filter,

    - $\hat{x}_{t-1}^+$, the estimated a posteriori system state vector at time step $t-1$, is the output of the previous iteration of Kalman Filter and it contains the coefficients of basis functions which are used to approximate measurement y_{t-1}.
    - $\hat{x}_t^-$, the a priori system state vector at time t, is a predicted estimate of the system state before the measurement y_t is introduced into the system. Given d as the degree of the basis functions, the system state vector (both $\hat{x}_{t-1}^+$ and $\hat{x}_t^-$) are $(d+1) \times 1$ dimensional column vector (see figure 4.1).
    - F_t is the state transition matrix, G_t is the input transition matrix (G_t is set to be a $(d+1) \times (d+1)$ identity matrix for functional approximation) and

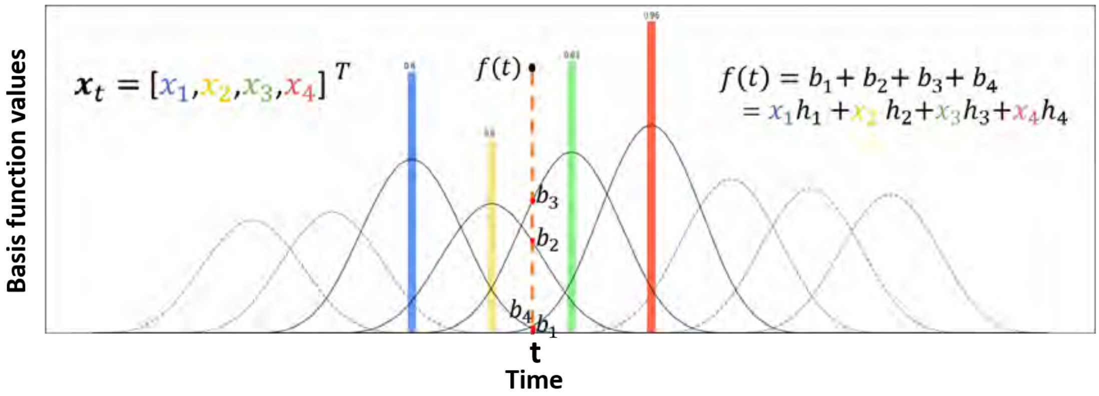

Basis function values
$x_t = [x_1, x_2, x_3, x_4]^T$
$f(t)$
$f(t) = b_1 + b_2 + b_3 + b_4$
$= x_1 h_1 + x_2 h_2 + x_3 h_3 + x_4 h_4$
b_3
b_2
b_4
b_1
t
Time

Figure 4.1: Functional approximation with b-spline basis functions. Functional approximation at time t. The degree of the basis function d is set to 3 in this example. Only $d+1$ nonzero basis functions contributes to approximate the measurement at time t, while the function values of other basis functions are all zero. The solid black lines are 4 b-spline basis functions that contribute to the approximation the measurement y_t at time t. The four colored vertical bars x_1, x_2, x_3 and x_4 correspond to the coefficients of four b-spline functions. h_1, h_2, h_3 and h_4 are the initial values of the basis functions at time t. The functional approximation $f(t)$ of y_t are the linear combination of basis functions values at time t.

u_t is the deterministic input variable to the system. F_t, G_t and u_t transform $\hat{x}^+_{t-1}$ to $\hat{x}^-_t$.

The functionality of the first equation (algorithm 4.1 line 3) can be summarized as follows:

(a) when y_{t-1} and y_t share the same basis functions, the state transition matrix F is a $(d+1) \times (d+1)$ identity matrix and the input transition matrix u is a $(d+1) \times 1$ zero column vector. In this case, $\hat{x}^-_t$ is kept the same as $\hat{x}^+_{t-1}$.

(b) when a new basis function needs to be added to approximate y_t, F is an $(d+1) \times (d+1)$ matrix with an upper diagonal (see Equation 4.1) and $u_t = (0, ...0, 1)^T_{d+1}$. The $\hat{x}^-_t$ is generated by removing the first element in the $\hat{x}^+_{t-1}$ column vector, pushing the other elements up one position, and adding a new element at the bottom of $\hat{x}^-_t$ (see figure 4.2 for a more detailed explanation). Such way of recursively approximate raw data with b-splines was proposed in (Jauch 2017) and have been applied for real sea level monitoring (Strandberg 2019).

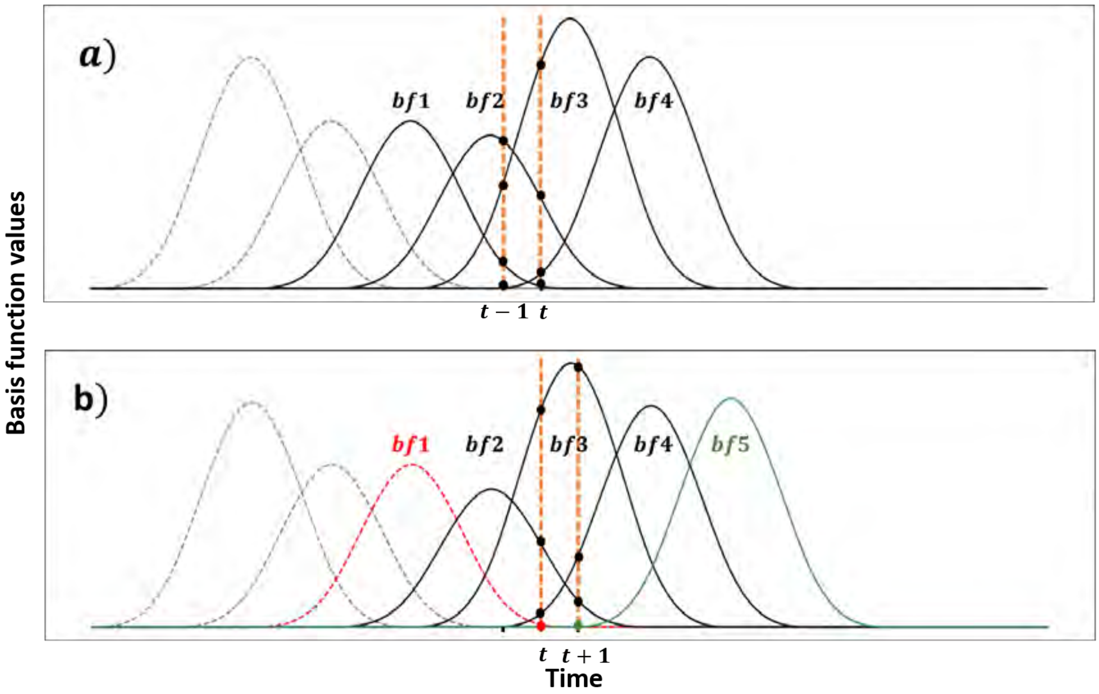


Figure 4.2: Apply Kalman Filter for functional approximation: two state transition situations This figure illustrates two different state transitions when applying Kalman Filter to b-spline approximation. a) Sampling points at time t-1 and time t shares the same basis functions $bf1$, $bf2$, $bf3$ and $bf4$. b) For the sampling point at time $t+1$, the leftmost basis function $bf1$ is removed and a rightmost basis function $bf5$ is added to approximate the measurement at time $t+1$.

$$\mathbf{I}_{upper} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}_{d+1 \times d+1} \tag{4.1}$$

In summary, the first equation in Kalman Filter realizes the change of basis functions, so that the system state vector $\hat{x}_t^-$ always contains the coefficients of $d+1$ related basis functions to approximate y_t.

2. In the second equation (algorithm 4.1 line 4) of Kalman Filter,

- P_{t-1}^{+} and P_t^{-} are the covariance matrix that describe the uncertainty in the estimates $\hat{x}_{t-1}^{+}$ and $\hat{x}_t^{-}$, respectively (Simon 2006). Both P_{t-1}^{+} and P_t^{-} are $(d+1) \times (d+1)$ dimensional matrices.
- Q_t is the process noise matrix. With the same settings as in paper (Jauch 2017), $Q_t = \bar{q}\mathbf{I}_{d+1}$ where $\mathbf{I}_{d+1}$ is the $(d+1) \times (d+1)$ identity matrix. Q_t can directly impact P_t^{-}. The larger the $\bar{q}$, the larger the value of the elements on the diagonal in P_t^{-}.

The second equation in Kalman Filter calculates a priori covariance matrix of the current state P_t^{-} from P_{t-1}^{+} and Q_t (Simon 2006). P_t^{-} is then used in the following equation to calculate Kalman gain K_t.

3. In the third equation (algorithm 4.1 line 5) of Kalman Filter,

   - P_t^{-} is the a priori covariance matrix of the current state which is calculated from the second equation.
   - R_t is the measurement noise covariance matrix which has a great impact on the smoothness of the approximation function. R_t is a r-dimensional diagonal matrix and its θ-th ($\theta < r$) diagonal element corresponds to noise level of θ-th element in y_t.

     The input measurement at each iteration y_t is a $r \times 1$ column vector $(y, 0, .., 0)_r^T$ where r is the dimension of measurements (a positive integer smaller than d) (Jauch 2017). Taking $r = 3$ as example, elements in $(y, 0, 0)^T$ are the measurements of raw signal value at time t, its first order derivative and second order derivative. As the derivatives of raw signals can not be measured directly, derivatives in y_t are set to zero to act as a smoothing effect, and the smoothness is controlled by the elements in R_t.

     There is a trade-off between bias and variance in functional approximation: we want to approximate the raw signal as closely as possible while having a smooth approximation as we will utilize the derivatives of approximation function to generate functional shape features. Therefore, diagonal elements in R_t are hyperparameters that need to be carefully tuned.
   - H_t is the observation matrix that relates x_t and y_t. H_t is a $r \times (d+1)$ matrix that contains basis functions' values and their corresponding derivatives at time t.

- K_t is the Kalman gain calculated according to the state uncertainty P_t^- and measurement noise covariance R_t.

  Considering both the measurement noise and the system model uncertainties, Kalman gain is the weight between the measurement y_t and the a prior estimate $\hat{x}_t^-$ when forming the a posteriori state estimate $\hat{x}_t^+$ (Simon 2006). Once K_t is determined, $\hat{x}_t^+$ can be computed combining y_t and $\hat{x}_t^-$.

4. In the fourth and fifth equation (algorithm 4.1 line 7,8) of Kalman Filter,

   - y_t is the sensor measurement at time t.
   - $\hat{x}_t^+$ is the a posteriori estimate of basis functions coefficients vector at time t after the measurement y_t is introduced into the system (Simon 2006). It is generated by combining the information contained in the measurement y_t and the a priori system state vector $\hat{x}_t^-$.
   - P_t^+ is the a posteriori covariance matrix that describes the uncertainty in the estimates $\hat{x}_t^+$ (Simon 2006).

Whenever a data point is passed from the sensor, the corresponding basis functions coefficients are estimated by Kalman Filter in a real time manner without waiting for the availability of the entire time series. And once the coefficients of basis functions $\hat{x}_t^+$ are estimated, the approximation function, which is the linear combination of basis functions, can be obtained. Similarly, as H_t (algorithm 4.1 line 6,7,8) contains not only basis functions' values but also their corresponding derivatives at time t, derivatives of the estimated approximation function can also be calculated in a recursive way. Based on the approximation function and its derivatives, functional shape features can be extracted in a real time manner (function shape features will be introduced in section 4.2.3).

Simplification for sensor data with fixed frequency. As can be seen from algorithm 4.1, the calculation of covariance matrix and Kalman gain K_t (line 4,6,8) does not depend on the state vector $\hat{x}_t$ and measurement y_t (line 3,7). In the case where P_t and K_t converge to steady states P_∞ and K_∞ (which means that the underlying system, the process and measurement noise covariances are time-invariant), time varying Kalman Filter can be simplified into Steady-State Kalman Filter (SSKF) (shown in algorithm 4.2) (Simon 2006).

Algorithme 4.2 : Steady state Kalman Filter (Simon 2006)

```
Input: $\hat{x}_{t-1}^{+}, y_t, H_t, F_t, u_t, K_\infty, G_t$
/* Predict
$\hat{\boldsymbol{x}}_t^- \leftarrow \boldsymbol{F}_t \hat{\boldsymbol{x}}_{t-1}^+ + \boldsymbol{G}_t \boldsymbol{u}_t$
/* Update
$\hat{\boldsymbol{x}}_t^+ \leftarrow \hat{\boldsymbol{x}}_t^- + \boldsymbol{K}_\infty (\boldsymbol{y}_t - \boldsymbol{H}_t \hat{\boldsymbol{x}}_t^-)$
Output: $\hat{x}_t^+$
```

As sensor data has a fixed sampling frequency and we generate b-spline basis functions on equidistant nodes, H_t,F_t, Q_t and u_t change periodically. In this case, H_t,F_t, Q_t and u_t have only η different variations with η the number of data points in each basis function knot interval. A knot interval is the interval between the starting points of two successive b-spline basis functions. Once the sampling frequency of sensor, the interval distance between basis function knots and the degree of b-spline are known, H_t,F_t, Q_t and u_t can be calculated before applying Kalman Filter for real time approximation without having to re-compute it at each iteration.

1. When η = 1 (there is only one data point between two basis function knots), the system is time invariant (both F_t, H_t and Q_t are constant matrices) and the model is asymptotically stable (eigenvalues of F_t lies inside the unit circle). There exists a steady state covariance P_∞ of P_t and it is reached at time $t = T$ where $||P_T - P_{T-1}|| < \epsilon$ and ϵ is a small positive real number (Anderson 2012). Therefore there exists also a steady state kalman gain K_∞ (see algorithm 4.3). One way of determining K_∞ is by numerical simulation (Simon 2006). We can iterate algorithm 4.3 and propagate K_t to obtain its converged value.

Algorithme 4.3 : Offline calculation of Kalman gain

```
Input: $P_{t-1}^+, F_t, Q_t, H_t, R_t$
/* Predict
$\boldsymbol{P}_t^- \leftarrow \boldsymbol{F}_t \boldsymbol{P}_{t-1}^+ \boldsymbol{F}_t^T + \boldsymbol{Q}_t$
/* Update
$\boldsymbol{K}_t \leftarrow \boldsymbol{P}_t^- \boldsymbol{H}_t^T (\boldsymbol{H}_t \boldsymbol{P}_t^- \boldsymbol{H}_t^T + \boldsymbol{R}_t)^{-1}$
$\boldsymbol{P}_t^+ \leftarrow (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t) \boldsymbol{P}_t^- (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t)^T + \boldsymbol{K}_t \boldsymbol{R}_t \boldsymbol{K}_t^T$
Output: $P_t^+$,$K_t$
```

2. When η >= 2, H_t, F_t, Q_t and u_t change with a period of η. As F_t, Q_t and u_t depend on whether basis functions in $\hat{x}_t^-$ change compared to $\hat{x}_{t-1}^+$ (see figure 4.2), they actually have only two different variations.

   (a) The first variation corresponds to the situation where data point at time $t-1$ and time t shares the same basis functions and we have $F_t = \mathbf{I}_{d+1}$, $Q_t = \bar{q}\mathbf{I}_{d+1}$ and $u_t = (0, ..., 0)^T_{d+1}$.

   (b) The second variation corresponds to the situation where a new basis function is needed to approximate the data point arriving at time t (see figure 4.2). We have F_t an upper diagonal identity matrix (see equation 4.1) and $u_t = (0, ..0, 1)^T_{d+1}$. As discussed in subsection 4.2.1.2, the second variation of F and u enables the update of the basis functions from t to $t+1$. The second variation of Q_t only changes the last element on the diagonal while keeping the same as the first variation: $Q_{t_{d+1,d+1}} = \bar{p}$. By setting $\bar{p}$ much larger than $\bar{q}$, the newly added basis function is given a larger state uncertainty thus the coefficient of new added basis function can deviate from initial value $\bar{x}$ and easily updated by the new measurement y_t (Jauch 2017).

It is not trivial to demonstrate the periodic convergence of Kalman gain when η >= 2 but its convergence can be illustrated by numerical simulation before the real time application.

The time complexity of applying SSKF for b-spline approximation in each iteration is $O(d^2)$ where d is the degree of basis functions. The time complexity of ordinary least square with a sliding window is $O(C^2_{window}N_{window} + C^3_{window})$ where C_{window} is the number of basis functions in the window and N_{window} is the number of data points in the sliding window. As each data point in the sliding window is approximated by $d+1$ basis functions (see figure 4.1), the number of basis functions C_{window} is always greater than or equal to $d+1$. Therefore the time complexity of b-spline approximation by SSKF is undoubtedly much smaller than least square with a sliding window. Besides, SSKF approximate raw points in an incremental way and one only needs to store and update the state vector in the previous estimation. If we use least square method, all data points in sliding window need to be stored and updated in real time application.

4.2.1.3 Real time functional approximation algorithm

Algorithm 4.4 describes the initialization step before launching SSKF to approximate raw signal in real time. $\mathcal{H}$ (calculated in lines 2-7), $\mathcal{F}$ (calculated in lines 8-12), $\mathcal{U}$ (calculated in lines 13-20), $\mathcal{K}$ (calculated in lines 21-29) contains all the possible variations of H_t, F_t, u_t and K_t, respectively (η different variations in total for each parameter).

In real time approximation (Algorithm 4.5), one can utilize $\mathcal{H}, \mathcal{F}, \mathcal{U}, \mathcal{K}$ calculated in algorithm 4.4 without the need to recompute H_t, F_t, u_t, K_t at each iteration. W is the total number of iterations for numerical simulation (a sufficiently large integer, e.g.10^4) and we iterate W times algorithm 4.3 to obtain the convergence values of K_t and stored them (η different variations in total) in $\mathcal{K}$.

Once $\mathcal{H}, \mathcal{F}, \mathcal{U}, \mathcal{K}$ are obtained from algorithm 4.4, we can apply SSKF for real time functional approximation according to algorithm 4.5. The input variable rem records the position order of the data points in each knot interval. For example, if we suppose $\eta = 3$ which means each knot interval contains 3 data points, the relative position rem of the first, second, third, fourth, fifth, sixth and seventh data point will be 0, 1, 2, 0, 1, 2 and 0. The first three data points are located in the first node interval, the fourth to sixth data points are in the second node interval, and the seventh data point is in the third node interval. Due to the periodic change, the data point located at same relative position rem within different knot intervals share the same H_t, F_t, u_t, K_t. As previously mentioned, all η variations of H_t, F_t, u_t, K_t can be calculated offline and stored in $\mathcal{H}, \mathcal{F}, \mathcal{U}, \mathcal{K}$ for real time application.

In section 4.2.1, we use SSKF to estimate the coefficients of basis functions recursively. It enables us to approximate raw sensor measurements and obtain a derivable approximation function in real time. Since there are different kinds of sensors coming from various aircraft systems, section 4.2.2 and 4.2.3 will introduce how the useful information can be extracted from multiple variables based on approximation functions. The overall framework of the method including the content of section 4.2.1, 4.2.2 and 4.2.3 is illustrated in figure 4.3.

4.2.2 Path lying in multidimensional space

While the raw signals are approximated by b-spline basis functions by SSKF in real time, a multidimensional space can be formed by combining all the variables at the same time. As shown in figure 4.3, two variables are incorporated together to form a

Algorithme 4.4 : Initialization

```
Input: $d$, $\bar{p}$, $\bar{q}$,$\bar{x}$, $\eta$, $R$, $\bar{\mathbf{k}}$,$W$
$\mathcal{H} \leftarrow \mathbf{0}^{3\times\eta\times(d+1)}$
for r ← 0 to 2 do
    for i ← 1 to $\eta$ do
        $\mathcal{H}[r,i,:] \leftarrow D^r\mathbf{B}_d(i)^T$ from theorem 2.18. and 3.15. in (Lyche 2008)
    end
end
$\mathcal{F} \leftarrow \mathbf{0}^{\eta\times(d+1)\times(d+1)}$
for i ← 1 to $\eta - 1$ do
    $\mathcal{F}[i,:,:] \leftarrow \mathbf{I}_{(d+1)\times(d+1)}$
end
$\mathcal{F}[\eta,:,:] \leftarrow F$ from equation 4.1
$\mathcal{Q} \leftarrow \mathbf{0}^{\eta\times(d+1)\times(d+1)}$
$\mathcal{U} \leftarrow \mathbf{0}^{\eta\times(d+1)}$
for i ← 1 to $\eta$ do
    $\mathcal{Q}[i,:,:] \leftarrow \bar{q}\mathbf{I}_{(d+1)\times(d+1)}$
    $\mathcal{U}[\eta,:] \leftarrow \mathbf{0}^{(d+1)\times 1}$
end
$\mathcal{Q}[\eta,d+1,d+1] \leftarrow \bar{p}$
$\mathcal{U}[\eta,d+1] \leftarrow \bar{x}$
$\mathcal{K} \leftarrow \mathbf{0}^{\eta\times(d+1)\times 3}$
$P \leftarrow \bar{p}\mathbf{I}^{(d+1)\times(d+1)}$
for i ← 1 to $W\eta$ do
    $rem \leftarrow i$ mod $\eta$
    $P, K \leftarrow Algorithm4.3(P, \mathcal{F}[rem,:,:], \mathcal{Q}[rem,:,:], \mathcal{H}[:,rem,:], R)$
    if i > $(W-1)\eta$ then
        $\mathcal{K}[i-(W-1)\eta,:,:] \leftarrow K$
    end
end
Return: $\mathcal{H}, \mathcal{F}, \mathcal{U}, \mathcal{K}$
```

Algorithme 4.5 : Real time functional approximation by SSKF

```
Input: $\hat{x}_{t-1}^+, y_t, \mathcal{H}, \mathcal{F}, \mathcal{U}, \mathcal{K}, \eta, rem$
$\hat{x}_t^+ \leftarrow Algorithm4.2(\hat{x}_{t-1}^+, y_t, \mathcal{H}[:, rem, :], \mathcal{F}[rem, :, :], \mathcal{U}[rem, :], \mathcal{K}[rem, :, :], \mathbf{I}_{(d+1)\times(d+1)})$
$rem \leftarrow rem + 1$
if $rem \geq \eta$ then
    $rem \leftarrow 0$
end
Return: $\hat{x}_t^+, rem$
```

two-dimensional space where each axis represents a individual variable.

4.2.3 Functional shape features

Same as the offline method, to capture shape features for the multi-dimensional path, we apply three aggregation functions arc length, velocity and curvature (See Chapter 3). Each aggregation function extracts a different shape feature that represents different aspects of the relationship among variables. Compared to the offline method where the shape features can be extracted only after the whole time series is approximated, FUTURA generate functional shape features 'on the fly'. For the sensor measurement y_t at time t, once its related coefficients of basis functions are estimated in real time by Steady State Kalman Filter, the functional shape features are generated directly without the complete time series.

Figure 4.3 illustrates how the functional shape feature 'velocity' is extracted from raw time series in a real time manner.

4.2.4 Functional isolation forest

Same as the offline method, FUTURA applies FIF on functional shape features to identify future turbulence (see Chapter 3 for FIF principle).

As the multivariate sensor signals are transformed into the univariate shape feature by FUTURA, apart from FIF, various state-of-the-art time series classification and anomaly detection methods can be applied on the shape feature for turbulence prediction. A comparison of the state-of-the-art time series classification as well as anomaly detection methods for their turbulence prediction performance based on the

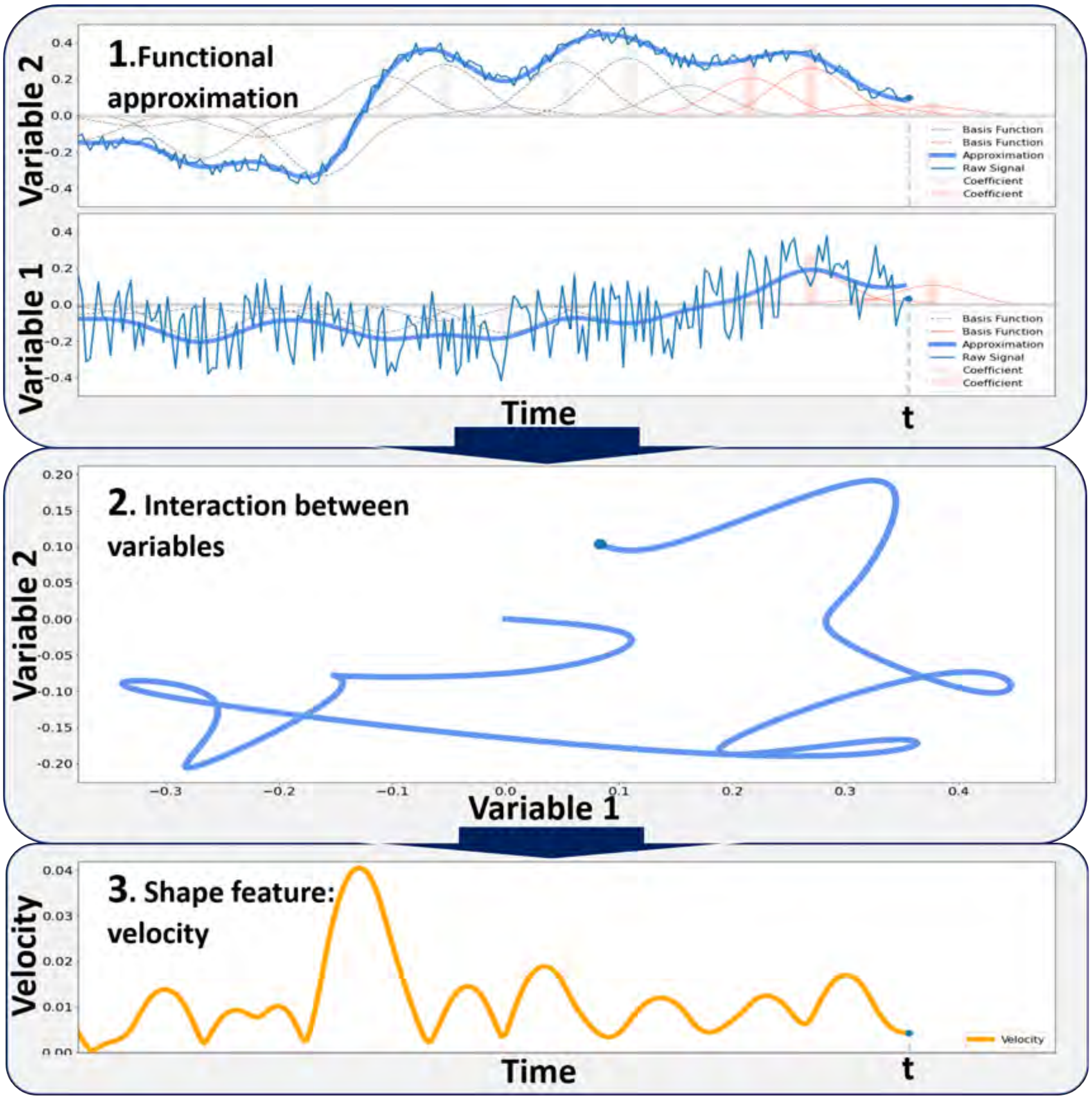


Figure 4.3: Real-time generation of functional shape feature 'Velocity'. A bivariate time series data set is generated to illustrate the framework. The degree of b-spline basis function is set to 3. In the first step, noisy raw signals are approximated by basis functions. 4 b-spline functions (pink solid lines) contribute to the functional approximation at time t and the red vertical bars are their corresponding coefficients. Grey dotted lines are the historical b-spline functions that no longer participate in the estimation of measurement at time t and the grey vertical bars illustrate their corresponding coefficients. In the second step, two variables are represented together. Such representation traces a path in a two-dimensional space where the time information is included implicitly. The trajectory and shape of this path not only reflect the information contained in each variable, but also closely reflect the relationship between them. Then, in the third step, geometry shape feature 'Velocity' is extracted from the bi-dimensional path curve in the second step. Velocity is the slope of the curve at instant and it captures the instantaneous changes between the relationships of variables.

extracted shape feature ’velocity’ is conducted and the results are presented in section 5.4.5 of chapter 5. FIF is utilized as a part of the FUTURA as it has the highest AUC value and the second highest true positive rate according to the experimental results.

4.3 Conclusion

In this chapter, we introduced FUTURA for severe atmospheric turbulence early identification. FUTURA applies steady state Kalman Filter on functional approximation as a replacement of least square to extract functional shape features and then deploy functional isolation forest to detect the upcoming turbulence. Compared to offline method, FUTURA has a lower time complexity and memory requirement. Similarly, FUTURA can also captures the dynamic relations between variables thanks to the utilization of functional shape features.

Chapter 5

Experimental Result

Table of contents

Abstract.

In this chapter we present all the experimental results and the analysis for both offline method and FUTURA. Predicting serious severe turbulence 30 seconds in advance, experimental results show that the offline approach reaches a 0.53 true positive rate while keeping a zero false positive rate and FUTURA can reaches a 0.40 true positive rate with a zero false positive rate. Although there is a performance degradation, FUTURA makes it possible for real time application and achieves a better performance (a 3% increase in AUC and a 31% increase in TPR in the case of zero FPR) than gradient boosting based methods proposed in the recent literature for turbulence early identification. It also meets the zero false alarm requirement for optimizing the passenger experience and the aircraft operational reliability. In addition, experiments are also conducted to study the influence of hyper-parameters in FUTURA.

5.1 Introduction

In chapters 3 and 4, we introduced the principles of the offline method and the online method FUTURA. In order to assess their capabilities and attributes, we carried out a range of experiments to evaluate them on real data.

We first expose our preliminary studies conducted for data exploration. We describe the sensor data used in this thesis for turbulence early identification. A study of the correlation between sensor parameters is conducted using Pearson coefficient. As atmospheric turbulence can be classified into several different categories according to their formation mechanism, to better understand the flight behavior during the turbulence, we also apply time series clustering to categorize different types of turbulence according to load factor (section 5.2).

Then, we show the turbulence prediction performance of the offline method as well as its generalization capacity by applying it to the flight data of a different aircraft model (section 5.3).

Finally,we present the turbulence prediction performance of FUTURA, its execution time for approximating raw signals as functions, the influence of hyperparameters on its capabilities, and a comparison of outcomes using different classifiers and anomaly

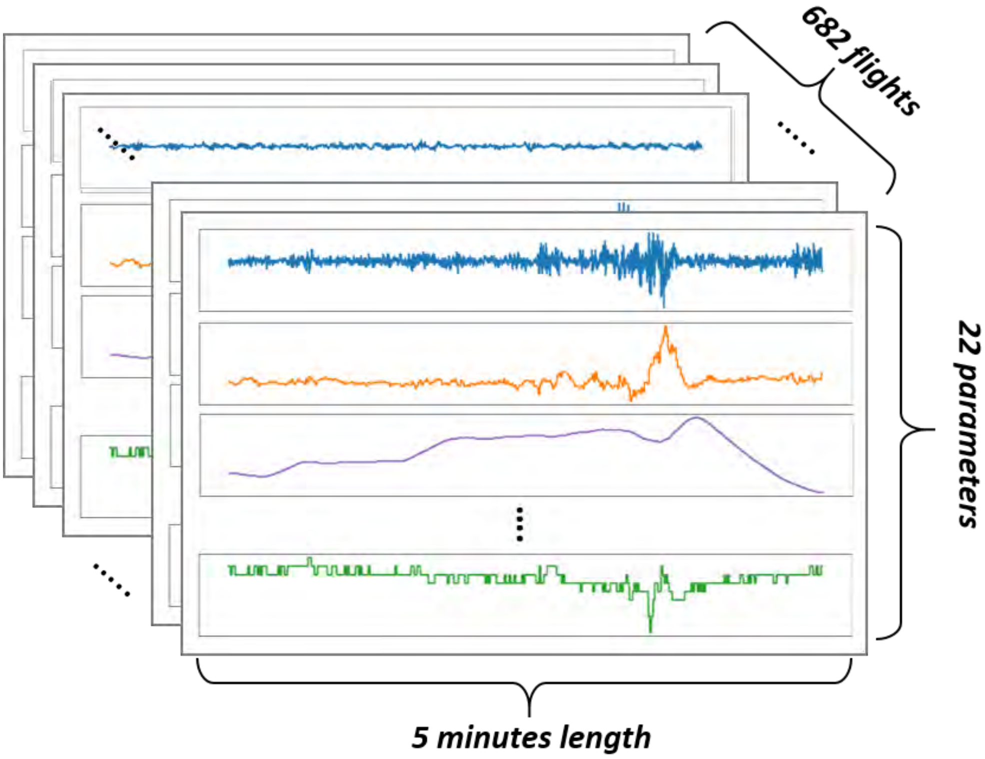


Figure 5.1: Dataset illustration. The dataset contains 682 flights in total. Each flight is a multivariate time series of 22 parameters and 5 minutes length.

detection algorithms on top of it (section 5.4).

The discussion and conclusion are presented in section 5.5.

5.2 Data Exploration

5.2.1 Dataset

The data set is collected from on-board sensors of a generic aircraft model; it contains 682 flight recordings segmented from the cruise phase of 682 different flights. Each flight recording is a multivariate time series of 22 parameters and 5 minutes length.

All the flights have been labeled by domain experts into two classes: ‘Severe Turbulence’ and ‘Normal flight’ according to load factor. There are 62 flights labelled as ‘Severe Turbulence’ and 620 labelled as ‘Normal flights’. The 22 parameters have different sampling frequencies (from 8 hertz to 64 hertz). As presented in the section of chapter 1, atmospheric turbulence can be classified as Convective induced turbulence, Low-level turbulence, Mountain wave turbulence, Clear-air turbulence and Aircraft-induced turbulence according to their nature and generation mechanisms. Among them, Clear-air turbulence, which accounts for 17% of the total turbulence cases in the whole data set, is not included in the 62 turbulence flights for experiments. The reason is that clear-air turbulence does not cause any fluctuation in all 22 flights parameters before the arrival of turbulence thus it is not predictable using the current 22 flight parameters. With the current data set, clear-air turbulence comes suddenly without any sign before.

5.2.2 Study of variables correlation

Different sensors come from different systems and measure different physical quantities. Not only their own individual values but also the relationships between them could contain useful information indicating upcoming turbulence. To better understand the dataset and prepare for turbulence prediction tasks, we first studied if there are correlations between these parameters using Pearson coefficient (Cohen 2009).

The Pearson Correlation Coefficient of two real-valued random variables a and b is defined as (Cohen 2009):

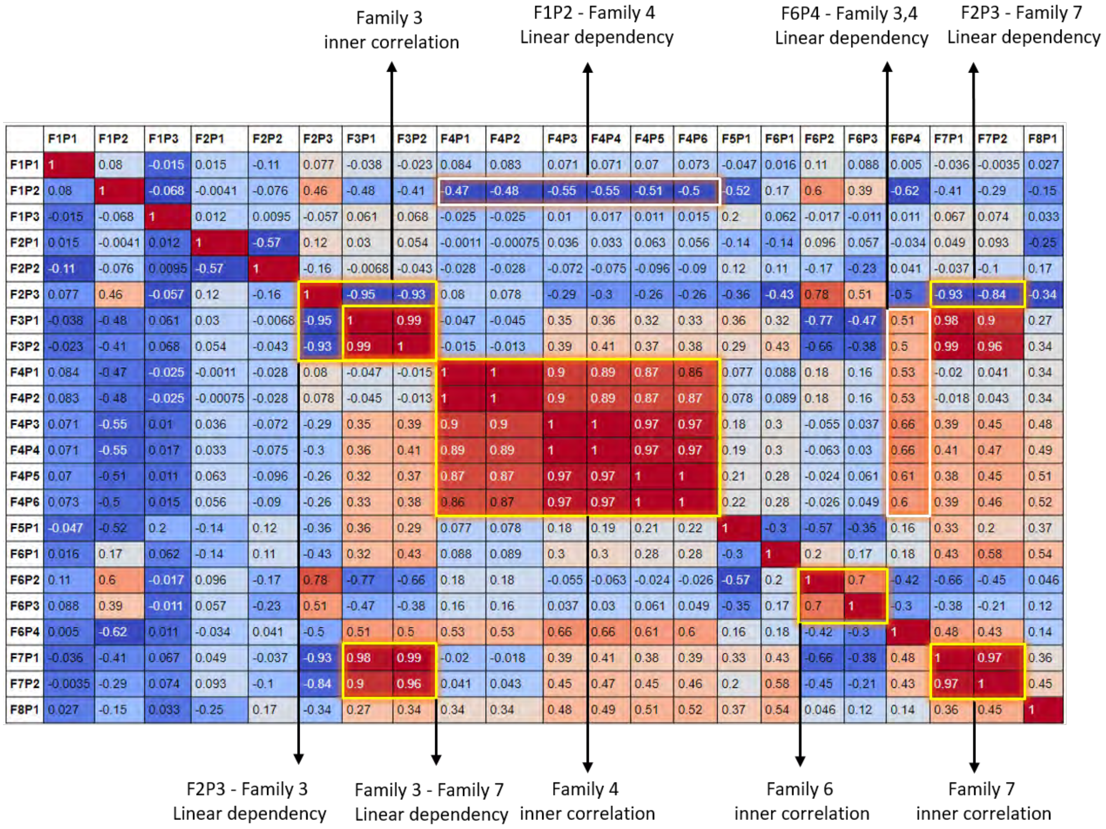
$$\rho(a, b) = \frac{cov(a, b)}{\sigma_a \sigma_b} \tag{5.1}$$

where cov is the covariance, σ_a is the standard deviation of a and σ_b is the standard deviation of b.

The Pearson Correlation Coefficient measures the linear correlation between two variables. It is a number between -1 and 1. The closer to -1, the stronger negative linear relation between two variables and vice versa for 1. A Pearson Correlation Coefficient of 0 implies that two variables have no linear dependency between them.

We classified 22 parameters into 8 different families according to their physical properties. Due to confidentiality reasons, we anonymize the names of flight parameters. A parameter named '$F1P1$' means it is the first parameter in sensor family 1. Figure 5.2 shows the linear dependency between flights parameters measured by Pearson Correlation Coefficient.

Observing the correlation matrix along the diagonal, we can find that there are strong


Family 3
inner correlation
F1P2 - Family 4
Linear dependency
F6P4 - Family 3,4
Linear dependency
F2P3 - Family 7
Linear dependency
F2P3 - Family 3
Linear dependency
Family 3 - Family 7
Linear dependency
Family 4
inner correlation
Family 6
inner correlation
Family 7
inner correlation

	F1P1	F1P2	F1P3	F2P1	F2P2	F2P3	F3P1	F3P2	F4P1	F4P2	F4P3	F4P4	F4P5	F4P6	F5P1	F6P1	F6P2	F6P3	F6P4	F7P1	F7P2	F8P1
F1P1	1	0.08	-0.015	0.015	-0.11	0.077	-0.038	-0.023	0.084	0.083	0.071	0.071	0.07	0.073	-0.047	0.016	0.11	0.088	0.005	-0.036	-0.0035	0.027
F1P2	0.08	1	-0.068	-0.0041	-0.076	0.46	-0.48	-0.41	-0.47	-0.48	-0.55	-0.55	-0.51	-0.5	-0.52	0.17	0.6	0.39	-0.62	-0.41	-0.29	-0.15
F1P3	-0.015	-0.068	1	0.012	0.0095	-0.057	0.061	0.068	-0.025	-0.025	0.01	0.017	0.011	0.015	0.2	0.062	-0.017	-0.011	0.011	0.067	0.074	0.033
F2P1	0.015	-0.0041	0.012	1	-0.57	0.12	0.03	0.054	-0.0011	-0.00075	0.036	0.033	0.063	0.056	-0.14	-0.14	0.096	0.057	-0.034	0.049	0.093	-0.25
F2P2	-0.11	-0.076	0.0095	-0.57	1	-0.16	-0.0068	-0.043	-0.028	-0.028	-0.072	-0.075	-0.096	-0.09	0.12	0.11	-0.17	-0.23	0.041	-0.037	-0.1	0.17
F2P3	0.077	0.46	-0.057	0.12	-0.16	1	-0.95	-0.93	0.08	0.078	-0.29	-0.3	-0.26	-0.26	-0.36	-0.43	0.78	0.51	-0.5	-0.93	-0.84	-0.34
F3P1	-0.038	-0.48	0.061	0.03	-0.0068	-0.95	1	0.99	-0.047	-0.045	0.35	0.36	0.32	0.33	0.36	0.32	-0.77	-0.47	0.51	0.98	0.9	0.27
F3P2	-0.023	-0.41	0.068	0.054	-0.043	-0.93	0.99	1	-0.015	-0.013	0.39	0.41	0.37	0.38	0.29	0.43	-0.66	-0.38	0.5	0.99	0.96	0.34
F4P1	0.084	-0.47	-0.025	-0.0011	-0.028	0.08	-0.047	-0.015	1	1	0.9	0.89	0.87	0.86	0.077	0.088	0.18	0.16	0.53	-0.02	0.041	0.34
F4P2	0.083	-0.48	-0.025	-0.00075	-0.028	0.078	-0.045	-0.013	1	1	0.9	0.89	0.87	0.87	0.078	0.089	0.18	0.16	0.53	-0.018	0.043	0.34
F4P3	0.071	-0.55	0.01	0.036	-0.072	-0.29	0.35	0.39	0.9	0.9	1	1	0.97	0.97	0.18	0.3	-0.055	0.037	0.66	0.39	0.45	0.48
F4P4	0.071	-0.55	0.017	0.033	-0.075	-0.3	0.36	0.41	0.89	0.89	1	1	0.97	0.97	0.19	0.3	-0.063	0.03	0.66	0.41	0.47	0.49
F4P5	0.07	-0.51	0.011	0.063	-0.096	-0.26	0.32	0.37	0.87	0.87	0.97	0.97	1	1	0.21	0.28	-0.024	0.061	0.61	0.38	0.45	0.51
F4P6	0.073	-0.5	0.015	0.056	-0.09	-0.26	0.33	0.38	0.86	0.87	0.97	0.97	1	1	0.22	0.28	-0.026	0.049	0.6	0.39	0.46	0.52
F5P1	-0.047	-0.52	0.2	-0.14	0.12	-0.36	0.36	0.29	0.077	0.078	0.18	0.19	0.21	0.22	1	-0.3	-0.57	-0.35	0.16	0.33	0.2	0.37
F6P1	0.016	0.17	0.062	-0.14	0.11	-0.43	0.32	0.43	0.088	0.089	0.3	0.3	0.28	0.28	-0.3	1	0.2	0.17	0.18	0.43	0.58	0.54
F6P2	0.11	0.6	-0.017	0.096	-0.17	0.78	-0.77	-0.66	0.18	0.18	-0.055	-0.063	-0.024	-0.026	-0.57	0.2	1	0.7	-0.42	-0.66	-0.45	0.046
F6P3	0.088	0.39	-0.011	0.057	-0.23	0.51	-0.47	-0.38	0.16	0.16	0.037	0.03	0.061	0.049	-0.35	0.17	0.7	1	-0.3	-0.38	-0.21	0.12
F6P4	0.005	-0.62	0.011	-0.034	0.041	-0.5	0.51	0.5	0.53	0.53	0.66	0.66	0.61	0.6	0.16	0.18	-0.42	-0.3	1	0.48	0.43	0.14
F7P1	-0.036	-0.41	0.067	0.049	-0.037	-0.93	0.98	0.99	-0.02	-0.018	0.39	0.41	0.38	0.39	0.33	0.43	-0.66	-0.38	0.48	1	0.97	0.36
F7P2	-0.0035	-0.29	0.074	0.093	-0.1	-0.84	0.9	0.96	0.041	0.043	0.45	0.47	0.45	0.46	0.2	0.58	-0.45	-0.21	0.43	0.97	1	0.45
F8P1	0.027	-0.15	0.033	-0.25	0.17	-0.34	0.27	0.34	0.34	0.34	0.48	0.49	0.51	0.52	0.37	0.54	0.046	0.12	0.14	0.36	0.45	1

Figure 5.2: Linear dependencies between flights parameters measured by Pearson Correlation Coefficient. Yellow boxes highlight strong linear dependencies between flight parameters while white boxes highlight the relative weak linear correlations.

linear correlations between parameters inside family 3, 4, 6 and 7. Besides, some cross-family linear dependencies can also be found in the correlation matrix; e.g. the strong linear dependencies between family 3 and family 7, $F2P3$ and Family 7. Figure 5.3 illustrate four examples of inner and cross family correlation. The inner and cross family linear dependency demonstrates the strong association between parameters within and cross sensor families, which can not only provide insights to better understand turbulence phenomenon but also demonstrate the existence of relationship among variables and help us to reduce the parameter redundancy. The limitation of Pearson Correlation Coefficient is that it can only measure linear relationships between variables in a time interval and cannot reflect the non-linear, non-deterministic and time varying relationships.

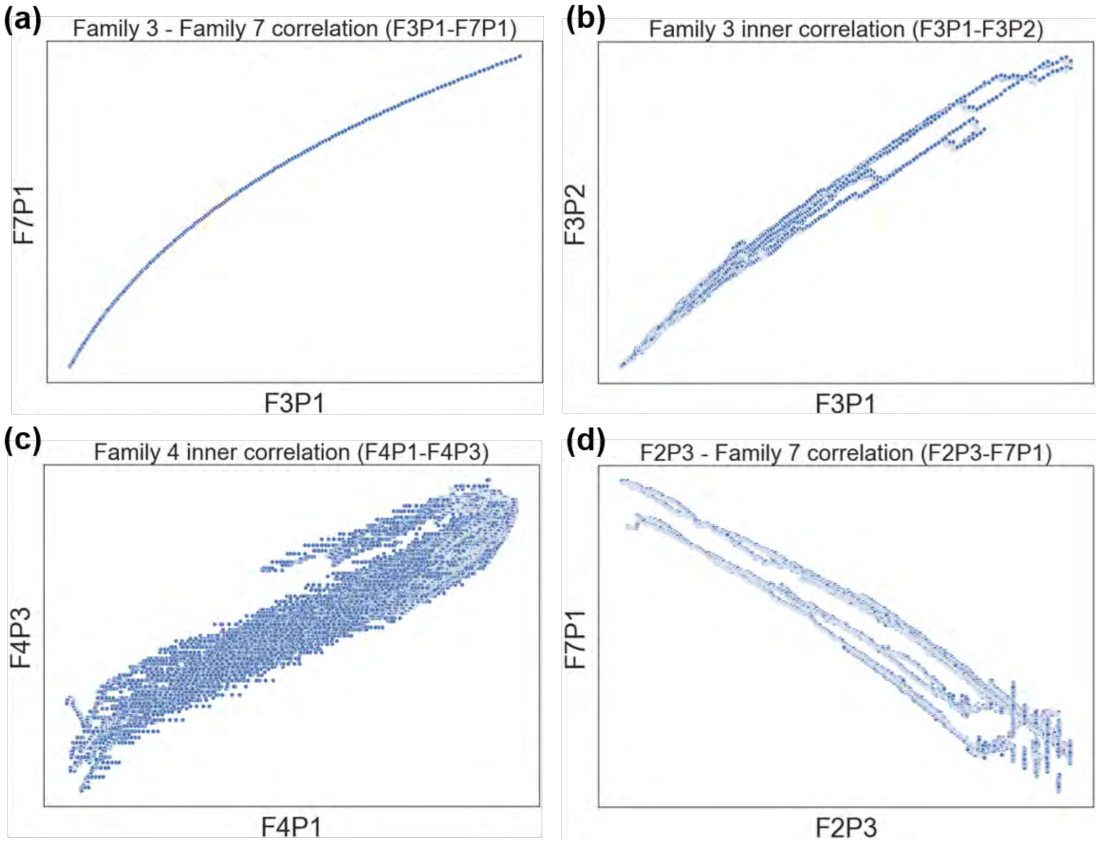


Figure 5.3: Examples of inner and cross parameter family linear dependency. (a) the relationship between family 3 and 7. (b) internal relation within family 3. (c) internal relation within family 4. (d) the relationship between $F2P3$ and family 7.

5.2.3 Time series clustering

According to the load factor, flights have different behaviors during the turbulence phase. As shown in figure 5.4, severe turbulence could happened gradually with an increasing intensity or happened suddenly after a low-intensity continuous fluctuation.

To better understand the flight behavior during the turbulence, we applied time series clustering to categorize different types of turbulence according to load factor. For this, we used the popular k-means algorithm (MacQueen 1967) with dynamic time warping (Sakoe 1978) as time series similarity measures.

Silhouette coefficient (Rousseeuw 1987) is used to evaluate the clustering performance. Based on the pairwise distance between and within clusters, Silhouette coefficient measures the quality of the clustering that does not require the ground truth of

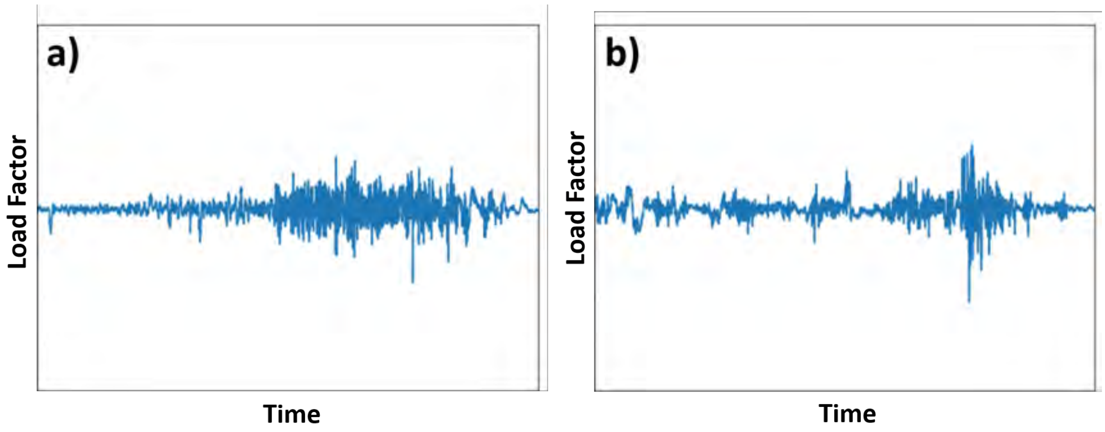


Figure 5.4: Flights can have different behaviors during turbulence phases. a) The intensity of turbulence increase gradually. b) A continuous fluctuation happens before the overshot of load factor.

Table 5.1: Experimental results of clustering on the load factor signals of all turbulence flights.

Number of clusters	2	3	4	5	6	7	8	9	10
Silhouette coefficient	0.09	0.04	-0.05	0.09	-0.07	0.003	-0.04	-0.03	0.11

time series clusters. The higher separation between clusters and cohesion within clusters, the better a clustering is. As the most common metric for clustering performance evaluation, Silhouette coefficient returns a score between -1 and $+1$ (Rousseeuw 1987). The closer to $+1$, the better is the clustering. A silhouette coefficient equals -1 indicate an incorrect clustering and a score around 0 means clusters are overlapped.

We performed the clustering on the load factor signals of all turbulence flights by setting the number of clusters from 2 to 10. The experimental result is shown in table 5.1. No matter how many cluster centers are set, it can be seen from the table that the silhouette coefficient is always around 0. Turbulence flights are not been clustered into several mutually distinct subcategories. We therefore treat all 62 turbulence flights as one category rather than design dedicated turbulence prediction algorithms for different turbulence subcategories.

5.3 Results of offline method

5.3.1 Data preparation

As mentioned in the previous section, the experiment data are collected from a general commercial aircraft model (we name it aircraft model A for convenience) and all the flights have been labeled by domain experts into two classes: 'Turbulence' (62 flights in total) and 'Normal flight'(620 flights in total).

We use the sensors' recording extracted 30 seconds before severe turbulence happens and mix them with the normal flight signals. The goal of the model is to identify whether each flight will encounter turbulence in 30 seconds. Each flight contains the recording of 22 flight variables with same length (60s) but different sampling frequencies (from 8 hertz to 64 hertz).

5.3.2 Experimental settings

All variables are standardized to share the same scale. We use 300 B-spline functions of order five as basis functions for all variables considering that the data does not exhibit periodicity. The grid search of λ_m is fixed on logarithmic scale in [-8,-6,-4,-2,0,2,4,6,8]. Three different shape features are used in the experiment: arc length, velocity and curvature. For functional isolation forest, the number of functional isolation trees is set to 100 and Brownian bridge is used as dictionary. Sub-sampling size is set to 256.

Functional isolation forest is applied on three shape features respectively. To demonstrate the effectiveness of our approach, we also perform multivariate functional isolation forest directly on all 22 sensors variables.

5.3.3 Results

We use True Positive Rate (TPR) and False Positive Rate (FPR) to evaluate the detection and false alarm rates for turbulence alerting system. Here, detection rate means the percentage of real severe turbulence events that are successfully identified in advance and false alarm rate is the percentage of erroneous alerts among all the alerts. Besides, AUC (Area Under Curve) of ROC (Receiver Operating Characteristic) is also used for the evaluation as it shows the model general classification performance regardless of skew in class distribution and the change of threshold (Bradley 1997).

Table 5.2 shows the preliminary experimental results. In this experiment, the threshold of functional isolation forest is set to 0.091, which corresponds to treating the top 62 flights with the highest anomaly scores as turbulence flight.

Compared with applying multivariate functional isolation forest directly on all 22 flight variables (see the first row of table 5.2), outlier detection based on shape feature (whether it is arc length, velocity or curvature) obtains much better result in both TPR, FPR and AUC, which demonstrates the effectiveness of our approach in early identifying in-flight turbulence. Velocity has the highest TPR and AUC as well as the lowest FPR (see the third row of table 5.2), which may be due to its ability to capture the instantaneous changes in the relationships of variables that the upcoming turbulence information exhibits.

Table 5.2: Experiment results of the offline method. The anomaly detection model is trained and tested on the data of aircraft model A. The threshold is set to 0.091.

	TPR	***FPR***	***AUC***
Multivariate functional isolation forest - All 22 flight variables	0.113	0.089	0.552
Functional isolation forest - Arc length	0.710	0.029	0.947
Functional isolation forest - Velocity	**0.742**	**0.026**	**0.959**
Functional isolation forest - Curvature	0.661	0.034	0.906

Considering our application scenarios and actual operational needs, a turbulence alerting system should have a perfect operational robustness. This requires us to maximize TPR while keeping a zero FPR.

Figure 5.5 shows the ROC curve of turbulence early classification model using the shape feature velocity. Moving up from the bottom left corner of the curve, TPR can reach up to 0.532 with a zero FPR. This can be achieved by setting the threshold to 0.0484. From an operational point of view, such result is encouraging as there is currently no accurate prediction for in-flight turbulence. The result obtained of setting threshold as 0.0484 are shown in table 5.3.

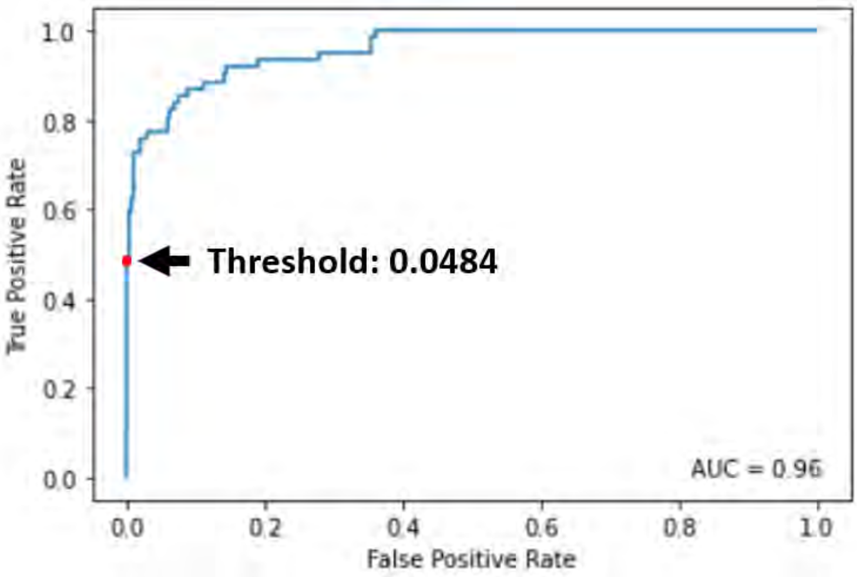


Figure 5.5: ROC curve of the turbulence early classification model using the shape feature velocity. The red point represent a 0.532 TPR with a zero FPR. This can be achieved by setting the threshold to 0.0484.

5.3.4 Application on new aircraft models

To test the generalization ability of the proposed turbulence prediction model, experiments are also conducted on a new aircraft model. More concretely, the turbulence prediction model is trained using the sensor data collected from one aircraft model (we name it as aircraft model A for convenience) and tested or applied on another (we name this new aircraft model as aircraft model B). We use the terms ’train’ and ’test’ for an unsupervised functional isolation forest algorithm. It does not signify that the true labels of the data are set as the training targets to optimize model parameters. In fact, we first save the functional isolated forest (formed by isolated trees) built on the shape feature velocity of model A flights. Then velocities from flights of model B are extracted and traverse each isolation tree of the trained model to obtain their corresponding anomaly scores.

Same as experimental setting as model A, we use the sensors recording of aircraft model B extracted 30 seconds before severe turbulence happens and merge them with the normal flight signals. The dataset of aircraft model B contains 1364 flights, among which 124 are the flights that encountered turbulence, and 1240 are the normal flights.

Table 5.3: Experiment results of the offline method. The anomaly detection model is trained and tested on the data of aircraft model A. The threshold is set to 0.0484.

	TPR	*FPR*	*AUC*
Multivariate functional isolation forest - All 22 flight variables	0.097	0.044	0.552
Functional isolation forest - Arc length	0.500	0.003	0.947
Functional isolation forest - Velocity	**0.532**	**0**	**0.959**
Functional isolation forest - Curvature	0.452	0.008	0.906

The threshold of functional isolation forest is set to 0.0484, with which we obtain the zero false positive rate in aircraft model A (we obtained 0.532 true positive rate and zero false positive rate using shape feature velocity). The experimental results are shown in table 5.4.

Table 5.4: Experiment results of the offline method. The anomaly detection model is trained on the data of aircraft model A and tested on the aircraft model B. The threshold is set to 0.0484.

	TPR	*FPR*	*AUC*
Functional isolation forest - Arc length	0.411	0.012	0.955
Functional isolation forest - Velocity	**0.460**	**0.007**	**0.964**
Functional isolation forest - Curvature	0	0.053	0.619

Comparing all three shape features arc length, velocity and curvature, velocity still obtains the best result among all three features, with a TPR of 0.460, a FPR of 0.007 and a AUC of 0.964. Compared with experimental results of velocity in aircraft model A (see table 5.3), aircraft model B have a slightly lower true positive rate and a higher false positive rate, but a higher AUC. The same conclusions hold for arc length. Curvature, on the other hand, has a noticeable performance degradation in both TPR, FPR and AUC. The reason for the performance difference may come from the difference between the two datasets: the functional isolation forest model is constructed based on the functional shape features of model A which comes from the sensor parameter values and their correlations. Model A and model B have different flight dynamic properties and different load factor distribution, resulting in different value ranges and different behaviors in functional shape features, which future lead to a shift in data distribution between two aircraft models.

Experiments are further conducted by using the data of aircraft model B for both training and testing (see table 5.5). Comparing to the result presented in table 5.4, while velocity and arc length has slight performance improvement, curvature has a huge enhancing. It is more appropriate to construct the turbulence prediction algorithm for aircraft model B using the flight data from B.

Comparing the results derived from dataset of model A (see table 5.3) and dataset from model B (see table 5.5), although TPR decreases and FPR increases on B, AUC achieves a higher value. As AUC is a general measure of model classification performance regardless of the threshold, a higher AUC means that the offline method achieves a better overall classification performance on dataset of model B. But a higher TPR and FPR also indicates that the offline method has a degradation in robustness.

Table 5.5: Experiment results of the offline method. The anomaly detection model is trained and tested on the data of aircraft model B. The threshold is set to 0.0484.

	TPR	***FPR***	***AUC***
Functional isolation forest - Arc length	0.427	0.010	0.955
Functional isolation forest - Velocity	**0.468**	**0.006**	**0.969**
Functional isolation forest - Curvature	0.315	0.022	0.896

5.4 Results of FUTURA

To demonstrate the effectiveness of FUTURA in turbulence early identification, experimental study is conducted on the same dataset (aircraft model A) as the offline method. We also compared FUTURA with Gradient Boosting Classifier (GBC) and Gradient Boosting Regressor (GBR) which were applied recently to predict atmospheric turbulence 10s before it occurs (Emara 2021). As mentioned in chapter 2, predicting turbulence 10s in advance is of limited usefulness: flight crews and passengers need some time to prepare for the upcoming turbulence and fasten the seatbelt. We therefore focus on a 30s of earliness which makes more practical sense for a turbulence prediction.

5.4.1 Comparison of FUTURA with Gradient Boosting Classifier (GBC)

5.4.1.1 Experimental settings

To train the model, we conduct a 2-fold cross-validation that is repeated 5 times with stratified sampling on the dataset. Such method of cross-validation ($5 \times 2cv$) is recommended by (Dietterich 1998) for the situation where the learning algorithm is efficient enough to execute ten times.

To measure the accuracy of the model, we calculate the True Positive Rate (TPR) and False Positive Rate (FPR) to evaluate the detection and false alarm rates for turbulence alerting system. Here, detection rate means the percentage of severe turbulence events that are successfully identified in advance and false alarm rate is the percentage of erroneous alerts among all the alerts. We then plot the Receiver Operating Characteristic (ROC) curves based on the results on the test sets and calculate the corresponding Area Under the Curve (AUC). We calculate the mean and standard deviation of the AUC over 10 iterations.

The hyperparameters of FUTURA include the degree of basis functions d, the number of sample points per knot interval η and $\overline{p}, \overline{q}$, R introduced in section 4. The shape feature 'velocity' is used in the experiment. For functional isolation forest, we use brownian bridge as dictionary. The number of functional isolation trees is set to 100 and sub-sampling size is set to 256 (the same settings as in (Staerman 2019)).

According to the paper of Emara *et al.* (Emara 2021), GBC and GBR are implemented using scikit-learn (Pedregosa 2011) and the considered hyperparameters of both methods are the number of estimators, the subsample fraction, the minimum number of samples required to split an internal node, the minimum number of samples required to be at a leaf node, the maximum depth of individual estimators and the learning rate. The best hyper-parameters configuration of both FUTURA and GBC are found by Bayesian Optimization (Gaussian processes model with expected improvement acquisition function) (GPyOpt 2016; Frazier 2018).

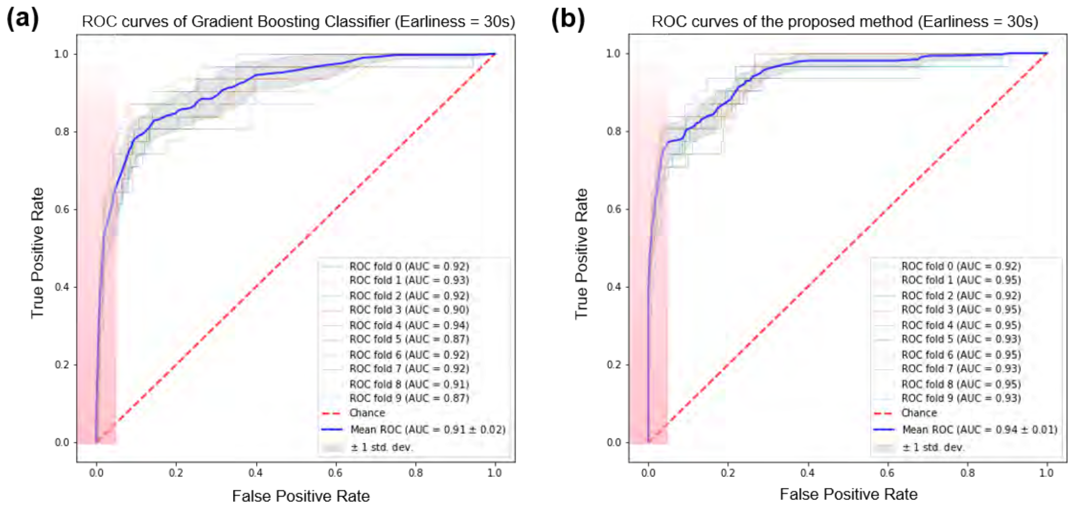


Figure 5.6: FUTURA outperforms Gradient Boosting Classifier (GBC) when the turbulence is to be predicted 30 sec prior it occurs. (a) ROC curves of GBC with earliness equals 30 sec (b) ROC curves of FUTURA with earliness equals 30 sec. The blue lines are the average of ROC curves and the grey area illustrates the ± 1 standard deviation. The light red squares show areas where FPR approaches zero. GBC achieved an AUC mean of 0.91 with a standard deviation of 0.02 while FUTURA achieves an AUC mean of 0.94 with an AUC standard deviation of 0.01.

5.4.1.2 Results and discussion

We plot ROC on the test sets and report AUC for both GBC and FUTURA (See figure 5.6). The mean curve and the ±1 standard deviation interval of ROC curves are also reported (see the blue line and the grey area in figure 5.6, respectively).

Our proposed method has a higher average AUC and lower standard deviation (0.94 ± 0.01) compared to GBC (0.91 ± 0.02). A higher average AUC shows that FUTURA has a better classification performance regardless of the choice of threshold, while a lower AUC standard deviation means that FUTURA produces more consistent classification results than GBC, thus has a better classification stability. Most importantly, the average ROC curve of FUTURA has a steeper increase in the region where the FPR approaches zero (Illustrated by the red gradient area in figure 5.6). This implies that FUTURA is more likely to have a higher TPR while keeping a very low FPR. As mentioned in the introduction, such property is critical for a robust turbulence detection method as the passengers should not be frequently alterted when there is no turbulence so that they really pay attention when there is one.

Table 5.6: FUTURA is better than Gradient Boosting Classifier (GBC) no matter how far in advance turbulence is predicted. (a) shows the TPR with zero FPR. Each column illustrates the detection rate with zero false alarm for the corresponding prediction earliness. (b) shows the TPR when FPR equals 0.005 for two methods. As there are 620 normal flights in total, a value of 0.005 for FPR means that 3 normal flights are mistaken by algorithms for future turbulence. (c) shows the AUC for FPR from 0 to 0.005, which is another measure of model performance when FPR approaches zero. For each method and each earliness value, 2-fold cross-validation are conducted on the dataset 5 times with stratified sampling. The result is obtained by averaging 10 experiments results on test sets.

(a) TPR(FPR= 0)

Earliness (sec)	10	20	30	40	50
FUTURA	**0.55**	**0.31**	**0.40**	**0.25**	**0.18**
GBC	0.33	0.13	0.09	0.18	0.05

(b) TPR (FPR=0.005)

Earliness (sec)	10	20	30	40	50
FUTURA	**0.61**	**0.40**	**0.48**	**0.39**	**0.23**
GBC	0.38	0.32	0.21	0.25	0.15

(c) AUC ($\times 10^{-3}$) (for FPR from 0 to 0.005)

Earliness (sec)	10	20	30	40	50
FUTURA	**5.93**	**3.86**	**4.44**	**3.54**	**2.55**
GBC	3.81	2.65	2.03	2.26	1.26

In Table 5.6, we compare the performance of FUTURA with GBC when predicting turbulence 10, 20, 30, 40 and 50 seconds prior it occurs. It is worth noticing that a FPR value of 0.005 is set in (b) and (c). We take this very small value of FPR to explore how the model performs when only very small false alarm rate are tolerated.

We can see from Table 5.6 that FUTURA outperforms GBC no matter how far in advance turbulence is predicted (see (a), (b) and (c)). This demonstrates that FUTURA has a better turbulence prediction performance especially when a strong operational robustness (low false alarm rate) is demanded. The reasons for the superior performance of FUTURA are three-fold: (1) Functional approximation represents raw signals by combining b-spline basis functions thus implicitly reduces the data noise. (2) Functional shape feature ’velocity’ captures the dynamic relationship among variables, which may provide valuable information indicating upcoming turbulence. (3) Functional Isolation Forest (FIF) is able to detect various shape deviations from normal data (Staerman 2019), and thus better distinguish between normal flights and ’turbulent flights’ based on ’velocity’.

Increasing the earliness gradually from 10s to 50s, the performance of both models showed a downward trend. In general, the farther away from turbulence, the less accurate the predictions.

5.4.2 Comparison of FUTURA with Gradient Boosting Regressor (GBR)

(Emara 2021) also applied Gradient Boosting Regressor (GBR) for turbulence early identification. The authors first predicted the EDR values using GBR 10s prior to turbulence happen, then classified turbulence flights by setting thresholds on the predicted EDR.

Similar to GBC, to compare the turbulence prediction performance of GBR with FUTURA, we applied GBR on flight data to predict the value of load factor 30 sec in advance. But unlike GBC, classification by the prediction results of GBR does not output class probabilities for each sample. Therefore, when GBR is applied on a test set, we can only get a fixed TPR/FPR pair and cannot obtain a ROC curve by varying the threshold on class probabilities. We therefore plotted TPR/FPR pairs obtained by GBR under 100 different configurations of GBR hyperparameters generated from random search (Bergstra 2012). In figure 5.7, each red cross represents a TPR/ FPR pair obtained by GBR under one hyperparameter configuration. For comparison, the ROC curves of FUTURA that predicts turbulence 30s piror to turbulence happen is also shown in the figure. Same as figure 5.6 (a), the blue line in figure 5.7 is the average of ROC curves of FUTURA on the 10 trials and the grey area illustrates the ± 1 standard deviation.

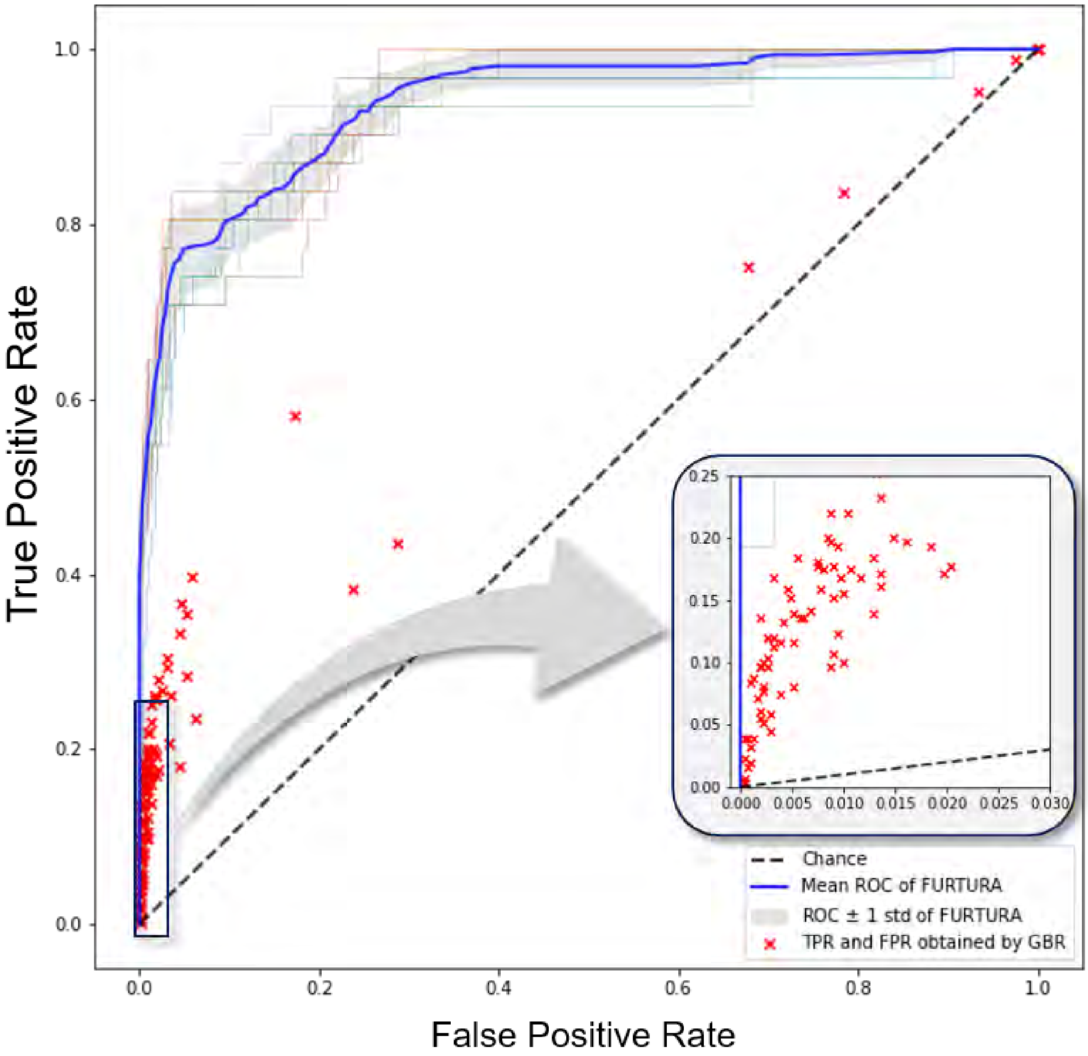


Figure 5.7: FUTURA outperforms Gradient Boosting Regressor (GBR) when the turbulence is to be predicted 30 sec prior it occurs. A $5 \times 2cv$ is conducted on the dataset and the result is obtained by averaging 10 experiments results on test sets. The blue line is the average of ROC curves of FUTURA and the grey area illustrates the ± 1 standard deviation. Each red cross mark represents a TPR/ FPR pair obtained by GBR under one hyperparameter configuration, and also obtained by averaging 10 experimental results. 100 different hyperparameter combinations are generated from Random Search.

As can be seen from figure 5.7, no matter what hyperparameter combinations GBR takes, the red cross is always below the blue line. Especially, the biggest mean TPR obtained by GBR with a zero FPR is 0. This shows that FUTURA outperforms GBR when the turbulence is to be predicted 30 sec prior it occurs.

5.4.3 Comparison of time complexity and memory requirement between FUTURA and the offline method

The embedded aircraft control computer has strict requirement on memory footprint and computational effort of algorithms that runs on it. In this section, we compare the time complexity as well as the memory requirement of FUTURA and the offline method.

As FUTURA and the offline method both apply anomaly detection algorithms for the final step of turbulence prediction, the main difference of two methods is that FUTURA uses SSKF to recursively approximate raw time series and the offline method uses least square for function approximation.

5.4.3.1 Time complexity

Applying a turbulence alerting function into embedded system, the computation capability constraint need to be taken into account. In addition, to predicting turbulence in a real time manner, the algorithm needs to mine useful information from high frequency time series data coming continuously from on board sensors. The high calculation frequency puts forward requirements on the complexity of the algorithm. Therefore, to make the practical application of data driven turbulence alerting function possible, it is desirable yet essential to have a low time complexity algorithm.

The time complexity of ordinary least square with a sliding window is $O(C_{window}^2 N_{window} + C_{window}^3)$ where C_{window} is the number of basis functions in the window and N_{window} is the number of data points in the sliding window.

The time complexity for Kalman Filter is $O(d^2r) + O(dr^2) + O(d^3) + O(r^3)$ per iteration where d is the degree of basis functions and r is the dimension of measurement y_t. The time complexity for SSKF is $O(d^2) + O(dr)$. Table 5.7 illustrates the time complexity comparison of KF and SSKF and Table 5.8 shows the notations and their corresponding matrix dimensions. As the converged state covariance matrix as well as Kalman gain are calculated in a offline manner, simplifying KF as SSKF reduces the time complexity of overall algorithm thus better adapts to real time application.

Table 5.7: Time complexity comparison of KF and SSKF

	KF	**SSKF**
1. $\hat{\boldsymbol{x}}_t^- \leftarrow \boldsymbol{F}_t \hat{\boldsymbol{x}}_{t-1}^+ + \boldsymbol{G}_t \boldsymbol{u}_t$	$O(d^2)$	$O(d^2)$
2. $\boldsymbol{P}_t^- \leftarrow \boldsymbol{F}_t \boldsymbol{P}_{t-1}^+ \boldsymbol{F}_t^T + \boldsymbol{Q}_t$	$O(d^3)$	-
3. $\boldsymbol{K}_t \leftarrow \boldsymbol{P}_t^- \boldsymbol{H}_t^T (\boldsymbol{H}_t \boldsymbol{P}_t^- \boldsymbol{H}_t^T + \boldsymbol{R}_t)^{-1}$	$O(dr^2) + O(d^2 r) + O(r^3)$	-
4. $\hat{\boldsymbol{x}}_t^+ \leftarrow \hat{\boldsymbol{x}}_t^- + \boldsymbol{K}_t(\boldsymbol{y}_t - \boldsymbol{H}_t \hat{\boldsymbol{x}}_t^-)$	$O(dr)$	$O(dr)$
5. $\boldsymbol{P}_t^+ \leftarrow (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t) \boldsymbol{P}_t^- (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t)^T + \boldsymbol{K}_t \boldsymbol{R}_t \boldsymbol{K}_t^T$	$O(dr^2) + O(d^2 r) + O(d^3)$	-
Total	$O(dr^2) + O(d^2 r) + O(d^3) + O(r^3)$	$O(d^2) + O(dr)$

Comparing the time complexity of the least square with SSKF for functional approximation, as each data point in the sliding window is approximated by $d + 1$ basis functions (see figure 4.1), the number of basis functions C_{window} is always greater than or equal to $d + 1$. Acting as a smoothing effect, the value of r is often less than or equal to 3. Therefore the time complexity of b-spline approximation by SSKF is undoubtedly much smaller than least square with a sliding window.

The time complexity of the offline method depends on the number of basis functions in the sliding window. The more basis functions used, the better the performance of the approximation, but the computation time grows cubically. Considering the strict computational constraints of aircraft embedded systems, such characteristic is not desirable for the real-time application of the algorithm. The time complexity of functional approximation using steady state Kalman Filter, however, depends only on the degree of basis functions. Using more basis functions (less sampling points between two b-spline knots thus smaller η) would not result in the increase in computation effort, which is an advantage for real time application.

5.4.3.2 Memory requirement

As mentioned in the chapter 4, using least square for functional approximation, all data points in sliding window N_{window} need to be stored and updated in real time application.

Table 5.8: Notations in Kalman filter and their dimensions.

	Notation	Dimension
$\hat{x}_{t-1}^{+}$	estimated system state vector at time step t-1	$d+1 \times 1$
$\hat{x}_{t}^{-}$	predicted system state vector at time t (a priori estimate before the measurement y_t is introduced into the system)	$d+1 \times 1$
$\hat{x}_{t}^{+}$	estimated system state vector at time step t (a posteriori estimate after the measurement y_t is introduced into the system).	$d+1 \times 1$
P_{t-1}^{+}	covariance matrix of the previous state at time t-1	$d+1 \times d+1$
P_{t}^{-}	a priori covariance matrix before the measurement y_t is processed	$d+1 \times d+1$
P_{t}^{+}	a posteriori covariance matrix for the current state after the measurement y_t is processed	$d+1 \times d+1$
u_t	deterministic input variable to the system	$d+1 \times 1$
y_t	measurement vector	$r \times 1$
F_t	state transition matrix	$d+1 \times d+1$
G_t	input transition matrix	$d+1 \times d+1$
Q_t	the process noise matrix	$d+1 \times d+1$
R_t	the measurement noise covariance matrix	$r \times r$

While SSKF approximate raw points in an incremental way and one only needs to store and update the state vector of dimension $d+1$ where $d+1 \ll N_{window}$.

5.4.3.3 Running time comparison

To compare the actual CPU processing time of approximating multivariate time series by SSKF and least square method, experiments are conducted on a 2 hours flight recordings of 22 sensors on an Intel(R) Core(TM) i5-8365U 1.9 GHz processor. SSKF is applied directly on the 2 hours flight data for functional approximation while least square is implemented on a sliding window of 10 seconds length and 1 second step size. Experimental result shows that the total processing time of SSKF is 17.4 seconds compared to 739.9 seconds by using least square with a sliding window. It should be noted that the experiments is conducted using ordinary least square method where the penalty term has not been added. The huge gap in execution time not only reflects the infeasibility of least square methods running on flight control computers but also

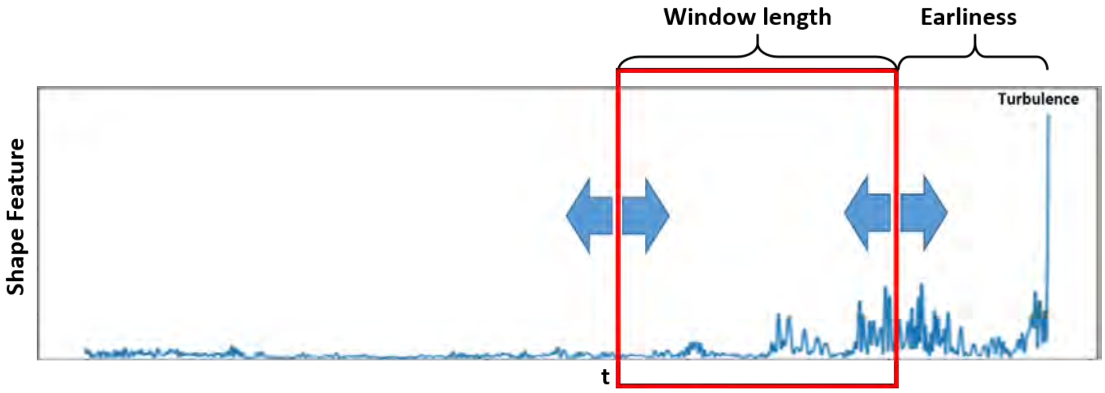


Figure 5.8: Illustration of earliness and sliding window length in turbulence prediction. This figure shows the evolution of a shape feature with time. Red box is the sliding window where the data inside are utilized for turbulence prediction.

highlights the advantages of FUTURA in the aspect of efficiency.

5.4.4 Hyperparameter study: Earliness and sliding window length

Earliness is an important hyperparameter as it is desirable to precisely predict turbulence as soon as possible. A common hypothesis is that the further away from turbulence the harder it is to predict its occurrence. To verify the impact of earliness on turbulence prediction performance, we conducted the following experiments. In addition to earliness, we also studied the influence of the length of sliding window (which is applied on the extracted shape feature to perform anomaly detection) on prediction performance.

A visualization of earliness and window length is illustrated in figure 5.8. A sliding window is applied on the shape feature Velocity for the following anomaly detection task.

5.4.4.1 Experiment

We iterate the algorithm for turbulence prediction setting the earliness and window length as 1,10,20,30,40,50,60,70,80,90,100,110,120 seconds respectively. Figure 5.8 illustrates the window length and earliness in the experiment. For each earliness and window length pairs, we repeat the prediction 10 times and calculate the mean True Positive Rate (TPR) corresponding to the zero False Positive Rate (FPR).

Figure 5.9 shows the result of experiment. The lighter the color, the higher the value of TPR. It can be deduced that:

- Looking at the heat map from the vertical direction, there is a negative correlation between TPR-Earliness: TPR normally decreases when earliness increases.
- Unlike earliness, there is no simple monotonic relationship between window length and TPR. But the best window lengths tend to increase as Earliness increases (the blue box in each line illustrates the maximum TPR in each earliness value), which may indicate that the further away from turbulence, the more information is needed to predict its occurrence.

A 3D visualization is showed in figure 5.10 a). Figure 5.10 b) and c) are side views of 5.10 a). Figure 5.10 b) shows the negatively correlated relation between TPR and earliness. In addition, TPR decreases non-linearly with the increase of earliness. Whatever the window length values, the TPR stays in a very low level when earliness is larger than 60 seconds. Precisely predicting turbulence more than 60 seconds in advance is tricky. Inversely, predicting turbulence a few seconds prior to it occurs will result in a big improvement in detection rate, except that it makes limited sense from an operation point of view. We therefore set the earliness to 30 seconds to ensure the predictive performance while leaving the crew and passengers have enough time. From figure 5.10 c) we can see that there is no monotonic relation between TPR and window length. There is no one window length that is the best for all earliness.

5.4.5 Predicting turbulence based on shape feature velocity: a comparison of anomaly detection and classification methods

In the last step of FUTURA, right after the shape feature velocity is extracted, we applied an anomaly detection method called 'Functional Isolation Forest (FIF)'(Staerman 2019) to identify which flights will encounter turbulence. As being introduced in the chapter 2, a number of research works are conducted in the literature for univariate and multivariate time series anomaly detection as well as classification tasks. As the multivariate sensor signals are transformed into the univariate shape feature 'velocity' by FUTURA, apart from FIF, various state-of-the-art time series classification and anomaly detection

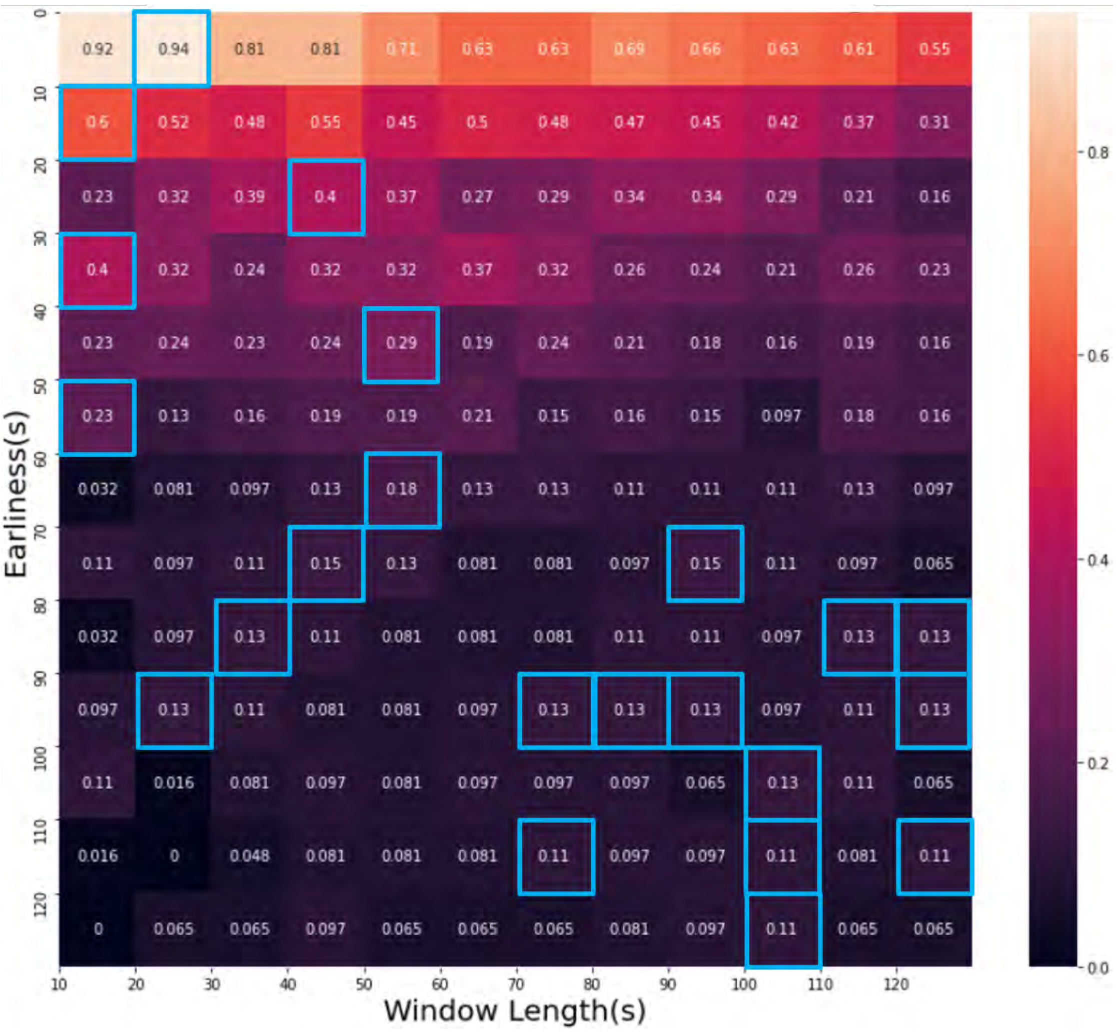


Figure 5.9: TPR (zero FPR) obtained by different earliness and sliding window length settings. x axis represents the window length and y axis are the earliness values. The number on each square represents the TPR obtained by zero FPR in a different window length and earliness configuration. The blue box in each line illustrates the maximum TPR in each earliness value.

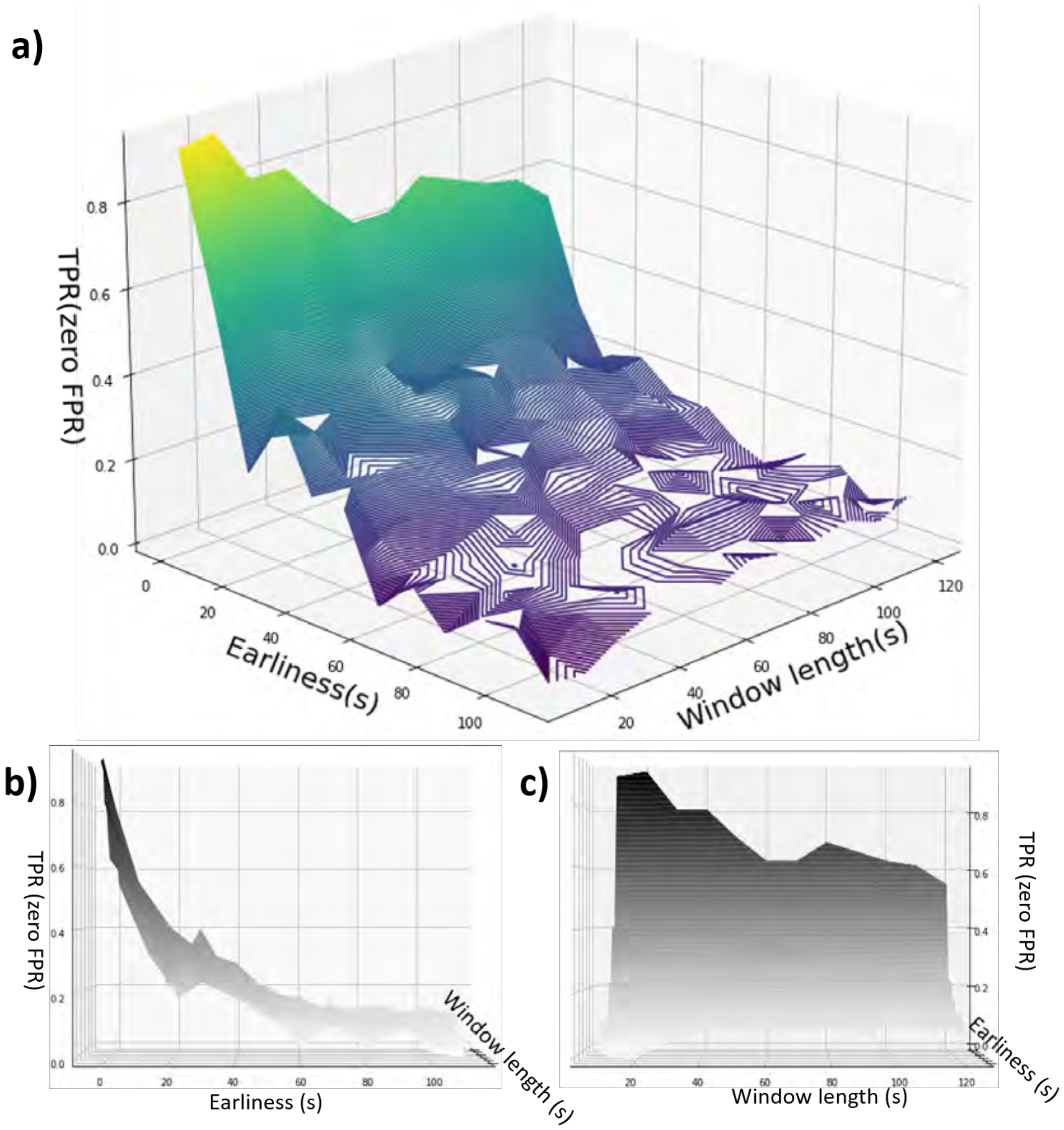


Figure 5.10: Influence of earliness and window length: A 3D visualization. a) TPR (zero FPR) obtained by different earliness and sliding window length settings. b) a side view of a) to illustrate the relation between TPR(zero TPR) and earliness. c) a side view of a) to illustrate the relation between TPR(zero TPR) and window length

.

methods can be applied on ’velocity’ for turbulence prediction. In this section, we make a comparison of the state-of-the-art time series classification as well as anomaly detection methods for their turbulence prediction performance based on the extracted shape feature ’velocity’. With the final purpose of applying the turbulence alerting algorithm on aircraft on-board computers, we wish to make our algorithm as simple (in terms of time complexity) as possible. To this end, the inference time (the running time to score new flights) of each method is also evaluated. Compared to the execution time of training which is conducted in an offline manner, the inference time reflects the delay and computation efforts for real time responses in the practical application thus it is a more valuable metric for an embedded function.

5.4.5.1 Experiment

Experiments are conducted using 5 state-of-the-art time series classification methods including Bag-of-SFA-Symbols (BOSS) (Schäfer 2015), Supervised Time Series Forest (STSF) (Cabello 2020), Time Series Forest (TSF) (Deng 2013), Shapelet Transform Classifier (Hills 2014) (Bostrom 2017), K-Nearest Neighbors with Dynamic Time Warping (KNN-DTW)(Xi 2006) and 3 anomaly detection methods including Functional Isolation Forest (FIF) (Staerman 2019), One Class Support Vector Machine (OCSVM) (Manevitz 2001), Local Outlier Factor (LOF) (Breunig 2000).

Using the same experimental settings as in section 5.4.1 above, we conduct a 2-fold cross-validation that is repeated 5 times with stratified sampling on the dataset. To measure the accuracy of the model, we plot the average Receiver Operating Characteristic (ROC) curves based on the results on the test sets and calculate the corresponding average Area Under the Curve (AUC). The mean True Positive Rate (TPR) with zero False Positive Rate (FPR) are also compared among all the methods. For LOF and OCSVM, the models are fitted only on inliers (normal flights) of training set and the anomaly score is given on each flights in the test set according to how isolated the object is from the training data (Breunig 2000) (Manevitz 2001). For KNN-DTW, the performance of the model is highly sensitive to the number of neighbors (k value). A grid search is therefore performed to determine the k value resulting in the highest AUC and the following ROC curve of KNN-DTW is the outcomes derived when k is equal to 23.

Figure 5.11 illustrates the ROC curves and their AUC values averaging from 10 times cross-validation on test set. The light colored areas illustrates the ± 1 standard deviation. FIF achieves the highest average AUC value of 0.94. STSF, TSF, LOF reach a AUC value of 0.92 and their ROC curves exhibit mutual intersection without one fully encompassing the others. KNN-DTW closely follows with a mean AUC value of 0.91. The mean AUC value of OCSVM is 0.82 and it exhibits a steep increase when FPR is close to zero, which will be further analyzed by in the following subsection. BOSS and Shapelet transform exhibit a suboptimal performance, attaining mean AUC values of only 0.63 and 0.59, respectively.

As discussed above, operational robustness is the key of on-board turbulence alerting functions. We therefore calculate and compare the mean TPR with zero FPR of all 8 methods. Although OCSVM only reaches an average AUC value of 0.88, it obtains the best mean TPR value of 0.46 with zero FPR. This can also be seen from its ROC curve: the growth is very steep until reaching a TPR value of 0.8 after which the growth is slowing down considerably. FIF reaches the second highest TPR of 0.39. The TPR for TSF, LOF and STSF are 0.28, 0.20 and 0.12, respectively. Finally, the average TPR of BOSS, Shapelet transform and KNN-DTW tend to zero, indicating that they cannot correctly identify upcoming turbulence without false alarms. Unlike AUC which measures the general classification performance regardless of threshold, TPR with zero FPR exhibits the robustness or conservative level: making positive classification only with strong evidence to eliminate false alarm. Performing well on one metric does not guarantee the same performance on another. The high AUC value of LOF does not guarantee its operational robustness and the situation with OCSVM is just the opposite.

Table 5.9: Comparison of average TPR(FPR=0) obtained by different anomaly detection and classification methods

FIF	OCSVM	TSF	STSF	BOSS	Shapelet	LOF	KNN-DTW
0.39	0.46	0.28	0.12	0	0.01	0.20	0

Comparing the average inference duration on test set, two classic novelty detection methods OCSVM and LOF perform the best, and the running time of both is less than 0.01 second. Three ensemble methods, TSF, STSF and FIF follow behind with running time of 0.407, 2.376 and 4.015, respectively. The average inference time of BOSS and Shapelet transform are 35.636 and 21.906 seconds, which demonstrates that they are not suitable for real time application.

Considering all three metrics, OCSVM has the highest TPR with zero FPR and the

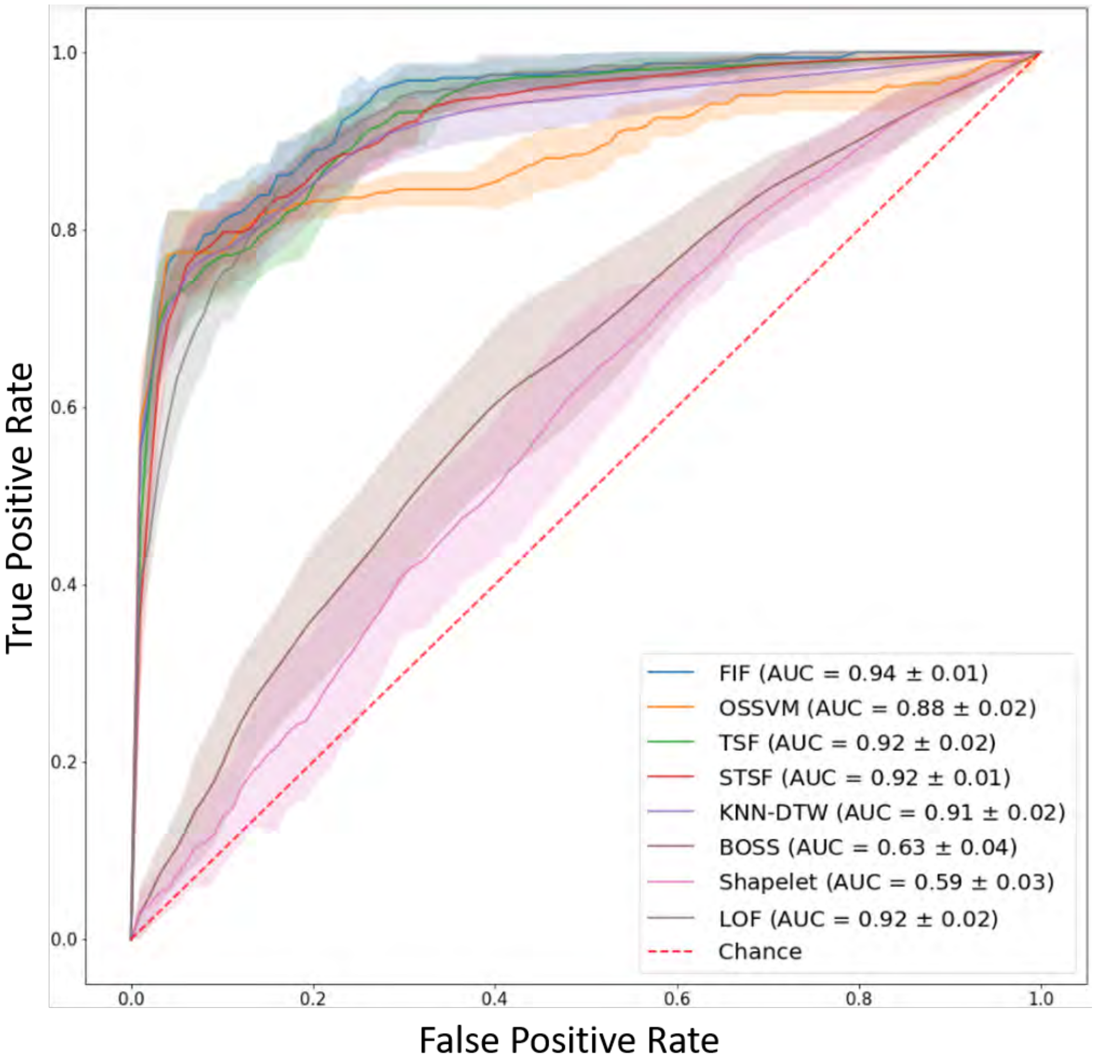


Figure 5.11: ROC curves and their AUC values of time series classification and anomaly detection methods for turbulence early identification. The colored lines are the ROC curves of different time series classification and anomaly detection methods and the light colored areas illustrates their corresponding ± 1 standard deviation. The average AUC values and standard deviation are shown in the lower right corner. The red dotted diagonal line reflects a classifier that correctly assign class labels to each class with a 50% probability, which performs no better than random chance.

shortest inference time but a relative low AUC. FIF has the highest AUC value and second highest TPR but a relative long running time. Both two methods make good candidate for turbulence early identification and can be used as a part of FUTURA algorithm.

Table 5.10: Comparison of average inference time (s) obtained by different anomaly detection and classification methods

FIF	OCSVM	TSF	STSF	BOSS	Shapelet	LOF	KNN-DTW
4.015	0.004	0.407	2.376	35.636	21.960	0.009	8.905

5.5 Conclusion

In this chapter, we conduct experiments on commercial aircraft flight recordings and comparison are made between our proposed approach with gradient boosting based methods. The experimental results show that the proposed method outperforms gradient boosting classifier and gradient boosting regressor in terms of turbulence early identification. In addition, predicting atmospheric turbulence 30 seconds prior to it occurs, the presented algorithm can detect 40% turbulence (9% for GBC and 0% for GBR) while keeping a zero false positive rate, which meets the zero false alarm requirement for optimizing the passenger experience and the aircraft operational reliability.

To compare the running time of approximating multivariate time series by SSKF integrated in FUTURA and least square method integrated in the offline method, experiments are conducted on a 2 hours flight recordings of 22 sensors. Experimental result shows that the total processing time of SSKF is 17.4 seconds compared to 739.9 seconds by using least square with a sliding window, which not only reflects the infeasibility of least square methods running on flight control computers but also highlights the advantages of FUTURA in the aspect of efficiency.

Experiments are also conducted to investigate the influence of earliness on turbulence identification performance and we found that TPR decreases non-linearly with the increase of earliness. Based on the functional shape feature extracted from multivariate time series data, we compared the turbulence prediction performance of 5 state-of-the-art time series classification methods and 3 anomaly detection methods.

OCSVM achieve the highest TPR with zero FPR and the shortest inference time. FIF has the highest AUC value and second highest TPR but a relative long running time.

Chapter 6

Conclusion

In this thesis, we design FUTURA (FUnctional shape feature for real time TURbulence Alerting), an on-board turbulence early identification model utilizing only the sensor data on-board of aircraft. FUTURA combines three technologies, namely Steady State Kalman Filter (SSKF), functional data analysis and time series anomaly detection for turbulence early identification. In the first step, we put forward SSKF for functional approximation. SSKF is utilized to approximate raw signals by estimating the coefficients of b-spline basis functions in a recursive manner. In the second step, functional shape features are extracted from the functional approximation obtained in the first step. Finally, a time series anomaly detection method is applied on the extracted shape feature to give an anomaly score to each flight. Turbulence is considered as an anomaly and by setting a threshold on anomaly scores, we can distinguish which flights will shortly encounter turbulence and which flights will not.

Comparing with the existing multivariate time series early classification methods where the relationship among variables is not considered, FUTURA captures the dynamic relations between variables which may contain important information that heralds the coming of the turbulence. Besides, the proposed method also makes real time turbulence prediction become possible due to its recursive nature and relative low computation complexity.

To investigate the performance and properties of FUTURA, a sequence of experiments is conducted. The method is evaluated using multivariate time series sensor data coming from flights that encountered different levels of air disturbance including severe turbulence and comparisons are made with gradient boosting based methods that are utilized in the recent literature for turbulence prediction task. Experimental results show that FUTURA can distinguish 40% of the severe turbulence cases (true positive rate) 30 seconds in advance while keeping a zero false positive rate, which not only outperforms the gradient boosting based methods, but also meets the zero false alarm requirement for optimizing the passenger experience and the aircraft operational reliability.

A limitation of our approach is that a sliding window is still needed to segment the extracted shape features to apply functional isolation forest for upcoming turbulence identification. Therefore, to have a fully recursive turbulence alerting method to reduce storage requirements, a further step will be replacing functional isolation forest by a recursive anomaly detection or classification approach. Another limitation is that the proposed method is not applicable on clear air turbulence using the current flight parameters. A possible improvement is to test FUTURA using extra flight parameters. Another line of research is to construct a specific predictive model for clear air turbulence prediction based on its physical mechanism.

As future work, we plan to test FUTURA on a new data set of the aircraft model A. The new data set contains the whole flight recordings, from take-off to touch down, of 51 flights with an average flight duration of 9 hours. In addition to the original 22 flights parameters, the new data set encompasses 3 additional flight parameters. We plan to first apply FUTURA on this new data set without the additional flight parameters to test its performance as well as its robustness in different flights phases. Compared to the cruise phase, maneuver involves more frequently in the stages like the climb and descent. Indeed, during the cruise phase, there is either little or no maneuver. Conducting experiments on the whole flight recordings allows us to investigate the impact of maneuver on turbulence predictive performance. Then, we will relaunch the experiments by adding the 3 additional flight parameters to study their influence.

In addition to the experiments on new data, another task to be tackled in the future is to experiment more functional shape features as well as their combinations that may reflect different aspects of relationships among variables. Moreover, it is also valuable to investigate in the theoretical demonstration for periodic convergence of Kalman gain.

Bibliographie

[Ahn 2020] Gilseung Ahn, Hwanchul Lee, Jisu Park et Sun Hur. *Development of indicator of data sufficiency for feature-based early time series classification with applications of bearing fault diagnosis.* Processes, vol. 8, no. 7, page 790, 2020.

[Alcalay 2018] Guillaume Alcalay. *Estimation de paramètres de vol avion et détection de pannes capteurs.* PhD thesis, Toulouse, ISAE, 2018.

[Anderson 2012] Brian DO Anderson et John B Moore. Optimal filtering. Courier Corporation, 2012.

[Bergstra 2012] James Bergstra et Yoshua Bengio. *Random search for hyper-parameter optimization.* Journal of machine learning research, vol. 13, no. 2, 2012.

[Bernard 2020] Ferrieres Bernard Lejeune et Teste. *curve shape analysis, https://github.com/Clej/curve_shape_analysis.git.* GitHub repository, 2020.

[Blázquez-García 2021] Ane Blázquez-García, Angel Conde, Usue Mori et Jose A Lozano. *A review on outlier/anomaly detection in time series data.* ACM Computing Surveys (CSUR), vol. 54, no. 3, pages 1–33, 2021.

[Bostrom 2017] Aaron Bostrom et Anthony Bagnall. *Binary shapelet transform for multiclass time series classification.* Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXII: Special Issue on Big Data Analytics and Knowledge Discovery, pages 24–46, 2017.

[Bradley 1997] Andrew P Bradley. *The use of the area under the ROC curve in the evaluation of machine learning algorithms.* Pattern recognition, vol. 30, no. 7, pages 1145–1159, 1997.

[Braei 2020] Mohammad Braei et Sebastian Wagner. *Anomaly detection in univariate time-series: A survey on the state-of-the-art.* Computing Research Repository (CoRR), 2020.

[Breunig 2000] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng et Jörg Sander. *LOF: identifying density-based local outliers.* In Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pages 93–104, 2000.

[Buda 2018] Mateusz Buda, Atsuto Maki et Maciej A Mazurowski. *A systematic study of the class imbalance problem in convolutional neural networks.* Neural networks, vol. 106, pages 249–259, 2018.

[Cabello 2020] Nestor Cabello, Elham Naghizade, Jianzhong Qi et Lars Kulik. *Fast and accurate time series classification through supervised interval search.* In 2020 IEEE International Conference on Data Mining (ICDM), pages 948–953. IEEE, 2020.

[Chawla 2002] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall et W Philip Kegelmeyer. *SMOTE: synthetic minority over-sampling technique.* Journal of artificial intelligence research, vol. 16, pages 321–357, 2002.

[Chawla 2010] Nitesh V Chawla. *Data mining for imbalanced datasets: An overview.* Data mining and knowledge discovery handbook, pages 875–886, 2010.

[Chen 2019] Qi Chen, Zhanhai Wang, Jian Wan, Tianmei Feng, Pu Chen, Chun Wang et Chen Zhang. *Design of a Turbulence Prevention System Based on ATG.* In 2019 IEEE 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT), pages 445–449. IEEE, 2019.

[Clancy 1975] Laurence Joseph Clancy. Aerodynamics. John Wiley & Sons, 1975.

[Cohen 2009] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang et Israel Cohen. *Pearson correlation coefficient.* Noise reduction in speech processing, pages 1–4, 2009.

[Deng 2013] Houtao Deng, George Runger, Eugene Tuv et Martyanov Vladimir. *A time series forest for classification and feature extraction.* Information Sciences, vol. 239, pages 142–153, 2013.

[Dietterich 1998] Thomas G Dietterich. *Approximate statistical tests for comparing supervised classification learning algorithms.* Neural computation, vol. 10, no. 7, pages 1895–1923, 1998.

[Emara 2021] Mariam Emara, Marcos dos Santos, Noah Chartier, Jamey Ackley, Tejas G Puranik, Alexia Payan, Michelle Kirby, Olivia J Pinon et Dimitri N Mavris. *Machine Learning Enabled Turbulence Prediction Using Flight Data for Safety*

Analysis. 32th Congress of the International Council of the Aeronautical Sciences, 2021.

[FAA 2014] FAA. Federal aviation administration: Aeronautical information manual. 2014.

[Faouzi 2022] Johann Faouzi. *Time series classification: A review of algorithms and implementations.* Machine Learning (Emerging Trends and Applications), 2022.

[Fauvel 2020] Kevin Fauvel, Daniel Balouek-Thomert, Diego Melgar, Pedro Silva, Anthony Simonet, Gabriel Antoniu, Alexandru Costan, Véronique Masson, Manish Parashar, Ivan Rodero*et al. A distributed multi-sensor machine learning approach to earthquake early warning.* In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 403–411, 2020.

[Frazier 2018] Peter I Frazier. *A tutorial on Bayesian optimization.* arXiv preprint arXiv:1807.02811, 2018.

[Ganganwar 2012] Vaishali Ganganwar. *An overview of classification algorithms for imbalanced datasets.* International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 4, pages 42–47, 2012.

[Ghalwash 2012] Mohamed F Ghalwash et Zoran Obradovic. *Early classification of multivariate temporal observations by extraction of interpretable shapelets.* BMC bioinformatics, vol. 13, pages 1–12, 2012.

[Ghalwash 2013] Mohamed F Ghalwash, Vladan Radosavljevic et Zoran Obradovic. *Extraction of interpretable multivariate patterns for early diagnostics.* In 2013 IEEE 13th International Conference on Data Mining, pages 201–210. IEEE, 2013.

[Gill 2014] Philip G Gill. *Objective verification of World Area Forecast Centre clear air turbulence forecasts.* Meteorological Applications, vol. 21, no. 1, pages 3–11, 2014.

[Goupil 2011] Philippe Goupil. *AIRBUS state of the art and practices on FDI and FTC in flight control system.* Control Engineering Practice, vol. 19, no. 6, pages 524–539, 2011.

[GPyOpt 2016] GPyOpt. *GPyOpt: A Bayesian Optimization framework in python.* http://github.com/SheffieldML/GPyOpt, 2016.

[Grabocka 2014] Josif Grabocka, Nicolas Schilling, Martin Wistuba et Lars Schmidt-Thieme. *Learning time-series shapelets.* In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 392–401, 2014.

[Gupta 2020a] Ashish Gupta, Hari Prabhat Gupta, Bhaskar Biswas et Tanima Dutta. *Approaches and applications of early classification of time series: A review.* IEEE Transactions on Artificial Intelligence, vol. 1, no. 1, pages 47–61, 2020.

[Gupta 2020b] Ashish Gupta, Hari Prabhat Gupta, Bhaskar Biswas et Tanima Dutta. *A divide-and-conquer–based early classification approach for multivariate time series with different sampling rate components in iot.* ACM Transactions on Internet of Things, vol. 1, no. 2, pages 1–21, 2020.

[Han 2005] Hui Han, Wen-Yuan Wang et Bing-Huan Mao. *Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning.* In Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1, pages 878–887. Springer, 2005.

[Hatami 2013] Nima Hatami et Camelia Chira. *Classifiers with a reject option for early time-series classification.* In 2013 IEEE symposium on computational intelligence and ensemble learning (CIEL), pages 9–16. IEEE, 2013.

[He 2013] Guoliang He, Yong Duan, Tieyun Qian et Xu Chen. *Early prediction on imbalanced multivariate time series.* In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pages 1889–1892, 2013.

[He 2015] Guoliang He, Yong Duan, Rong Peng, Xiaoyuan Jing, Tieyun Qian et Lingling Wang. *Early classification on multivariate time series.* Neurocomputing, vol. 149, pages 777–787, 2015.

[He 2020] Guoliang He, Wen Zhao et Xuewen Xia. *Confidence-based early classification of multivariate time series with multiple interpretable rules.* Pattern Analysis and Applications, vol. 23, pages 567–580, 2020.

[Hills 2014] Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp et Anthony Bagnall. *Classification of time series by shapelet transformation.* Data mining and knowledge discovery, vol. 28, pages 851–881, 2014.

[Hosmer Jr 2013] David W Hosmer Jr, Stanley Lemeshow et Rodney X Sturdivant. Applied logistic regression, volume 398. John Wiley & Sons, 2013.

[Huang 2019] Rongshun Huang, Huabo Sun, Chen Wu, Chun Wang et Binbin Lu. *Estimating eddy dissipation rate with QAR flight big data.* Applied Sciences, vol. 9, no. 23, page 5192, 2019.

[IATA 2018] IATA. *IATA Turbulence Workshop, London, September 2018. International Air Transport Association, Montreal, Canada.* www.iata.org, 2018.

[Jauch 2017] Jens Jauch, Felix Bleimund, Stephan Rhode et Frank Gauterin. *Recursive B-spline approximation using the Kalman filter.* Engineering Science and Technology, an International Journal, vol. 20, no. 1, pages 28–34, 2017.

[JAXA 2018] JAXA. R&D *of onboard safety avionics technology to prevent turbulence-induced aircraft accidents.* https://www.aero.jaxa.jp/eng/research/star/safeavio/, 2018.

[Kalman 1960] Rudolph Emil Kalman. *A new approach to linear filtering and prediction problems.* 1960.

[Kim 2018] Jung-Hoon Kim, Robert Sharman, Matt Strahan, Joshua W Scheck, Claire Bartholomew, Jacob CH Cheung, Piers Buchanan et Nigel Gait. *Improvements in nonconvective aviation turbulence prediction for the world area forecast system.* Bulletin of the American Meteorological Society, vol. 99, no. 11, pages 2295–2311, 2018.

[Lee 2022] Jeffrey Chi Wai Lee, Christy Yan Yu Leung, Mang Hin Kok et Pak Wai Chan. *A Comparison Study of EDR Estimates from the NLR and NCAR Algorithms.* Atmosphere, vol. 13, no. 1, page 132, 2022.

[Lejeune 2020] Clément Lejeune, Josiane Mothe, Adil Soubki et Olivier Teste. *Shape-based outlier detection in multivariate functional data.* Knowledge-Based Systems, vol. 198, page 105960, 2020.

[Lewis 1994] David D Lewis et Jason Catlett. *Heterogeneous uncertainty sampling for supervised learning.* In Machine learning proceedings 1994, pages 148–156. Elsevier, 1994.

[Lin 2012] Jessica Lin, Rohan Khade et Yuan Li. *Rotation-invariant similarity in time series using bag-of-patterns representation.* Journal of Intelligent Information Systems, vol. 39, pages 287–315, 2012.

[Lin 2015] Yu-Feng Lin, Hsuan-Hsu Chen, Vincent S Tseng et Jian Pei. *Reliable early classification on multivariate time series with numerical and categorical attributes.* In Advances in Knowledge Discovery and Data Mining: 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part I 19, pages 199–211. Springer, 2015.

[Lines 2012] Jason Lines, Luke M Davis, Jon Hills et Anthony Bagnall. *A shapelet transform for time series classification.* In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 289–297, 2012.

[Liu 2008] Fei Tony Liu, Kai Ming Ting et Zhi-Hua Zhou. *Isolation forest.* In 2008 eighth ieee international conference on data mining, pages 413–422. IEEE, 2008.

[Lyche 2008] Tom Lyche et Knut Morken. *Spline methods draft.* Department of Informatics, Center of Mathematics for Applications, University of Oslo, Oslo, pages 3–8, 2008.

[Ma 2017] ChaoHong Ma, XiaoQing Weng et ZhongNan Shan. *Early classification of multivariate time series based on piecewise aggregate approximation.* In Health Information Science: 6th International Conference, HIS 2017, Moscow, Russia, October 7-9, 2017, Proceedings 6, pages 81–88. Springer, 2017.

[MacQueen 1967] J MacQueen. *Classification and analysis of multivariate observations.* In 5th Berkeley Symp. Math. Statist. Probability, pages 281–297. University of California Los Angeles LA USA, 1967.

[Manevitz 2001] Larry M Manevitz et Malik Yousef. *One-class SVMs for document classification.* Journal of machine Learning research, vol. 2, no. Dec, pages 139–154, 2001.

[Meymaris 2019] Gregory Meymaris, Robert Sharman, Larry Cornman et Wiebke Deierling. *The NCAR In Situ Turbulence Detection Algorithm.* 2019.

[Mori 2019] Usue Mori, Alexander Mendiburu, Isabel Marta Miranda et José Antonio Lozano. *Early classification of time series using multi-objective optimization techniques.* Information Sciences, vol. 492, pages 204–218, 2019.

[NASA 2021] NASA. NASA *Microphone Detects Turbulence Hundreds of Miles Away.* https://www.nasa.gov/directorates/spacetech/spinoff/NASA_Microphone_Detects_Turbulence_Hundreds_of_Miles_Away, 2021. Accessed: 2021-03-16.

[Pedregosa 2011] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot et E. Duchesnay. *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, vol. 12, pages 2825–2830, 2011.

[Press 1992] William H Press, Saul A Teukolsky, William T Vetterling et Brian P Flannery. *Numerical Recipes in C. second.* New York, Press Syndicate of the University of Cambridge, 1992.

[Ramasamy 2016] Subramanian Ramasamy, Roberto Sabatini, Alessandro Gardi et Jing Liu. *LIDAR obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid.* Aerospace Science and Technology, vol. 55, pages 344–358, 2016.

[Ramsay 2008] James O Ramsay et Bernhard W Silverman. Functional data analysis. Springer New York, 2008.

[Rousseeuw 1987] Peter J Rousseeuw. *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis.* Journal of computational and applied mathematics, vol. 20, pages 53–65, 1987.

[SAE 2010] SAE. Guidelines for development of civil aircraft and systems. SAE ARP 4754 Rev. A., 2010.

[Sakoe 1978] Hiroaki Sakoe et Seibi Chiba. *Dynamic programming algorithm optimization for spoken word recognition.* IEEE transactions on acoustics, speech, and signal processing, vol. 26, no. 1, pages 43–49, 1978.

[Schäfer 2015] Patrick Schäfer. *The BOSS is concerned with time series classification in the presence of noise.* Data Mining and Knowledge Discovery, vol. 29, pages 1505–1530, 2015.

[Schmitt 2007] Nikolaus Peter Schmitt, Wolfgang Rehm, Thomas Pistner, Paul Zeller, Hermann Diehl et Peter Navé. *The AWIATOR airborne LIDAR turbulence sensor.* Aerospace Science and Technology, vol. 11, no. 7-8, pages 546–552, 2007.

[Schölkopf 2001] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola et Robert C Williamson. *Estimating the support of a high-dimensional distribution.* Neural computation, vol. 13, no. 7, pages 1443–1471, 2001.

[Sharman 2016] Robert Sharman et Todd Lane. *Aviation Turbulence.* Springer International Publishing, Switzerland, vol. 10, pages 978–3, 2016.

[Simon 2006] Dan Simon. Optimal state estimation: Kalman, h infinity, and nonlinear approaches. John Wiley & Sons, 2006.

[Srivastava 2016] Anuj Srivastava et Eric P Klassen. Functional and shape data analysis, volume 1. Springer, 2016.

[Staerman 2019] Guillaume Staerman, Pavlo Mozharovskyi, Stephan Clémençon et Florence d'Alché Buc. *Functional isolation forest.* In Asian Conference on Machine Learning, pages 332–347. PMLR, 2019.

[Strandberg 2019] Joakim Strandberg, Thomas Hobiger et Rüdiger Haas. *Real-time sea-level monitoring using Kalman filtering of GNSS-R data.* GPS solutions, vol. 23, no. 3, pages 1–12, 2019.

[Su 2019] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun et Dan Pei. *Robust anomaly detection for multivariate time series through stochastic recurrent neural network.* In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pages 2828–2837, 2019.

[Thabtah 2020] Fadi Thabtah, Suhel Hammoud, Firuz Kamalov et Amanda Gonsalves. *Data imbalance in classification: Experimental evaluation.* Information Sciences, vol. 513, pages 429–441, 2020.

[Traverse 2004] Pascal Traverse, Isabelle Lacaze et Jean Souyris. *Airbus fly-by-wire: A total approach to dependability.* In Building the Information Society: IFIP 18th World Computer Congress Topical Sessions 22–27 August 2004 Toulouse, France, pages 191–212. Springer, 2004.

[Trenkle 1973] Fritz Trenkle et Manfred Reinhardt. In-flight temperature measurements, volume 2. North Atlantic Treaty Organization, Advisory Group for Aerospace Research, 1973.

[Wang 2018] Xing Wang, Jessica Lin, Nital Patel et Martin Braun. *Exact variable-length anomaly detection algorithm for univariate and multivariate time series.* Data Mining and Knowledge Discovery, vol. 32, pages 1806–1844, 2018.

[Welch 1995] Greg Welch, Gary Bishop*et al. An introduction to the Kalman filter.* 1995.

[Xi 2006] Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei et Chotirat Ann Ratanamahatana. *Fast time series classification using numerosity reduction.* In Proceedings of the 23rd international conference on Machine learning, pages 1033–1040, 2006.

[Xing 2009] Zhengzheng Xing, Jian Pei et S Yu Philip. *Early Prediction on Time Series: A Nearest Neighbor Approach.* In International Joint Conference on Artificial Intelligence (IJCAI), pages 1297–1302, 2009.

[Xing 2011] Zhengzheng Xing, Jian Pei, Philip S Yu et Ke Wang. *Extracting interpretable features for early classification on time series.* In Proceedings of the 2011 SIAM international conference on data mining, pages 247–258. SIAM, 2011.

[Yanovsky 2005] FJ Yanovsky. *Evolution and prospects of airborne weather radar functionality and technology.* In 2005 18th International Conference on Applied Electromagnetics and Communications, pages 1–4. IEEE, 2005.

[Ye 2011] Lexiang Ye et Eamonn Keogh. *Time series shapelets: a novel technique that allows accurate, interpretable and fast classification.* Data mining and knowledge discovery, vol. 22, pages 149–182, 2011.

[Zenati 2018] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat et Vijay Chandrasekhar. *Adversarially learned anomaly detection.* In 2018 IEEE International conference on data mining (ICDM), pages 727–736. IEEE, 2018.