Detecting Flight Trajectory Anomalies and Predicting Diversions in Freight Transportation

Claudio Di Ciccio[a,*], Han van der Aa[b], Cristina Cabanillas[a], Jan Mendling[a], Johannes Prescher[a]

[a]*Institute for Information Business, Vienna University of Economics and Business*
Welthandelsplatz 1, 1020 Vienna, Austria
[b]*Department of Computer Sciences, Vrije Universiteit Amsterdam, Faculty of Sciences*
De Boelelaan 1081, 1081HV Amsterdam, The Netherlands

Abstract

Timely identifying flight diversions is a crucial aspect of efficient multi-modal transportation. When an airplane diverts, logistics providers must promptly adapt their transportation plans in order to ensure proper delivery despite such an unexpected event. In practice, the different parties in a logistics chain do not exchange real-time information related to flights. This calls for a means to detect diversions that just requires publicly available data, thus being independent of the communication between different parties. The dependence on public data results in a challenge to detect anomalous behavior without knowing the planned flight trajectory. Our work addresses this challenge by introducing a prediction model that just requires information on an airplane's position, velocity, and intended destination. This information is used to distinguish between regular and anomalous behavior. When an airplane displays anomalous behavior for an extended period of time, the model predicts a diversion. A quantitative evaluation shows that this approach is able to detect diverting airplanes with excellent precision and recall even without knowing planned trajectories as required by related research. By utilizing the proposed prediction model, logistics companies gain a significant amount of response time for these cases.

Keywords: Air transportation, Airplane trajectory, Aircraft navigation, Logistics, Machine learning, Prediction methods

---

*Corresponding author. *Phone number:* +43 1 31336 5222

Email addresses: `claudio.di.ciccio@wu.ac.at` (Claudio Di Ciccio), `j.h.vander.aa@vu.nl` (Han van der Aa), `cristina.cabanillas@wu.ac.at` (Cristina Cabanillas), `jan.mendling@wu.ac.at` (Jan Mendling), `johannes.prescher@wu.ac.at` (Johannes Prescher)

1. Introduction

The growth of inter-continental trade has led to a notable increase in multi-modal transport. Multi-modal transport involves at least two modes of transportation on two consecutive transportation legs, which have to be synchronized. This is, for instance, the case when air freight cargo is unloaded at airports in order to be distributed into the hinterland by trucks, or sea ship cargo being redistributed at sea ports. Because multi-modal transport faces increasing challenges in terms of efficiency, describing and planning such sequential dependencies is a common concern [1]. The desire for efficiency is on the one hand driven by lean or just-in-time production systems, which require timely delivery. On the other hand, efficiency is demanded from an environmental perspective in order to avoid unnecessary CO_2 emissions. A crucial problem in this context is that different parties involved in a transportation chain hardly exchange real-time information related to individual deliveries [2]. This makes it difficult for a receiving party to respond in a timely way to unexpected events that occur earlier on in the transportation.

The impact of such unexpected events is especially prominent in supply chains that involve cargo airplanes. In case an airplane has to land in an airport that is not the intended destination (i.e. the flight is diverted), re-planning and adaptation mechanisms must be triggered so that other parties involved in the chain can continue with the delivery of the cargo. For instance, it may be required to cancel the transport orders for the planned airport and to provide transport capacities at the diverted airport. The resultant impact on a company's ability to deliver goods in time, the utilization of trucks, and the additional costs it incurs, can be mitigated by timely responding to diversions. To enable parties to do so, diversions need to be detected as early as possible. Although there exist approaches that achieve this (e.g. [3, 4]), these typically depend on information about the planned flight trajectory in order to detect a diversion. However, such information is often not readily available in practice [2], especially for logistics service providers. Therefore, it is desirable to identify anomalous flight behavior without depending on such information completeness.

In this paper, we address the problem of alerting receiving parties, e.g. trucking companies, in case a delivering airplane is diverted. Based on real scenarios and hence, keeping the decision making problem as realistic as possible as suggested in [5] from the standpoint of logistics service

providers, we make use of event data that is semi-publicly available. More specifically, our contribution is a prediction model that detects flight trajectory anomalies based on minimal input data. We implemented the model as a prototype and tested it on a sample of flights yielding a high predictive accuracy. The prediction model provides considerable gains in response time. It is therefore suited to be integrated in decision systems that support operations of logistics service providers, in contrast to traditional model-based decision support [6].

The remainder of the paper is structured as follows. Section 2 discusses the background of our research by describing a real-world scenario and by considering related work. Section 3 defines our prediction model, which builds on feature extraction and the classification of regular and anomalous airplane behavior. Section 4 presents the evaluation of this framework with a focus on effectiveness and response time gain. Finally, Section 5 presents the conclusions of this work.

2. Background

Our research is motivated by the need to monitor air transportation in scenarios where only a limited amount of flight information is available and by the lack of support for detecting en-route diversions under these circumstances. In this section, we first describe a freight transportation scenario from the EU-FP7 GET Service project[1] to motivate the problem we tackle. Subsequently, we identify the research gap by considering existing work related to the problem.

2.1. Scenario

This section describes a real-world transport scenario that demonstrates the importance of prompt and accurate prediction of diverting airplanes. In this multi-modal transport scenario, goods are transported from John F. Kennedy International airport (New York, USA) to a plant located in Utrecht (The Netherlands) [2]. It consists of two *transportation legs*. The first leg comprises air transportation from New York to Amsterdam Airport Schiphol. At the airport, the goods are transferred to trucks that have been sent by a *logistics service provider* to pick up the cargo. In the second leg, the trucks transport the goods to the destination plant in Utrecht.

---

[1] http://getservice-project.eu/

In this scenario, the goal of the logistics service provider is to deliver the goods on time to Utrecht, i.e. in conformance with *service level objectives* that have been agreed upon with the client [7]. A crucial factor impacting a logistics provider's ability to meet its service level objectives is whether or not an airplane arrives on time at its connection point. However, it is possible that this does not happen, because the airplane has to divert and land at a different airport. Such diversions can occur, for instance, due to adverse weather conditions or technical difficulties. In order to be able to still meet its service level objectives, a logistics service provider must respond in a timely and efficient manner to such events.

Although diversions are relatively rare, their impact on the logistic chain is significant. To recognize the impact of a diversion on costs and CO_2 emissions, it must be considered that the freight of an airplane is, on average, loaded onto 30 trucks.[2] If the airplane diverts to a different destination airport, the logistics service provider has to cancel (or reroute) the trucks that have been sent to the Schiphol airport, and in parallel arrange for new transportation means to pick up the cargo in Eindhoven. Therefore, this requires a rerouting of up to 60 trucks for a single airplane. Optimization of scheduling around such unexpected events is therefore recognized as one of the most important fleet management decisions [8]. In order for these corrective actions to be effective, it is crucial that the logistics service provider becomes aware of the airplane diversion as soon as possible [9]. Unfortunately, the communication between logistics service providers and cargo airlines is in practice not as prompt as required [2]. In fact, logistics service providers do not receive real-time information and are generally even only notified once an airplane has already landed at another airport. These delayed diversion notifications threaten the ability of logistics service providers to meet their service objectives.

In order to reduce the impact of diversions in practical settings, we thus face the challenge to automatically detect flight diversions by only making use of data from public data services. As such, logistics service providers can respond to diversion in a timely manner, independent of the quality of communication with other parties involved in the logistics chain.

---

[2]According to a major logistics service provider that we have collaborated with in this research project.

2.2. Related Work

In order to address the aforementioned challenges we are especially interested in prior research that relates to flight monitoring for anomalous trajectory detection. To the best of our knowledge, our research work is the first that addresses the issue of predicting the diversions of flights. We also remark here that our approach operates under the requirements that trajectories are not known a priori, and that there is no limited geographical area that is specifically meant to be put under analysis. Previous techniques have challenged related issues in the area of monitoring aircraft routes based on flight data. The approaches in this context differ in the goal they pursue, which is different to ours. Their operating conditions also change in terms of information they require about planned flight trajectories, the circumscription of the geographical area in scope, and the number of factors used to detect anomalous behavior. Nevertheless, they provide useful insight in the general scope of the automated detection of anomalies in flight transportation data.

The techniques that consider the expected flight routes typically represent such information by means of *waypoints*, namely sets of coordinates through which airplanes pass as intermediate junction points of a segmented trajectory. For example, Krozel [3] describes a set of methods to detect and measure to which extent the actual route differs from the filed flight plan assuming that information about the due waypoints is available, and reasons about the causes of such discrepancies. Periodic updates of the position of the aircraft including its location, altitude, and speed are analysed to such extent. In our approach, we aim at predicting diversions, rather than signalling aircrafts that are not within normal navigation performance error limits. Nevertheless, we also make use of location, altitude, and speed values as representative quantities to be monitored. Reynolds and Hansman [4] introduce a model-based framework for flight conformance monitoring based on Fault Detection and Isolation techniques. To do so, the flight data are compared with a model consisting of aircraft positions, velocity, and acceleration, supplemented with future conditions to be compliant with, in terms of planned trajectory, destination and target states towards the next waypoint. They validate their model-based conformance checking with simulated flight tracks. Although we do not assume to have any predefined flight model at hand, we also propose a technique that does not analyse the registered values singularly, but rather considers their

difference-based evolution over the time.

The approaches that do not rely on a planned route typically stick to a limited geographic area. Gariel et al. [10] present a tool called AirTrajectoryMiner aimed at monitoring the health of the airspace. A *healthy airspace* is defined as the condition in which all airplanes fly according to the plan. The tool clusters registered flight trajectories as two-dimensional projections within a geographical area around an airport, so as to distinguish those flights that traverse the area following the procedures from those that may disturb the landing operations of other flights. The technique is unsupervised, because it does not require training flight data to be pre-labeled. Guo et al. [11, 12] extend the work of Gariel et al. by including the speed of the airplane in the representation of the routes, which demonstrates to be more effective for the detection of abnormal (i.e., infrequently observed) trajectories in a given airspace. Although our approach seeks flights that are going to land in unexpected airports rather than considering their compliance with standard routes in the geographical area around a specific airport, we also propose a technique that does not require segments of the registered routes to be pre-labeled as adhering to normal behavior or not. We also include the bi-dimensional representation of positions and speed in the features that we monitor for detecting anomalous flights. Smart et al. [13] aim at detecting anomalies in the descent phase of the flights landing in a specific airport. They introduce a two-phase approach based on the use of one-class Support Vector Machine (SVM) [14] classifiers, which have proven to work better than other methods such as K-means clustering for that purpose [15]. In particular, a number of different SVMs is trained, each on an altitude range at which the aircraft is flying in proximity of the airport before landing. For each altitude level (hence, for each SVM), different variables are passed as an input to the related SVM, based on the parameters that the crew would be focusing on to fly the aircraft safely. They thus include aircraft-specific metrics such as the degrees of the flap setting. In common with this technique, we share the adoption of the one-class SVM in the implementation of the automated classification of flight intervals in our approach.

In the domain of air transportation, a way to mitigate the risk of flight diversions is to design proper flight trajectories, since the probability of flight diversion may also be influenced by the route taken by the airplane. For instance, some routes may go through areas which are likely to be

affected by thunderstorms. Consequently, great effort has been made to develop approaches for the computation of appropriate flight trajectories [16, 17, 18]. These approaches consider a number of different factors. For instance, Besada et al. [16] present several formal languages for the definition of static information, including the flight plan, which involves the initial and final waypoints as well as the so-called flight intent, i.e., the expected behavior, and potential optimization criteria. Other approaches also incorporate weather information in the generation of flight trajectories, sometimes focused only on the terminal airspace, and sometimes extended to en-route flight areas [19, 20, 21].

Flight trajectories are also monitored for purposes other than flight diversion detection. For instance, there are specific-purpose approaches focused on collision avoidance [22, 23, 24, 25]. Collision avoidance is a form of conflict resolution that requires sophisticated decision making support [26]. Sislak et al. [27] present an approach to detect conflicts between the trajectories of two flights and two different conflict-resolution algorithms based on high-level flight-plan variations using evasion maneuvers. Landry et al. [23] present a methodology to model and analyze airport surface constraints in order to avoid collisions. Other trajectory monitoring research focuses on the detection of specific types of deviations. For instance, Timar et al. [25] assess the occurrence conditions and operational impact of S-turns. These patterns often occur in congested airspaces (i.e., cluttered environments) when an airplane has to employ maneuvers to delay its arrival at the destination airport. Applying it to our goal, detecting such movements could be an indication of landing intents in an airport different than the expected one.

In general, flight-track monitoring and anomalous flight behavior detection build on techniques that take position data as input to detect behavioral anomalies, like has been done in various other contexts [28]. Approaches coming from different areas than flight transport exhibit characteristics that are helpful to the purpose of automated flight diversion detection. Examples include techniques for video surveillance [29, 30, 31, 32], transportation monitoring [33, 34], travel time analysis [35], and road traffic anomaly trajectory detection [36, 37, 38, 39, 40, 41]. These approaches often use GPS [34, 35] and Automatic Identification System (AIS) data [33] for monitoring purposes, and SVMs [31] for the detection of anomalies. They usually target a limited geographic area, which is split up into grid cells, such that a trajectory can be expressed as an ordered sequence

7

of traversed cells. The dimensionality and complexity of the classification problem are thereby reduced. Still, they depend on fine-granular grids of the local area of interest. In our setting, this would essentially require the definition of grids between waypoints. This is highly unpractical, given that there are over 100,000 relevant flight routes. In summary, prior approaches show features that are interesting for our goal, namely: *(i)* The reliability on GPS and AIS data for monitoring purposes; *(ii)* the variety of factors used for computing trajectories and checking conformance, including latitude, longitude, altitude and speed in the case of air traffic monitoring; and *(iii)* the use of multi-dimensional decision models or classifiers for anomalous behaviour detection, such as SVMs, against other classification methods.

3. Prediction Model

This section describes the proposed prediction model for the automated detection of diverting airplanes. During a flight, an airplane transmits updates on its position, velocity, and altitude. We refer to these updates as *flight track events*. Whenever our model receives a flight track events, it predicts whether the airplane is diverting or whether it is still heading towards it intended destination. To make this prediction, the model performs three subsequent steps, as illustrated in Figure 1. Given the receipt of a flight track event, the prediction model first combines the received information with the information from the previously received event. By combining the information from these two events, our model extracts a set of features that characterizes the airplane's behavior during the time interval in-between the two events. The second step uses a one-class classifier to determine whether the behavior in that time interval should be considered as normal or anomalous. In the third and final step, we augment the classification of the behavior with the classifications of the airplane's prior intervals. If the level of anomalous behavior in the flight history surpasses a certain threshold, our model predicts a diversion. In the next sections, we describe the model's input and its individual steps in detail.

3.1. Flight Tracking

During a flight, an airplane transmits information that can be collected by a range of receiving devices. We refer to the gathering of positional data of an airplane throughout its flight as *flight*
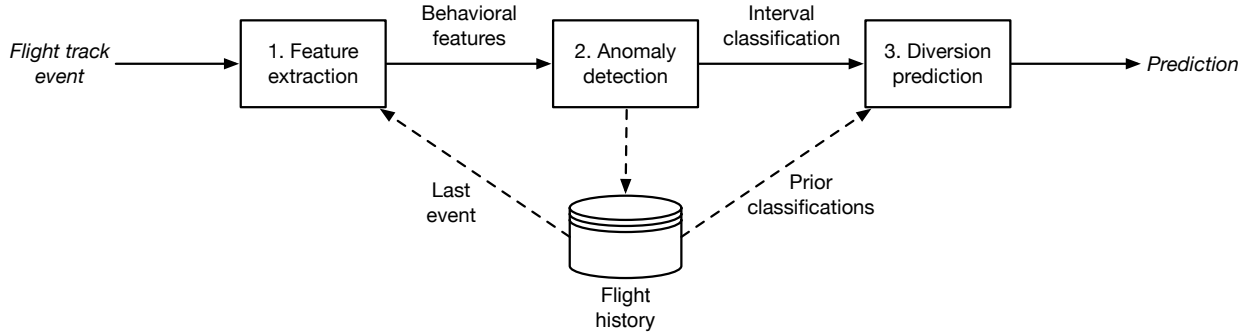
Figure 1: Overview of the prediction model for diversion detection

tracking. The information acquired by the flight tracking consists of a series of *events*, i.e., instantaneous information related to a specific point in time. Events describe the characteristics of the reported fact by means of the so-called *event attributes*. In the case of flight tracking, the event attributes includes flight number, speed, altitude, and geographical position.

Flight events are made (semi)-publicly available through several dedicated data providers such as Flightstats[3] and FlightRadar24[4]. Independent of the source from which it is obtained, the flight event e describing the status of an airplane at time τ as described by the following triple of attributes:

$$e = \langle p, h, v \rangle, \text{ with } p = \langle lat, lon \rangle$$

$p = \langle lat, lon \rangle$ represents a point in the geosphere identified by latitude *lat* and longitude *lon*, which, together with height h, describes the position of the airplane. Velocity v indicates the speed at which the airplane is flying. Flight events are received as sequences, with irregular time periods between subsequent updates. To capture the temporal dimension, all events are labeled with the time at which they are reported, namely τ. Therefore, when referring to specific occurred events, we will consider them as functions of τ, $e(\tau)$. By extension, we will consider their attributes as functions of time, too, i.e. $p(\tau)$, $h(\tau)$, $v(\tau)$. These sequences of flight events are the input that the prediction model will utilize.

---

[3]http://www.flightstats.com/
[4]http://www.flightradar24.com/

3.2. Feature Extraction

In the first step of the prediction model, we convert information from received flight track events into features that characterize the behavior of an airplane during a time interval. It is crucial to the applicability of our prediction model that these features can be computed based on a limited amount of information. We therefore derive the features based solely on information in the flight track event, i.e. time, position, altitude, and velocity, and on the locations of the origin and destination airports. The flight track events obtained from providers of public flight data, together with the geographic locations of the origin and destination airports, are used to derive features that characterize an airplane's behavior during a certain time interval. By extracting these features, we abstract from information that relate to a single point in time. Owing to this, features do not depend on the route that the airplane is following. This enables the prediction model to work for any flight trajectory, even if a flight between those two airports has never been observed before.

To this extent, we follow the general framework described in [42]. According to it, event attributes are hierarchically divided into *(i) constrained*, *(ii) monitorable*, and *(iii) free* attributes, where the first class is a sub-class of the second one, and in turn the second class is a sub-class of the third one. In our context, the *constrained* class describes the geographical position $p(\tau)$, given by a latitude-longitude pair. This attribute is constrained, because the obligatory expected initial and final values are set for it: respectively, the locations of the origin and destination airports. The second class, *monitorable* attributes, is represented in our case by flight speed $v(\tau)$ and altitude $h(\tau)$. These are monitored during the flight tracking, yet no constraint is assumed that define their initial and final value in our model. Such restrictions are indeed known by pilots and airline companies, but kept reserved and hence not known by external logistics service providers. Therefore we consider constraints on such attributes out of scope for our abstraction of data. Constrained and monitorable attributes are an input source for the discriminative classifier that determines whether the flight is showing an anomalous behavior. In our context, the class of *free* attributes refers to attributes such as the flight identifier. This attribute is used to pre-filter the stream of events in order to distinguish to which flight the current event belongs. Although flight codes respect normative assignments and flight identifiers can coincide with them, such knowledge goes

beyond the scope of our abstraction.

Given these attribute classes, the approach of [42] defines three main classes of features, namely *(i) progress from start, (ii) progress to end*, and *(iii) interval progress*. Each feature represents the dynamic evolution of a single event attribute along a time interval $I_\tau > 0$, defined by time instants τ and τ' ($\tau' = \tau + I_\tau$).

Progress from start and progress to end describe how further (resp., closer) an attribute's value gets to its initial (resp., final) expected value, by comparing the distance at the end of the time interval w.r.t. the beginning of the time interval. [5] Such variation is scaled by the difference between the initial value and the final one. Progress from start and progress to end can thus be derived only from constrained attributes, because their final and initial expected values are undefined for the other event-attribute classes. Interval progress is defined for monitorable attributes, as it considers the variation of the attribute value at time τ' and τ, scaled by the average of gathered values along the time interval. Hence, it considers the trend of the attribute under analyisis, locally to the given time interval, thus disregarding any initial or final value. In the context of our research, we have adapted the concepts expressed so far, defining the following features.

Distance completed (progress from start): an approximation of the fraction of the total distance from the origin airport that has been completed during interval I_τ. Equation 1 is used to compute the completed distance $\delta_d^{cmpl}(I_\tau)$, where $\Delta_\sigma(p', p'')$ represents the *great circle distance* Δ_σ between two positions p' and p''. p_O and p_D respectively denote the coordinates of the origin and destination airports.

$$\delta_d^{cmpl}(I_\tau) = \frac{\Delta_\sigma(p(\tau'), p_O) - \Delta_\sigma(p(\tau), p_O)}{\Delta_\sigma(p_O, p_D)} \tag{1}$$

Distance gained (progress to end): the fraction of the total distance gained towards the desti-

---

[5]The assumption made in [42] is that such features are defined for attributes whose domains can be mapped to, or constitute, metric spaces. As it can be inferred, this assumption applies for all the monitorable and constrained attributes in our context, as we deal with speed and altitude (numeric values for which the distance metric can be the arithmetic difference, respecting the triangle inequality) and geographical positions (for which the geodesic distance is defined).

nation airport in an interval I_τ, formulated as $\delta_d^{gain}(I_\tau)$ in Equation (2).

$$\delta_d^{gain}(I_\tau) = \frac{\Delta_\sigma(p(\tau), p_D) - \Delta_\sigma(p(\tau'), p_D)}{\Delta_\sigma(p_O, p_D)} \tag{2}$$

Velocity deviation (interval progress): average speed (v) deviation over an interval I_τ, formulated as $\delta_v(I_\tau)$ in Equation (3).

$$\delta_v(I_\tau) = 2 \times \frac{v(\tau') - v(\tau)}{v(\tau) + v(\tau')} \tag{3}$$

Altitude deviation (interval progress): average altitude (h) deviation over an interval I_τ, formulated as $\delta_h(I_\tau)$ in Equation (4).

$$\delta_h(I_\tau) = 2 \times \frac{h(\tau') - h(\tau)}{h(\tau) + h(\tau')} \tag{4}$$

The aforementioned features describe the behavior of an airplane during a certain time interval. We lastly introduce a feature that captures the flight-phase in which an interval occurs:

Phase: an approximation of the fraction of the total distance from origin to destination airport that has been completed after interval I_τ.

$$\delta_d^{ph}(I_\tau) = 1 - \frac{\Delta_\sigma(p(\tau'), p_D)}{\Delta_\sigma(p_O, p_D)} \tag{5}$$

In order to detect diversions, it is important to distinguish abnormal from regular behavior during an interval. What constitutes regular behavior, however, depends on the phase of the flight in which an interval occurs. This is because airplanes behave differently in different phases. Consider for example the velocity deviations $\delta_v(I_\tau)$ plotted in Figure 2 against the phase $\delta_d^{ph}(I_\tau)$. The continuous line depicts the behavior of a regular flight. At the start of the flight, the changes are the highest, as the airplane is ascending and gaining speed. In mid-flight, the velocity of an airplane is relatively stable. However, when an airplane nears its destination, it reduces its speed in order to prepare for the descend and landing. This results in large negative velocity deviations.
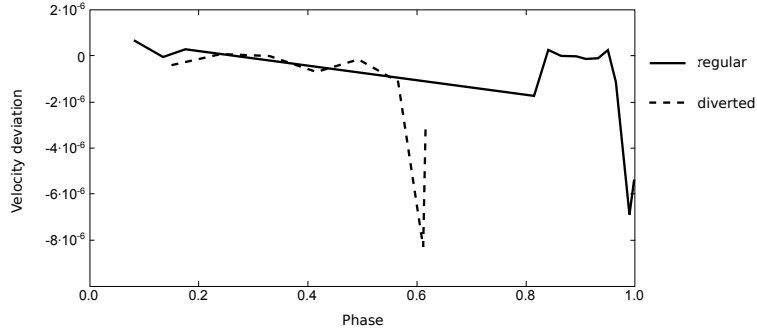
Figure 2: Velocity deviations for a regular and diverted flight plotted against the phase of the flight

By contrast, a diverted flight, depicted by the dashed line, shows highly different behavior. After a similar first half of the flight in comparison to the regular flight, the flight trajectory shows large decreases in speed around the point $\delta_d^{ph}(I_\tau) = 0.6$. This indicates that the airplane is greatly reducing its speed, as if to prepare for landing, while it is still far away from its destination airport. The combination of a such exceptional values of $\delta_v(I_\tau)$ with a low value for $\delta_d^{ph}(I_\tau)$ thus indicates that an airplane is landing at a different airport, i.e., a diversion. Similar differences between regular and diverting flights can also occur for other features. For instance, an airplane that is landing at a different airport will also show large fluctuations in altitude at a significant distance from its destination airport. The classification approach presented in Section 3.3 exploits these differences in behavioral features between regular and diverting flights in order to detect anomalous behavior.

3.3. Detecting Anomalous Behavior

Diverting airplanes exhibit behavior that is distinct from the behavior of non-diverting flights, because they are no longer traveling towards their intended destination. To identify diversions, it is thus vital to distinguish abnormal from regular behavior. In order to distinguish between regular and anomalous behavior during flight intervals, we make use of a classifier. We train the classifier to categorize intervals based on vectors of behavioral features, as described in Section 3.2.

With optimal data availability, a two-class classifier would be trained for this purpose. This requires a data set in which both classes, regular and anomalous flight behavior, are well-represented in and characterized by the available training data [43]. While it is relatively straightforward

13

to obtain the necessary data for regular flight intervals, obtaining sufficient data on anomalous flight intervals is a difficult endeavor. On the one hand, this difficulty follows from the relative infrequency with which diverted flights occur. Compared to non-diverting flights, diverting flights are a rare occurrence, which complicates the collection of data on these flights. On the other hand, even once data on diverting flights is obtained, it is cumbersome to identify which intervals of its flight trajectory should be classified as anomalous. This difficulty arises because an airplane that ends up diverting, generally starts off with behavior identical to that of non-diverting flights, i.e. the intervals in the first part of the trajectory are also regular for these flights. The airplane's anomalous behavior only manifests itself later on. Determining where exactly a trajectory switches from normal to abnormal is a fuzzy and laborious endeavor. To address this problem, we instead obtain a data set consisting of flight intervals obtained from just non-diverting flights. We use this data set to train a one-class classifier, which described the regular behavior during a flight. In particular, we make use of one-class SVMs for this task.

SVMs are the most commonly used classifiers for one-class classification problems [43]. Our choice for an SVM to tackle the classification task at hand is furthermore fueled by their suitability to handle high-dimensional, non-linear classification problems [31]. These characteristics are particularly relevant due to the complex patterns that must be identified to discriminate between anomalous and regular behavior. To enable the discovery of these non-linear relations, we train an SVM based on the Gaussian kernel. We furthermore have to account for the presence of noise and anomalous behavior in the training data. Though the training set only contains data obtained from non-diverting flights, this set can still contain anomalous flight intervals. Such intervals result from abnormal behavior during an otherwise normal flight, which may be caused by, e.g. weather conditions or due to efforts necessary to avoid collisions. To account for these outliers in the training data, we make use of ν-SVMs [44]. These adaptations of regular SVMs allow us to specify the (expected) level of noise in the data using a parameter ν.

To train the classifier for the task at hand, we have to optimize for three parameters: interval-length L, and SVM parameters ν and γ. Interval-length L describes the time range in which positional updates are gathered, and consequently the amount of behavior captured in a single

feature vector. Selecting a low L can result in a classifier that is sensitive to noise, but better captures short maneuvers. By contrast, while a high value for L likely yields a more robust classifier, it also affects the time it takes before a diversion prediction can be made. A too high L thus decreases the response time gained with our approach. The SVM ν and γ, respectively, capture the expected level of noise and the extend to which the decision hyperplane is fit to the training data. By fine-tuning ν, the support vectors are adjusted to the proper level of noise, whereas γ must be chosen in line with the characteristics of the data set. Selecting appropriate values for these parameters is paramount to the quality of the classification results and, hence, to the predictive accuracy of our prediction model. The interested reader is referred to Appendix A for a detailed description of one-class SVMs and the relevant parameters. The combination of best ν and γ parameters may depend on the chosen length of interval L, too. Therefore, the optimization process of ν and γ has to be conducted according to the chosen value of L.

3.4. Predicting Diversions

Information on a flight trajectory is received as a number of sequential flight track events. These are first converted into feature vectors, covering the evolution of flight data over given time intervals. The one-class classifier, described in the previous section, classifies each of these feature vectors as regular or anomalous. The first two steps of our prediction model thus represent any (partial) flight trajectory as a sequence of binary classifications. In its final step, our model converts this sequence into a prediction on whether or not a flight is diverting. Specifically, the model predicts a diversion when the number of consecutive anomalous flight intervals reaches a threshold t. Aside from its simplicity, this metric is intuitive because it is *robust* and *memoryless*.

Robustness is relevant because airplanes may exhibit anomalous behavior for brief periods of time, without leading to a diversion. Such behavior may occur due to, e.g. weather conditions, or collision avoidance. The metric t should account for this by not predicting a diversion at the first appearance of anomalous behavior. Only when an airplane behaves abnormally for a prolonged period of a time, a diversion should be predicted. Related to this, we select t as a memoryless metric, because early anomalous behavior in a flight should not affect predictions made later on. When a flight has returned to a normal course, i.e. it has started exhibiting regular behavior

again, prior anomalous behavior is no longer taken into account. Both objectives of robustness and memorylessness are thus achieved by taking t as the number of consecutive anomalous flight intervals.

The prediction represents the third and final step of our prediction model. In Section 4, we present an evaluation that demonstrates the prediction accuracy and response time gains achieved by using our model.

4. Evaluation

This section describes a quantitative evaluation of the proposed approach that demonstrates the performance on real flight data. Gathered results stem from the usage of a prototypical implementation of the prediction model described in this paper. It has been encoded in Java (for the data treatment and import) and Python (for the core prediction model), with the aid of the Scikit-learn library [45] providing the SVM algorithm. In Section 4.1 we present the setup for tests that we conducted, along with information on the metrics adopted to assess the accuracy of results. Section 4.2 provides an insight on the procedure that we followed to tune the model's parameters. Based on the criteria explained in Section 4.1 and the outcome of the optimization procedure of Section 4.2, the best performing classifiers are used to classify new unprocessed flight data. Section 4.3 and Section 4.4 describe the results achieved by the application of our technique on such data. The former focuses on the accuracy of results. The latter provides insights into the response time that can be gained using our prediction model. Section 4.5 concludes the section with a discussion.

4.1. Setting and Benchmark

This section describes the data and metrics that are used to evaluate the performance of the proposed approach. Available flight data is separated into three distinct datasets. The datasets have been acquired from FlightRadar24 and FlightStats. They report flight data of a period ranging from 10/07/2013 to 16/07/2013, and from 14/07/2013 to 11/08/2013, respectively. Data from FlightStats stem from flights mainly along the United States of America. Flight events from FlightRadar24 pertain flights taking off, or landing, in Europe – for the most part, following their

	S^{reg}						S^{div}		
Area	$S^{\mathrm{reg}_1}_{\mathrm{tv}}$	$S^{\mathrm{reg}_2}_{\mathrm{tv}}$	$S^{\mathrm{reg}_3}_{\mathrm{tv}}$	$S^{\mathrm{reg}_4}_{\mathrm{tv}}$	$S^{\mathrm{reg}_5}_{\mathrm{tv}}$	$S^{\mathrm{reg}_6}_{\mathrm{test}}$	$S^{\mathrm{div}_1}_{\mathrm{v}}$	$S^{\mathrm{div}_2}_{\mathrm{test}}$	Total
EU	126	122	125	120	126	127	24	22	792
US	51	55	52	57	51	50	10	12	338
Overall	177	177	177	177	177	177	34	34	1130

Table 1: Flights used for training, validating and testing the SVM, separated on the basis both of the flight area, i.e. Europe or USA, and the class they belong, i.e. regular (S^{reg}) or diverted (S^{div}).

routes within Europe. The interested reader is referred to Appendix B for a detailed description of the flight data we analyzed. Out of the collected flight tracks, we have sampled 1,130 flight tracks (792 over Europe, and 338 over the United States), out of which 1,062 were regular (746 EU, 316 US) and 68 diverted (46 EU, 22 US). These sets will be respectively identified with S^{reg} and S^{div}. We have utilized the flight tracks to build three logical sets, respectively adopted to train, test and validate the SVM adopted in our approach.

In the following, we refer to the set of flight tracks used to train the classifier as the training set ($\mathcal{S}_{\mathrm{t}}$). Recall from Section 3.3 that the SVM is a one-class classifier that is trained on behavior observed in regular flights. Therefore all trajectories in $\mathcal{S}_{\mathrm{t}}$ relate to non-diverting flights. The second dataset is the validation set ($\mathcal{S}_{\mathrm{v}}$) used to optimize the parameters of the classification approach. Finally, the test set ($\mathcal{S}_{\mathrm{test}}$) is used to objectively assess the performance of the approach with the optimal configuration found during the parameter optimization phase. Flight tracks in S_{test} are not used during the training nor the validation phases in order to avoid any bias in the assessment.

For the training and validation phase, we have adapted a K-fold cross-validation approach, with $K = 5$ [46]. To do so, we have first shuffled EU and US flights, and then divided them as follows. We have partitioned the set of 1,062 regular flight tracks (S^{reg}) in six non-overlapping subsets consisting of 177 flights each – hereafter, $S^{\mathrm{reg}_1}_{\mathrm{tv}}, \ldots, S^{\mathrm{reg}_5}_{\mathrm{tv}}, S^{\mathrm{reg}_6}_{\mathrm{test}}$. The set of 68 diverted flight tracks (S^{div}) has been sliced into two subsets of 34 flight tracks each – $S^{\mathrm{div}_1}_{v}$ and $S^{\mathrm{div}_2}_{v}$. In the notation adopted, the superscript of sets and subsets recalls whether comprised flight tracks are regular or diverted, resp. $^{\mathrm{reg}}$ or $^{\mathrm{div}}$. The subscript indicates whether the subsets are utilised for test ($_{\mathrm{test}}$),

17

training and validation ($_\text{tv}$), or validation only ($_\text{v}$).

Out of the six subsets of regular flight tracks, five have been used for the training and validation phase (hence, the "tv" subscript of $S_\text{tv}^{\text{reg}_1}, S_\text{tv}^{\text{reg}_2}, \ldots, S_\text{tv}^{\text{reg}_5}$). Over five rounds, we have assigned the combinations of four sets out of five to $\mathcal{S}_\text{t}$, for the training phase – for one of the five rounds, e.g., $\mathcal{S}_\text{t} = S_\text{tv}^{\text{reg}_1} \bigcup S_\text{tv}^{\text{reg}_3} \bigcup S_\text{tv}^{\text{reg}_4} \bigcup S_\text{tv}^{\text{reg}_5}$. We included the remaining subset ($S_\text{tv}^{\text{reg}_2}$ in the example) in the validation set, $\mathcal{S}_\text{v}$. Differently from the training phase, the validation phase takes into account diverted flights too. Therefore we also included $S_\text{v}^{\text{div}_1}$ in $\mathcal{S}_\text{v}$ – in the example round, $\mathcal{S}_\text{v} = S_\text{tv}^{\text{reg}_2} \bigcup S_\text{v}^{\text{div}_1}$. The sixth set extracted from the collection of regular flights, $S_\text{test}^{\text{reg}_6}$, has been used for the test, along with the remaining half of diverted flight tracks, $S_\text{test}^{\text{div}_2}$. Therefore, for all trained machines, $\mathcal{S}_\text{test} = S_\text{test}^{\text{reg}_6} \bigcup S_\text{test}^{\text{div}_2}$. Table 1 shows the distribution of flights over the United States and over Europe in the aforementioned sets.

Precision ($\mathcal{P}$), recall ($\mathcal{R}$), and F-score ($\mathcal{F}$) [47] are the classification metrics used to quantify the performance of the approach and a given parameter configuration. These scores are, respectively, formalized in Equations (6) to (8). In these equations, a true positive (tp) denotes a flight that is correctly predicted to divert, whereas a false positive (fp) denotes a flight that does not divert, yet is predicted to do so. A true negative (tn) and a false negative (fn), respectively, represent a correct and an incorrect prediction for a non-diverting flight.

$$\mathcal{P} = \frac{tp}{tp + fp} \qquad (6) \qquad\qquad \mathcal{R} = \frac{tp}{tp + fn} \qquad (7) \qquad\qquad \mathcal{F} = 2 \times \frac{\mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}} \qquad (8)$$

Precision indicates the fraction of predicted diverting flights that truly divert. Recall denotes the fraction of diverting flights that is predicted to divert. Finally, the F-score is the harmonic mean of the precision and recall [48].

4.2. Parameter Optimization

Recall that the performance of the prediction approach depends on four parameters: interval-length L, SVM parameters ν and γ, and threshold t. L represents the time interval between consecutive positional queries. The SVM parameters ν and γ, respectively capture the level of noise, and the desired similarity that intervals have to hold w.r.t. the training data in order to be

L [min]	t	ν	γ	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
2	8	0.06	8.00	0.89	0.74	0.81
3	6	0.02	8.00	0.86	0.75	0.80
4	3	0.01	1.00	0.91	0.75	0.82
5	4	0.03	4.00	0.88	0.71	0.79
6	3	0.02	0.50	0.95	0.65	0.77
7	3	0.01	2.00	0.90	0.66	0.76
8	2	0.01	0.25	0.84	0.70	0.76
9	2	0.01	0.50	0.84	0.68	0.75
10	2	0.01	1.00	0.80	0.79	0.80
11	2	0.01	0.50	0.86	0.68	0.76
12	2	0.01	0.50	0.87	0.64	0.73

Table 2: Performance of best models gathered with the coarse-grain grid-search, on the basis of the analyzed time-interval length
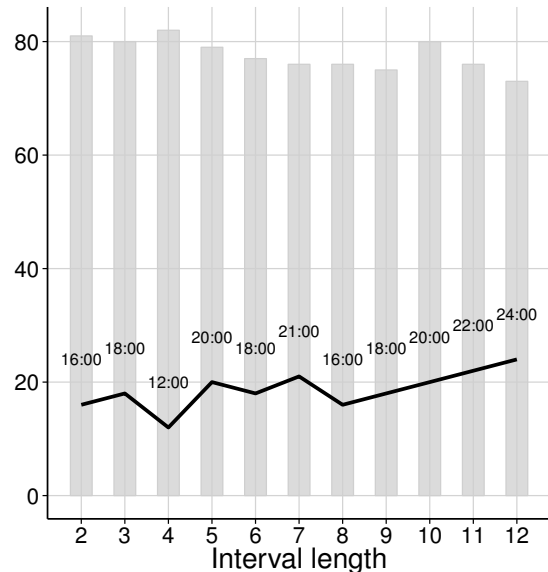

Figure 3: F-score (bars) and time needed to predict a diversion (line) in the validation phase of the SVM, with respect to the interval length

in turn classified as regular. Finally, t is the number of consecutive anomalous intervals that result in a diversion prediction. We optimize these parameters by performing a two-phase *grid search*. A grid search is an exhaustive search over a subset of possible values for each parameter. By starting out with a wide and coarse grid it is possible to identify an appropriate search region [49]. Afterwards, a finer grid search on the identified search region is conducted. The values that we selected for the coarse-grain grid search are:

$L \in \{2, \ldots, 12\}$ minutes $\qquad\qquad \nu \in \{0.01, 0.02, \ldots, 0.16\}$

$t \in \{1, \ldots, 10\}$ $\qquad\qquad\qquad\quad \gamma \in \{2^{-4}, 2^{-3}, \ldots, 2^4\}.$

For every combination of the aforementioned parameters, we trained 5 SVMs on the five different assignments of $\mathcal{S}_{\mathrm{t}}$, and validated them on the corresponding $\mathcal{S}_{\mathrm{v}}$. We thus assessed the predictions made in terms of precision, recall and F-score. Assigned values correspond to the average of the 5 cross-validations. Thereafter, for every pair of $\langle L, t \rangle$, we have selected the values for ν and γ that let the SVM attain the best F-score. Table 2 shows the gathered results. Our final objective is

19

to maximize the F-score, keeping the time needed for a diversion prediction (i.e., $L \cdot t$) as low as possible. Experiments show that the F-score is noticeably high (higher or equal to 0.8) if $L \leqslant 5$ min. The highest F-score is reached in correspondence with the minimum time-to-predict: indeed, when $L = 4$, predictions are given after 12 minutes, with $\mathcal{F} = 0.82$. Figure 3 shows how increasing sizes for L not only allow for lower F-score but also require more time for prediction – both trends impairing the results. A (partial) exception to the trend is reached at $L = 10$ ($\mathcal{F} = 0.8$), although the corresponding time to predict amounts to 20 minutes.

Based on the outcome of the coarse-grained search, we conducted a finer-grain grid search. Values for L were the sequence of quarters of minutes from 2 to 5 minutes – hence, one value assigned to L every 15 seconds. The values for ν, γ and t were restricted to those between the minimum and the maximum from the best performing combinations, as follows.

$L \in \{8, \ldots, 20\}$ quarters of minutes $\qquad\qquad \nu \in \{0.01, 0.02, \ldots, 0.14\}$

$t \in \{2, \ldots, 8\}$ $\qquad\qquad\qquad\qquad\qquad\qquad \gamma \in \{2^0, 2^1, \ldots, 2^4\}.$

Figure 4 graphically represents the accuracy of predictions made during the cross-validation. More specifically, Figure 4a depicts the precision-recall graph. In the figure, crosses represent the performances of all trained SVMs during the validation phase. For every interval length, the parameters combinations yielding the highest F-score have been saved. Their precision, recall, and F-score are depicted in Figure 4b. An empty circle, an empty square, and an empty diamond respectively represent those ones that achieved the best precision, recall, and F-score. They are reported both in Figure 4a and Figure 4b. Even after the fine-grain grid search, the best F-score is still achieved at an interval-length of 4 minutes ($L = 240$ seconds, $t = 3$, $\gamma = 1$, $\nu = 0.01$). Considering the best set-ups in terms of F-score for each interval, the topmost precision and F-score are attained having an interval length of 150 seconds and 225 seconds, respectively. Figure 4b shows that the best achieved F-score values balance recall and precision, respectively having peaks in the range of 195–255 seconds, and 120–175 seconds.
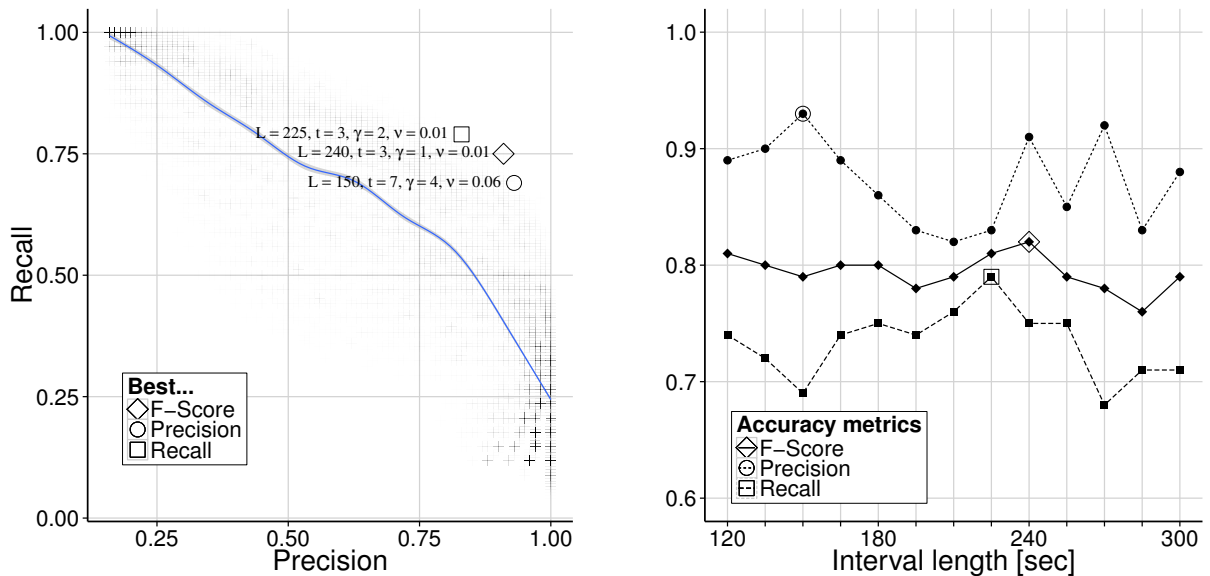
(a) Precision-recall graph for the validation phase of the SVM.

(b) The trend of best F-score values gained in the validation phase of the SVM with respect to the interval length, and the corresponding precision and recall.

Figure 4: Accuracy of prediction gained in the validation phase of the framework.

4.3. Effectiveness

In order to assess the potential of the proposed prediction model, we applied it on test set S_{test}. In particular, we have selected the best SVMs, ranked according to the F-score achieved in the validation phase, according to the different values of L, the time interval. The trend of resulting precision, recall and F-score is outlined in Table 3. Gathered test results are in line with the validation phase, thus showing that the classifiers do not suffer from overfitting with respect to the training data. Indeed, the minimum and maximum F-score respectively amount to 0.76 and 0.82, as in the validation phase. The corresponding values for precision and recall float in the range 0.78–0.96, and 0.68–0.79. The best set-up retrieved for the parameters optimization was achieved with an interval length of 240 seconds. During the test phase, such set-up is ranked the second with respect to the F-score, because its corresponding value is 0.81 whereas the set-up at 120 seconds reaches a value of 0.82. Nevertheless, the difference remains limited to a negligible amount (0.01).

In general terms, a higher recall may be more beneficial than a higher precision: deviations are

L [sec]	t	ν	γ	Precision	Recall	F-score	L [sec]	t	ν	γ	Precision	Recall	F-score
120	8	0.06	8.00	0.91	0.74	0.82	225	3	0.01	2.00	0.78	0.79	0.78
135	7	0.06	4.00	0.81	0.72	0.76	240	3	0.01	1.00	0.88	0.75	0.81
150	7	0.06	4.00	0.83	0.69	0.76	255	3	0.01	2.00	0.80	0.75	0.77
165	4	0.02	2.00	0.81	0.74	0.77	270	4	0.01	2.00	0.96	0.68	0.79
180	6	0.02	8.00	0.83	0.75	0.78	285	4	0.07	1.00	0.85	0.71	0.77
195	4	0.02	4.00	0.81	0.74	0.77	300	4	0.03	4.00	0.93	0.71	0.80
210	4	0.03	4.00	0.83	0.76	0.79							

Table 3: Performance of the best models on S_{test}, on the basis of best-ranked parameter combinations for SVMs, gathered during the validation phase.

far less frequent than regular flights, therefore false alarms (i.e., the drawback of a non-optimal precision) can be considered less damaging than overlooked deviations from the final destinations (i.e., the drawback of a non-optimal recall). Indeed, one can inspect flight data and verify whether the airplane is actually showing an unexpected behavior, once it is classified as diverting; on the other hand, if all the other flights (i.e., the vast majority) had to be checked to search for possible unreported anomalies, the approach would be of little to no avail.

4.4. Response Time Gain

We have shown that our prediction model is able to predict diverting flights with arguably high precision and recall. Despite this achievement, it is important to note that diversion detection is merely a means, rather than a goal, in the context of freight scenarios. The true goal of this model is to enable logistic service providers to respond adequately to diverting airplanes. The contribution of the prediction model for this lies in the response time gained by predicting diversions, instead of awaiting the delayed communication from airlines. These gains, straightforwardly, enable logistic service providers to react earlier to deviations, and thereby improve the impact of their corrective actions.

The scenario description of Section 2.1 indicates that logistic service providers need to deal with two separate issues in case of a flight deviation. Firstly, road transportation (e.g. trucks) needs to be arranged to pick up the cargo that arrives at the new destination airport. This issue is

crucial, since the quality and timeliness of this response affects a logistic service provider's ability to meet its service level objectives. Secondly, however, a logistic service provider also has to deal with trucks that have been assigned to pick up the cargo at the original destination airport. If a diversion is not recognized in time, these trucks travel unnecessary distances, resulting in additional costs and CO_2 emission. Note that in some cases it is possible to tackle both issues with a single response, namely by redirecting trucks assigned to the original destination to the new location. Nevertheless, the two issues call for separate metrics:

Time-gain w.r.t. planned arrival time $\Delta_t^{\mathrm{planned}}$: response time gained to cancel or redirect road transportation assigned to pick up cargo at the original arrival airport. $\Delta_t^{\mathrm{planned}}$ is computed as the difference between the planned arrival time t_{arr}^P and the time at which a diversion is predicted t_{div}, i.e. $\Delta_t^{\mathrm{planned}} = t_{div} - t_{arr}^P$. Note that t_{div} is the time at which an event is received that results in a predicted diversion, i.e. the last event of the t-th consecutive anomalous interval (e.g., the third one, assuming threshold $t = 3$).

Time-gain w.r.t. actual arrival time $\Delta_t^{\mathrm{actual}}$: response time gained to arrange road transportation to pick up cargo at the new arrival airport. $\Delta_t^{\mathrm{actual}}$ is computed as the difference between the actual arrival time (at the new destination airport) t_{arr}^A and t_{div}, i.e. $\Delta_t^{\mathrm{actual}} = t_{div} - t_{arr}^A$.

Using the configuration that performed best in terms of highest F-score in the validation phase ($L = 4$ min, $t = 3$, $\nu = 0.01$, $\gamma = 1$), our prediction model identified 55 diversions out of 68 diverted flights in S^{div} – 28 out of 34 in the validation phase, and 29 out of 34 in the test phase. Table 4 shows the response time gains for these 55 flights. The approach is on average able to predict a diversion 120 minutes before the originally scheduled landing time, and 62 minutes before the actual landing occurs. This gives logistic service providers more than one hour to react to a probable diversion. This is a significant gain in comparison to the case where logistic service providers have to wait for a notification of the diversion, which often occurs up to two hours *past* the actual landing time.

Metric	Average	Median	Minimum	Maximum
$\Delta_t^{\mathrm{planned}}$	02:00:54	01:01:53	−01:14:17	14:19:27
$\Delta_t^{\mathrm{actual}}$	01:02:48	00:32:20	00:00:00	09:15:38

Table 4: Response time gains

4.5. Discussion

In this section, we discuss our approach using real-world examples from our experiments. Our implemented prototype has been integrated with the GET Service project software platform for smart monitoring and planning of logistic processes. It is also currently being evaluated by one of the flight data providers. We examine here three selected prediction results, focusing on two examples of correctly predicted diversions and an undetected diversion. In the following, all registered and scheduled times are adapted to the Coordinated Universal Time (UTC) standard.

Both flights in Figure 5 are scheduled to go from Munich, Germany, to London, UK. The regular trajectory is respected in the case of Figure 5a. In the case of Figure 5b, instead, the flight diverts and lands back in Munich. The scheduled take-off and landing times are respectively 07:20 and the landing time is 09:05. The expected flight-time thus amounts to 1 hour and 45 minutes. The non-diverted flight takes approximately 1 hour and 40 minutes, as well as the diverted one. However, the latter makes a U-turn towards the origin airport, as shown in Figure 5b. Using our approach, we can predict the diversion before the actual landing of the airplane in an unexpected airport. Specifically, the take-off takes place at 07:42:57, hence with a delay of approximately 20 minutes, and the landing at 09:22:35 . Using the SVM that gained the best F-score in the validation phase (i.e., $L = 4$ min, $t = 3$, $\nu = 0.01$, $\gamma = 1$), the diversion is predicted when the airplane is localized at the coordinates highlighted by a circle and labeled as "Diversion predicted" in Figure 5b. The correspoding event in the flight tracking is timestamped 08:48:03. As a consequence, the diversion is predicted approximately 34 minutes ahead of the actual landing time, i.e., $\Delta_t^{\mathrm{actual}} = 34{:}32$. If we consider the delay of 20' accumulated on the take-off to be added also to the expected time of arrival, as it usually happens in reality, the prediction is made 37 minutes before the expected time of arrival, $\Delta_t^{\mathrm{planned}} = 36{:}57$.

An example of correctly predicted diversion on a longer-haul flight is depicted in Figure 6. The

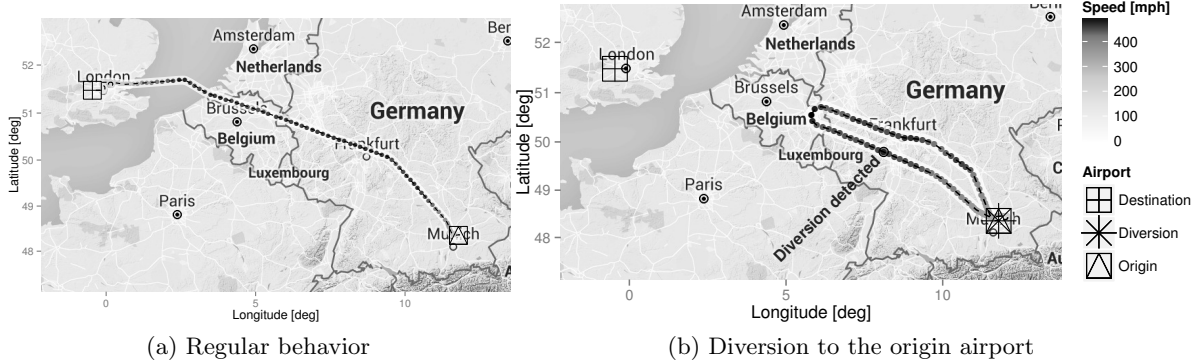(a) Regular behavior (b) Diversion to the origin airport

Figure 5: Representation of a real case study from Munich to London

flight departs at 12:46:57 from Antalya, Turkey, in time with its scheduled departure at 12:45. Its expected arrival is in Skellefteå, Sweden, after 4 hours and 10 minutes. The flight diverts in Trondheim, Norway, at 18:34:22. Differently from the previous case, no U-turn is drawn by the trajectory of the airplane. Nevertheless, the diversion is predicted by our approach at 17:32:15, approximately one hour before the actual landing: $\Delta_t^{\text{actual}} = 62:07$. Considering the ETA instead, the prediction is made 1 hour and 23 minutes before the expected time of landing: $\Delta_t^{\text{planned}} = 82:45$. Appendix B reports in detail the processed data and the analysis results for the aforementioned example flights.

Figure 7 shows an example of one of the diverted flights that were not recognized as such by our approach (i.e., false negatives) during the test phase. As shown in the figure, the airplane takes off at Geneva, and terminates its flight in London. However, it diverts to Stansted airport, instead of landing at Gatwick. The reason for the missed diversion prediction may reside in the rapid manoeuvres that the pilot does in proximity of the final destination: the irregular behavior is detected during intervals of few minutes each. Indeed, the first anomaly is detected approx. 25 minutes before landing, and further single anomalous intervals are signalled in the following 12 and 8 minutes: not enough time to reach the third anomaly in a row, for time-intervals of 4 minutes. In fact, we recall here that the parameter setting for this evaluation is such that $L = 4$ and $t = 3$. As a consequence, 12 minutes are needed to make predictions. The detected consecutive anomalies amounted at most to 2, i.e., not a sufficient number to classify the flight as diverted. This suggests a further investigation towards better techniques to determine situations that show a high

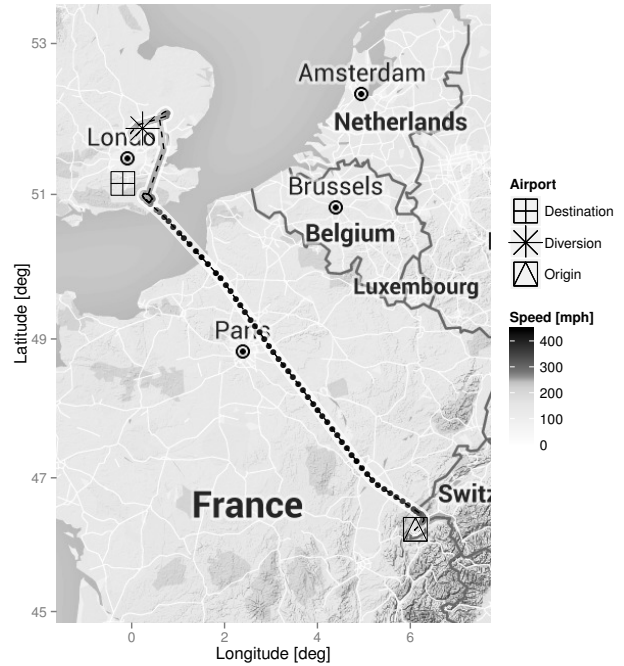Figure 6: From Antalya to Skellefteå, diverted in Trondheim.

Figure 7: From Geneva to London Heathrow, diverted at London Stansted.

possibility of diversion, independent of the reiteration of the irregular behavior. Nevertheless, it is also worth to be noticed that even if the predictions were made immediately after the first diverting manoeuvres, the time-gain w.r.t. the actual landing time would be negligible. Furthermore, the distance separating the airports of Gatwick and Stansted is relatively short. Similar conditions hold for five of the remaining false negatives, here omitted for the sake of space. In all such cases, the anomalous behavior is shown at the end of the flight, for a time which is not sufficient to be considered as a sign of probable diversion.

5. Conclusions

In this paper we tackle synchronization problems in multi-modal transport and the challenge to timely react to unexpected behavior. Our contribution is a model for the prediction of flight diversions based on the automated detection of anomalous behavior. In contrast to prior research, our technique does not require information on planned flight paths, and still provides predictions of diversions with high accuracy. We model the flight trajectory as a sequence of positional updates

26

that describe flight's location, altitude and velocity. Such data is transformed into relevant features that characterize the behavior of an airplane during a time interval, which are processed by a one-class classifier. We evaluated our technique on an extensive set of real-world data demonstrating its accuracy in terms of the F-measure and a substantial time-to-prediction gain.

We plan to extend our work in several ways in the future. Firstly, we intend to expand the approach such that it not only predicts the occurrence of diversions, but also computes to which airport the airplane will most likely divert. Also, knowing how diversions can be predicted for airplanes, we intend to investigate the prediction of breakdowns and diversions in other transportation contexts, e.g. road, inland waterway or railway transportation, such that the model can be used in any multi-modal logistics scenario.

Acknowledgements

Bibliography

[1] J. Leukel, V. Sugumaran, Formal correctness of supply chain design, Decision Support Systems 56 (2013) 288–299. `doi:10.1016/j.dss.2013.06.008`.

[2] S. Treitl, P. Rogetzer, M. Hrusovsky, C. Burkart, B. Bellovoda, W. Jammernegg, J. Mendling, E. Demir, T. van Woensel, R. Dijkman, M. van der Velde, A.-C. Ernst, GET Service Project - Deliverable 1.1: Use Cases, Success Criteria and Usage Scenarios (2013).
URL `http://getservice-project.eu/Documents2/GET%20Service%20D1_1%20Use%20Cases%20Success%20Criteria%20and%20Usage%20Scenarios%20v1_1%20Public.pdf`

[3] J. Krozel, Intelligent Tracking of Aircraft in the National Airspace System, in: American Institute of Aeronautics and Astronautics, 2002.

[4] T. G. Reynolds, R. J. Hansman, Conformance Monitoring Approaches in Current and Future Air TrafficControl Environments, digital Avionics Systems Conference (October 2002).

[5] D. Arnott, G. Pervan, Eight key issues for the decision support systems discipline, Decision Support Systems 44 (3) (2008) 657–672.

[6] J. Shim, M. Warkentin, J. F. Courtney, D. J. Power, R. Sharda, C. Carlsson, Past, present, and future of decision support technology, Decision Support Systems 33 (2) (2002) 111–126.

[7] K. Werner, A. Schill, Automatic Monitoring of Logistics Processes using Distributed RFID based Event Data, in: International Workshop on RFID Technology (IWRT), INSTICC PRESS, 2009, pp. 101–108.
URL http://dblp.uni-trier.de/db/conf/iwrt/iwrt2009.html#WernerS09

[8] E. W. T. Ngai, T. K. P. Leung, Y. H. Wong, M. C. M. Lee, P. Y. F. Chai, Y. S. Choi, Design and Development of a Context-aware Decision Support System for Real-time Accident Handling in Logistics, Decis. Support Syst. 52 (4) (2012) 816–827. doi:10.1016/j.dss.2011.11.016.

[9] W. Burgholzer, G. Bauer, M. Posset, W. Jammernegg, Analysing the impact of disruptions in intermodal transport networks: A micro simulation-based model, Decision Support Systems 54 (4) (2013) 1580–1586, rapid Modeling for Sustainability.

[10] M. Gariel, A. N. Srivastava, E. Feron, Trajectory Clustering and an Application to Airspace Monitoring, Trans. Intell. Transport. Sys. 12 (4) (2011) 1511–1524. doi:10.1109/TITS.2011.2160628.

[11] Y. Guo, Q. Xu, Y. Yang, S. Liang, Y. Liu, M. Sbert, Anomaly detection based on trajectory analysis using kernel density estimation and information bottleneck techniques, Tech. rep., Technical report 108, University of Girona (2014).

[12] Y. Guo, Q. Xu, S. Liang, Y. Fan, M. Sbert, Xaibo: An extension of aib for trajectory clustering with outlier, in: S. Arik, T. Huang, W. K. Lai, Q. Liu (Eds.), Neural Information Processing - 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part II, Vol. 9490 of Lecture Notes in Computer Science, Springer, 2015, pp. 423–431. doi:10.1007/978-3-319-26535-3_48.
URL http://dx.doi.org/10.1007/978-3-319-26535-3_48

[13] E. Smart, D. J. Brown, J. Denman, A Two-Phase Method of Detecting Abnormalities in Aircraft Flight Data and Ranking Their Impact on Individual Flights, IEEE Transactions on Intelligent Transportation Systems 13 (3) (2012) 1253–1265. doi:10.1109/TITS.2012.2188391.

[14] C. J. C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Min. Knowl. Discov. 2 (2) (1998) 121–167. doi:10.1023/A:1009715923555.

[15] C. Jesse, H. Liu, E. Smart, D. Brown, Analysing Flight Data Using Clustering Methods, in: International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Vol. 5177, Springer, 2008, pp. 733–740. doi:10.1007/978-3-540-85563-7_92.

[16] J. A. Besada, G. Frontera, J. Crespo, E. Casado, J. Lopez-Leones, Automated Aircraft Trajectory Prediction Based on Formal Intent-Related Language Processing, IEEE Transactions on Intelligent Transportation Systems 14 (3) (2013) 1067–1082. doi:10.1109/TITS.2013.2252343.

[17] C. R. Hargraves, S. W. Paris, Direct trajectory optimization using nonlinear programming and collocation, in: AAIA Guidance, Control and Dynamics, Vol. 10, 1987, pp. 388–342.

[18] C. Zhang, N. Wang, J. Chen, Trajectory generation for aircraft based on differential flatness and spline theory,

in: International Conference on Information Networking and Automation (ICINA), Vol. 1, 2010, pp. V1–110–V1–114. `doi:10.1109/ICINA.2010.5636425`.

[19] A. Nilim, L. E. Ghaoui, V. Duong, M. Hansen, Trajectory-based air traffic management under weather uncertainty, in: USA/Europe Air Traffic Management Research and Development Seminar, 2001.

[20] J. Pannequin, A. Bayen, I. Mitchell, H. Chung, S. Sastry, Multiple aircraft deconflicted path planning with weather avoidanceconstraints, in: AIAA Guidance, Navigation and Control Conference, 2007.

[21] M. Kamgarpour, V. Dadok, C. Tomlin, Trajectory generation for aircraft subject to dynamic weather uncertainty, in: IEEE Conference on Decision and Control (CDC), 2010, pp. 2063–2068. `doi:10.1109/CDC.2010.5717889`.

[22] D. B. Kirk, Initial functional performance assessment of a terminal airspace conflict probe application, in: AIAA GN&C, 2003.

[23] S. J. Landry, X. W. Chen, S. Y. Nof, A decision support methodology for dynamic taxiway and runway conflict prevention, Decis. Support Syst. 55 (1) (2013) 165–174. `doi:10.1016/j.dss.2013.01.016`.

[24] H. Tang, J. E. Robinson, D. G. Denery, Tactical Conflict Detection in Terminal Airspace, Journal of Guidance, Control, and Dynamics 34 (2) (2011) 403–413.

[25] S. D. Timar, K. Griffin, S. Borener, C. Knickerbocker, Analysis of s-turn approaches at John F. Kennedy airport, in: Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st, IEEE, 2012, pp. 3C1–1.

[26] K. P. Sycara, Machine learning for intelligent support of conflict resolution, Decis. Support Syst. 10 (2) (1993) 121–136. `doi:10.1016/0167-9236(93)90034-Z`.

[27] D. Sislák, P. Volf, M. Pechoucek, Agent-Based Cooperative Decentralized Airplane-Collision Avoidance, IEEE Transactions on Intelligent Transportation Systems 12 (1) (2011) 36–46. `doi:10.1109/TITS.2010.2057246`.

[28] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis, et al., Semantic trajectories modeling and analysis, ACM Computing Surveys (CSUR) 45 (4) (2013) 42.

[29] F. Lettich, L. O. Alvares, V. Bogorny, S. Orlando, A. Raffaetà, C. Silvestri, Detecting avoidance behaviors between moving object trajectories, Data & Knowledge Engineering.

[30] J. Owens, A. Hunter, Application of the Self-Organizing Map to Trajectory Classification, in: IEEE International Workshop on Visual Surveillance (VS), IEEE Computer Society, 2000, pp. 77–.

[31] C. Piciarelli, C. Micheloni, G. Foresti, Trajectory-Based Anomalous Event Detection, IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1544–1554. `doi:10.1109/TCSVT.2008.2005599`.

[32] Y. Zhou, S. Yan, T. Huang, Detecting Anomaly in Videos from Trajectory Similarity Analysis, in: IEEE International Conference on Multimedia and Expo, 2007, pp. 1087–1090. `doi:10.1109/ICME.2007.4284843`.

[33] M. Watagawa, E. Kobayashi, N. Wakabayashi, Monitoring of vessel traffic using AIS data and ALOS satellite image, in: OCEANS, 2012, pp. 1–4. `doi:10.1109/OCEANS-Yeosu.2012.6263457`.

[34] B. Storey, R. Holtom, The use of historic GPS data in transport and traffic monitoring, Traffic Engineering

Control 44 (10) (2003) 376–379.

[35] E. Mazloumi, G. Currie, G. Rose, Using GPS data to gain insight into public transport travel time variability, Journal of Transportation Engineering 136 (7) (2009) 623–631.

[36] C. Chen, D. Zhang, P. S. Castro, N. Li, L. Sun, S. Li, Real-time detection of anomalous taxi trajectories from GPS traces, in: Mobile and Ubiquitous Systems: Computing, Networking, and Services, Springer, 2012, pp. 63–74. `doi:10.1007/978-3-642-30973-1_6`.

[37] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, S. Li, iBAT: Detecting Anomalous Taxi Trajectories from GPS Traces, in: International Conference on Ubiquitous Computing, ACM, 2011, pp. 99–108. `doi:10.1145/2030112.2030127`.

[38] L. X. Pang, S. Chawla, W. Liu, Y. Zheng, On Detection of Emerging Anomalous Traffic Patterns Using GPS Data, Data Knowl. Eng. 87 (2013) 357–373. `doi:10.1016/j.datak.2013.05.002`.

[39] J. A. Barria, S. Thajchayapong, Detection and Classification of Traffic Anomalies Using Microscopic Traffic Variables, IEEE Transactions on Intelligent Transportation Systems 12 (3) (2011) 695–704. `doi:10.1109/TITS.2011.2157689`.

[40] Q.-J. Kong, Q. Zhao, C. Wei, Y. Liu, Efficient Traffic State Estimation for Large-Scale Urban Road Networks, IEEE Transactions on Intelligent Transportation Systems 14 (1) (2013) 398–407. `doi:10.1109/TITS.2012.2218237`.

[41] S. Dunne, B. Ghosh, Weather Adaptive Traffic Prediction Using Neurowavelet Models, IEEE Transactions on Intelligent Transportation Systems 14 (1) (2013) 370–379. `doi:10.1109/TITS.2012.2225049`.

[42] C. Cabanillas, C. Di Ciccio, J. Mendling, A. Baumgrass, Predictive Task Monitoring for Business Processes, in: Business Process Management (BPM), Vol. 8659, Springer, 2014, pp. 424–432. `doi:10.1007/978-3-319-10172-9_31`.

[43] S. S. Khan, M. G. Madden, One-class classification: taxonomy of study and review of techniques, The Knowledge Engineering Review 29 (03) (2014) 345–374.

[44] B. Schölkopf, A. J. Smola, R. C. Williamson, P. L. Bartlett, New Support Vector Algorithms, Neural Computation 12 (5) (2000) 1207–1245. `doi:10.1162/089976600300015565`.

[45] 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, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830. URL `http://dl.acm.org/citation.cfm?id=2078195`

[46] R. Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in: International Joint Conference on Artificial Intelligence, 1995, pp. 1137–1145.

[47] T. M. Mitchell, Machine Learning, 1st Edition, McGraw Hill series in computer science, McGraw-Hill, Inc., New York, NY, USA, 1997.

[48] R. A. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co.,

Inc., Boston, MA, USA, 1999.

[49] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification (2010).

[50] C. Cortes, V. Vapnik, Support-Vector Networks, Machine Learning 20 (3) (1995) 273–297. `doi:10.1007/`
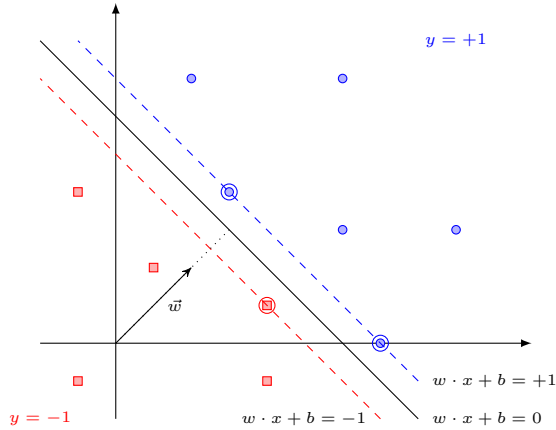`BF00994018`.

Figure A.8: Classification hyperplane separating two classes.

Appendix A. One-class Support Vector Machines

This appendix presents the details of the one-class Support Vector Machines (SVMs) used in our prediction model to distinguish regular from anomalous flight behavior, as described in Section 3.3.

Support Vector Machines (SVMs) are a particular kind of classifiers. In machine learning, *classifiers* aim to assign an unknown data object to one of several pre-defined categories. Classifiers are inferred from (i.e. *trained on*) a labeled set of training examples. SVMs take a feature vector $\vec{x_i} = (x_i^1, \ldots, x_i^l)$ as input. SVMs classify these inputs on the basis of the position a data point has in a numeric hyperspace, on which its features are mapped [50]. To achieve this, SVMs construct a hyperplane around each class in the training data. The edge of this hyperplane constitutes a *decision boundary* that characterizes the instances in the class. Figure A.8 illustrates this for a linear classification problem. There, the dashed lines represent the hyperplanes that provide the decision boundaries for the classes. The vectors on these decision boundaries are referred to as *support vectors*, the namesake of SVMs.

Since support vectors, by definition, capture linear relations, a transformation step is required to solve non-linear classification problems. These transformations map the input data in a higher-dimensional feature space, where the problem does have a linear solution. Figure A.9 illustrates this with an example. There, non-linear relations are converted into a linear problem by applying a transformation based on a quadratic function. SVMs can use a broad spectrum of such *kernel functions* to perform transformations. In this paper, we adopt the widely-employed *Gaussian*

kernel, which is defined as follows:

$$k(\vec{x}, \vec{x}') = \exp\left\{-\frac{\|\vec{x} - \vec{x}'\|^2}{2\sigma^2}\right\} = \exp\left\{-\gamma\|\vec{x} - \vec{x}'\|^2\right\}. \tag{A.1}$$

The parameter $\gamma = \frac{1}{2\sigma^2}$ plays an important role for SVMs based on the Gaussian kernel. Since this parameter captures the standard deviation of the data, its value determines to what extend the decision hyperplane is fit to the training data. As shown in Figure A.10, the higher the value for γ, the more the vectors overfit the data. Choosing a value for γ in line with the characteristics of the data set is therefore paramount to the success of an SVM using the Gaussian kernel.

The amount of noise in the training data set represents a second characteristic that affects the performance of an SVM. To account for the presence of noise, adaptations of SVMs exist that let a fraction of data (i.e. the *outliers*) fall in the area between the support vectors and the hyperplane. In this work we adopt the approach by Schölkopf et al. [44] to achieve this. Their technique, called ν-SVC, uses a parameter $\nu \in [0, 1]$ to represent the fraction of acceptable outliers in the training data. For instance, with $\nu = 0.1$, at most 10% of the data will be treated as outliers. Figure A.11 demonstrates the impact of ν on the classification performance. By fine-tuning ν, the support vectors are adjusted to the proper level of noisiness, resulting in a better fitted classifier.

For classification tasks with two classes, e.g. regular and anomalous behavior, ideally a two-class classifier is trained. However, this requires that both classes are well-represented in and characterized by the available training data [43]. If this is not the case, a *one-class* classifier can be trained instead. One-class classification, also referred to as anomaly detection, requires only information on a single class, i.e. the *target class*. These classifiers are particularly useful when plenty of data is available on the target class, while the other class is severely undersampled. A one-class classifier describes the characteristics of this target class. Any new data object that does not conform to these characteristics, is recognized as an anomaly. Due to the limited availability of training data for anomalous behavior, in this work we make use of such one-class classification.

Appendix B. Data sets

In this appendix, we provide a detailed view into the data treated by the implementation of our prediction model. Flight tracking data was collected from Flightstats[6] and FlightRadar24[7]. As a repository of historic flight data, flight events were extracted and stored in a period ranging from 14/07/2013 to 11/08/2013 (Flightstats) and from 10/07/2013 to 16/07/2013 (FlightRadar24). The formats of flight track events were different, according to the platform: FlightRadar24 provided JSON-formatted data (see Listing 1), whereas FlightStats events were recorded in XML documents (Listing 2). Two data adapters were thus implemented to import both formats and unify the heterogeneous information. The flight events of both flight tracking data comprised the following attributes: (i) a unique flight identifier, (ii) an aircraft identifier, (iii) the flight code, (iv) the time and date of the event, (v) the IATA/FAA codes of departure and arrival airports, (vi) the geographical coordinates of the aircraft, (vii) the altitude of the aircraft, and (viii) the speed of the aircraft. As a third source of information, we also accessed the open CSV databse of the OpenFlights.org portal[8], which provides the exact geographical coordinates of world-wide airports. This allowed us to link the IATA/FAA codes of origin and destination airports of flight track events with their actual position. Table B.5 lists the attributes of the unified flight events used as a primary input for our prediction model.

To represent the progress of the flight, we have grouped consecutive flight track events within a time interval, and extracted the following gains achieved during the interval (features – see Section 3.2 in the paper): (i) the gain w.r.t. the distance remaining to the arrival airport, (ii) the gain w.r.t. the distance from the departure airport, (iii) the gain in speed, (iv) the gain in altitude. Tables B.6 and B.7 show how such features were extracted from the track of a diverted flight from Munich to Heathrow, and from Antalya to Skellefteå, respectively. Both case studies are described in the paper in Section 4.5. For the sake of readability, the aforementioned features are abbreviated in the table as "Dist. left", "Dist. gained", "Diff. speed" and "Diff. alt.", respectively. Time intervals amount to 4 minutes in the example. A subset of the event attributes is also reported in the table

---

[6]http://www.flightstats.com/
[7]http://www.flightradar24.com/
[8]http://openflights.org/data.html

for the last event in the interval. Extracted features are consecutively examined and detected anomalies are collected. Once a number of subsequent anomalies lie above a given threshold (3, in the example), a diversion is predicted. Tables B.6 and B.7 show the data analysed along with the analysis results for the example flights. The events leading to the predictions of the diversions is highlighted in the table.

Listing 1: A sample from FlightRadar24 data

```
{
 "175603a": [
  "aircraft_id": "7380a9",
  "latitude": 43.0913, "longitude": 17.1492,
  "altitude": 37000,
  "speed": 468,
  "timestamp": 1373414405,
  "origin": "TLV", "destination": "CDG",
  "flight": "LY323",
  ...
 ],
 "171e09e": [
  "aircraft_id": "a96c96",
  "latitude": 40.593, "longitude": -73.4894,
  "altitude": 4000,
  "speed": 244,
  "timestamp": 1372636803,
  "origin": "SEA", "destination": "JFK",
  "flight": "DL1154",
  ...
 ],
 ...
}
```

Listing 2: A sample from FlightStats data

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<response><!-- ... -->
 <appendix>
  <airports>
   <airport>
    <fs>JFK</fs><iata>JFK</iata><!-- ... -->
   </airport>
   <airport>
    <fs>LHR</fs><iata>LHR</iata><!-- ... -->
   </airport>
  </airports>
 </appendix>
 <flightTracks>
  <flightTrack>
   <flightId>305226070</flightId><!-- ... -->
   <departureAirportFsCode>JFK</departureAirportFsCode><arrivalAirportFsCode>LHR</arrivalAirportFsCode><!-- ... -->
   <positions>
   <position>
    <lon>-0.4185999929904938</lon><lat>51.46500015258789</lat>
    <speedMph>181</speedMph>
    <altitudeFt>150</altitudeFt><!-- ... -->
    <date>2013-08-08T05:24:52.000Z</date>
   </position>
   <position>
    <lon>-0.41609999537467957</lon><lat>51.46500015258789</lat>
    <speedMph>181</speedMph>
    <altitudeFt>175</altitudeFt><!-- ... -->
    <date>2013-08-08T05:24:47.000Z</date>
   </position>
   <!-- ... -->
  </flightTrack>
 </flightTracks>
</response>
```

Listing 3: A sample from OpenFlights.org data

```
id,name,city,country,IATAFAA,ICAO,latitude,longitude,altitude,timezone,DST
...
346,"Franz Josef Strauss",Munich,Germany,MUC,EDDM,48.353783,11.786086,1487,1,E
347,Nurnberg,Nuernberg,Germany,NUE,EDDN,49.4987,11.066897,1046,1,E
348,"Leipzig Halle",Leipzig,Germany,LEJ,EDDP,51.432447,12.241633,465,1,E
349,Saarbrucken,Saarbruecken,Germany,SCN,EDDR,49.214553,7.109508,1058,1,E
350,Stuttgart,Stuttgart,Germany,STR,EDDS,48.689878,9.221964,1276,1,E
351,Tegel,Berlin,Germany,TXL,EDDT,52.559686,13.287711,122,1,E
352,Hannover,Hannover,Germany,HAJ,EDDV,52.461056,9.685078,183,1,E
...
```

| Event attribute	Description	Units	Example type
eventID	Progressive number identifying the event		19473339
aircraftID	Identifier of the aircraft		3958344
flightID	Identifier of the flight		24936159
flightCode	Concatenation of airline code and flight number		LH2472
timestamp	Occurrence time of this event		2013-07-14 07:51:02
departure	IATA/FAA code and location of the origin airport	<IATA code, deg, deg>	<MUC, 48.353783, 11.786086>
arrival	IATA/FAA code and location of the destination airport	<IATA code, deg, deg>	<LHR, 51.4775, -0.461389>
location	Coordinates of the aircraft (resp., latitude and longitude)	<deg, deg>	<49.7942, 8.1168>
altitude	Altitude of the aicraft	Feet	36050
speed	Ground speed of the aircraft	knots	448

Table B.5: Flight events description

Event attributes					Features				Anomaly	Diversion
Time	Latitude	Longitude	Speed	Alt.	Dist. left	Dist. gained	Diff. speed	Diff. alt.	detection	prediction
07:51:02	48.3635	11.7777	176	1700						
07:55:22	48.5382	11.4731	332	9630						
07:59:22	48.9083	11.2436	388	18680	0.0546	$2.9743 \cdot 10^{-7}$	$1.8054 \cdot 10^{-6}$	$7.4204 \cdot 10^{-6}$		
08:03:22	49.3153	10.9882	416	25430	0.0828	$3.2675 \cdot 10^{-7}$	$8.0839 \cdot 10^{-7}$	$3.5521 \cdot 10^{-6}$		
08:07:22	49.6306	10.4288	456	29330	0.1331	$5.8422 \cdot 10^{-7}$	$1.0647 \cdot 10^{-6}$	$1.6532 \cdot 10^{-6}$		
08:12:03	49.9571	9.7260	420	35000	0.1944	$7.1190 \cdot 10^{-7}$	$-9.5439 \cdot 10^{-7}$	$2.0469 \cdot 10^{-6}$		
08:16:03	50.0717	8.9818	464	36000	0.2550	$7.0269 \cdot 10^{-7}$	$1.1553 \cdot 10^{-6}$	$3.2693 \cdot 10^{-7}$		
08:20:03	50.2063	8.2271	444	36000	0.3164	$7.1259 \cdot 10^{-7}$	$-5.1129 \cdot 10^{-7}$	0.0		
08:24:03	50.3826	7.5117	452	36000	0.3749	$6.7957 \cdot 10^{-7}$	$2.0725 \cdot 10^{-7}$	0.0		
08:28:03	50.5606	6.8086	432	36000	0.4321	$6.6406 \cdot 10^{-7}$	$-5.2517 \cdot 10^{-7}$	0.0		
08:32:03	50.7191	6.1176	452	36000	0.4878	$6.4605 \cdot 10^{-7}$	$5.2517 \cdot 10^{-7}$	0.0		
08:36:03	50.3427	5.9824	436	36000	0.4943	$7.5422 \cdot 10^{-8}$	$-4.1824 \cdot 10^{-7}$	0.0		
08:40:03	50.1778	6.7147	444	36000	0.4349	$-6.8914 \cdot 10^{-7}$	$2.1102 \cdot 10^{-7}$	0.0	✓	
08:44:03	49.9892	7.4146	440	35980	0.3774	$-6.6744 \cdot 10^{-7}$	$-1.0503 \cdot 10^{-7}$	$-6.4497 \cdot 10^{-9}$	✓	
08:48:03	49.7942	8.1168	448	36050	0.3192	$-6.7540 \cdot 10^{-7}$	$2.0912 \cdot 10^{-7}$	$2.2558 \cdot 10^{-8}$	✓	✓
08:52:03	49.5972	8.8057	444	35950	0.2616	$-6.6842 \cdot 10^{-7}$	$-1.0409 \cdot 10^{-7}$	$-3.2239 \cdot 10^{-8}$	✓	✓
08:56:03	49.3920	9.5026	424	30150	0.2028	$-6.8226 \cdot 10^{-7}$	$-5.3485 \cdot 10^{-7}$	$-2.0368 \cdot 10^{-6}$	✓	✓
09:00:03	49.1668	10.0852	408	25000	0.1524	$-5.8495 \cdot 10^{-7}$	$-4.4639 \cdot 10^{-7}$	$-2.1676 \cdot 10^{-6}$	✓	✓
09:04:03	48.8666	10.6016	392	21480	0.1056	$-5.4328 \cdot 10^{-7}$	$-4.6425 \cdot 10^{-7}$	$-1.7579 \cdot 10^{-6}$	✓	✓
09:08:13	48.5894	11.0697	352	13850	0.0627	$-4.9831 \cdot 10^{-7}$	$-1.2481 \cdot 10^{-6}$	$-5.0136 \cdot 10^{-6}$	✓	✓
09:12:25	48.4356	11.6073	324	11000	0.0161	$-5.4085 \cdot 10^{-7}$	$-9.6160 \cdot 10^{-7}$	$-2.6625 \cdot 10^{-6}$	✓	✓
09:16:25	48.4192	12.0722	228	4000	-0.0219	$-4.4273 \cdot 10^{-7}$	$-4.0369 \cdot 10^{-6}$	$-1.0832 \cdot 10^{-5}$	✓	✓

Table B.6: Analysis of the diverted flight from Munich to Heathrow.
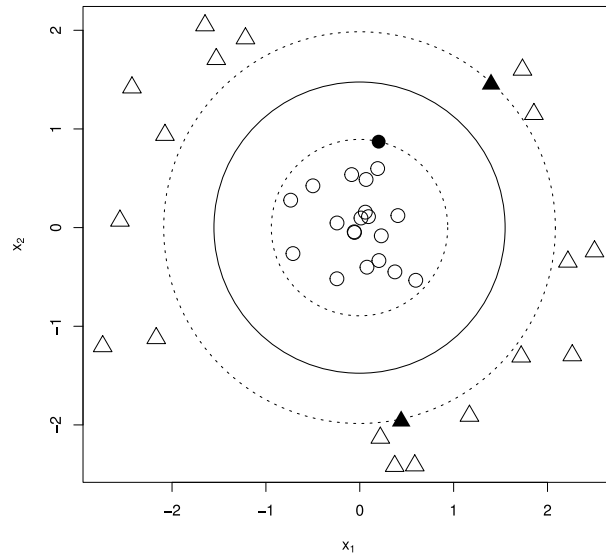
(a) Input space
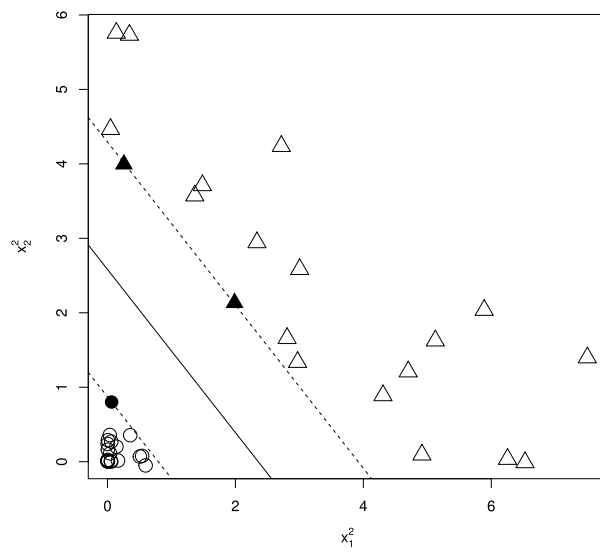
(b) Feature space

Figure A.9: An SVM with a non-linear decision boundary, with data and decision hyperplane plotted in the feature space (A.9a) and in the quadratic kernel space (A.9b)

38

(a) $\gamma = 2^{-3}$

(b) $\gamma = 1$

(c) $\gamma = 2^3$
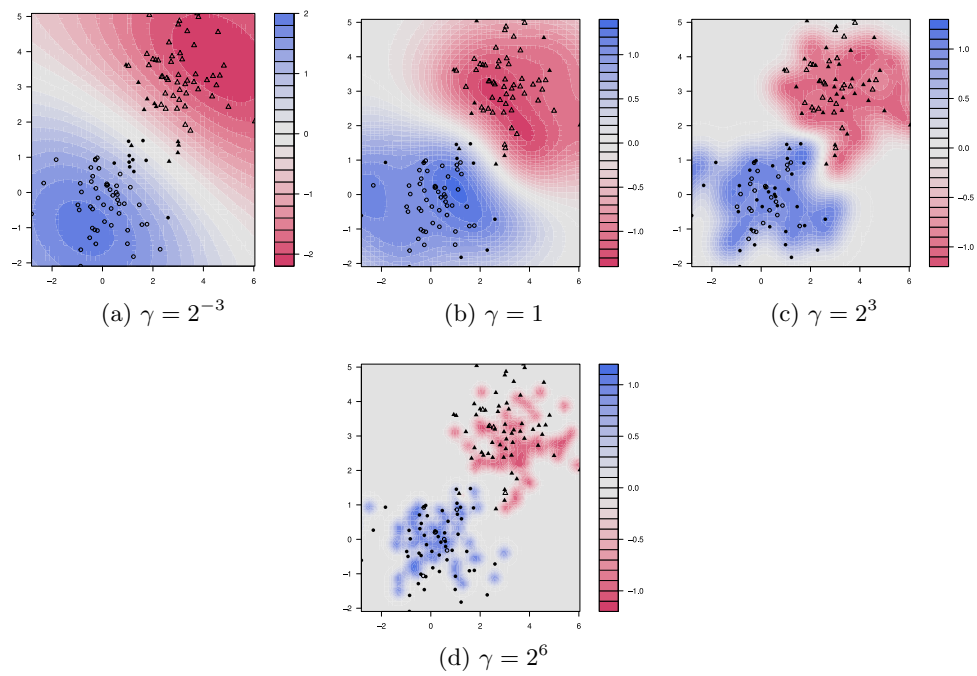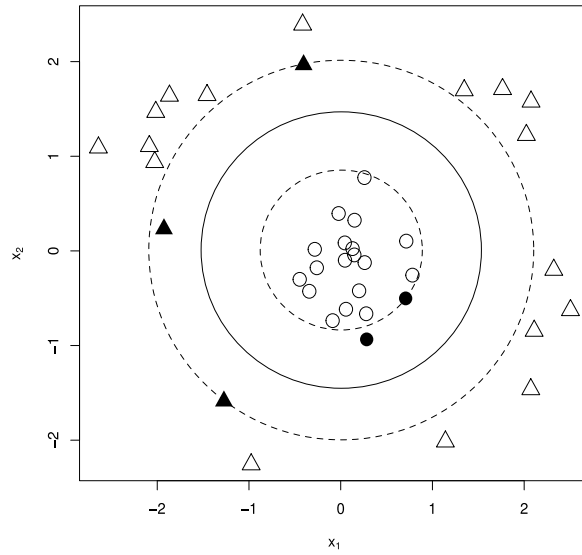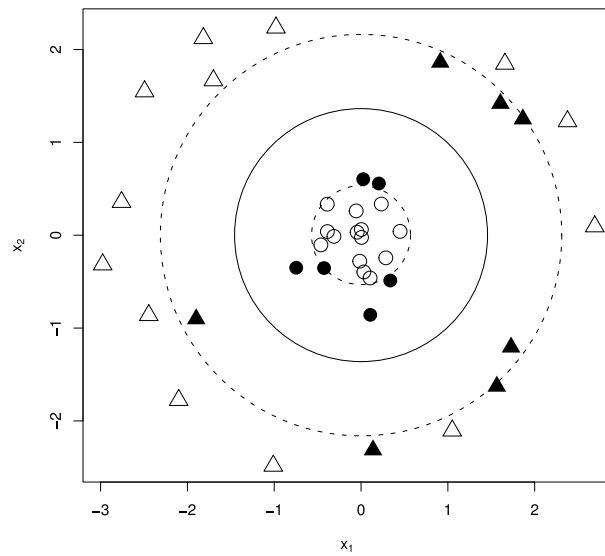
(d) $\gamma = 2^6$

Figure A.10: The effect of parameter γ on the decision hyperplane of an SVM with Gaussian kernel

(a) $\nu = 0.1$

(b) $\nu = 0.3$

Figure A.11: Dealing with outliers using soft-margin SVMs and parameter ν

40

Event attributes					Features				Anomaly	Diversion
Time	Latitude	Longitude	Speed	Alt.	Dist. left	Dist. gained	Diff. speed	Diff. alt.	detection	prediction
16:26:19	36.8639	30.8093	180	1700						
16:30:19	37.0355	30.9193	292	9680						
16:34:49	37.4789	30.9488	404	18000	-0.0075	$3.4034 \cdot 10^{-08}$	$3.7366 \cdot 10^{-06}$	$6.9796 \cdot 10^{-06}$	✓	
16:38:49	37.9577	30.9818	448	24200	-0.0043	$3.6646 \cdot 10^{-08}$	$1.1987 \cdot 10^{-06}$	$3.4103 \cdot 10^{-06}$	✓	
16:43:44	38.5967	30.9638	448	29600	0.0063	$1.2334 \cdot 10^{-07}$	0.0	$2.3313 \cdot 10^{-06}$		
16:47:49	39.0721	30.7334	452	32900	0.0359	$3.4362 \cdot 10^{-07}$	$1.0317 \cdot 10^{-07}$	$1.2256 \cdot 10^{-06}$		
16:51:49	39.5435	30.5022	468	35250	0.0653	$3.4107 \cdot 10^{-07}$	$4.0369 \cdot 10^{-07}$	$8.0043 \cdot 10^{-07}$		
16:55:49	40.0424	30.2528	452	36000	0.0966	$3.6322 \cdot 10^{-07}$	$-4.0369 \cdot 10^{-07}$	$2.4434 \cdot 10^{-07}$		
16:59:49	40.509	30.0158	452	36000	0.1260	$3.4090 \cdot 10^{-07}$	0.0	0.0		
17:03:49	40.9961	29.7648	468	36000	0.1567	$3.5665 \cdot 10^{-07}$	$4.0369 \cdot 10^{-07}$	0.0		
17:07:49	41.4784	29.5121	472	36000	0.1872	$3.5439 \cdot 10^{-07}$	$9.8777 \cdot 10^{-08}$	0.0		
...	...	...	...	...	...	...	...	...		
19:12:15	56.3606	22.2014	452	36000	0.9199	$2.6492 \cdot 10^{-07}$	$1.0317 \cdot 10^{-07}$	0.0		
19:16:15	56.8248	21.8382	456	36000	0.9464	$3.0812 \cdot 10^{-07}$	$1.0225 \cdot 10^{-07}$	0.0		
19:20:15	57.2664	21.4743	444	36000	0.9724	$3.0117 \cdot 10^{-07}$	$-3.0950 \cdot 10^{-07}$	0.0		
19:24:15	57.7369	21.0757	456	36000	0.9999	$3.1993 \cdot 10^{-07}$	$3.0950 \cdot 10^{-07}$	0.0	✓	
19:28:15	58.1878	20.6635	456	36000	0.9720	$-3.2444 \cdot 10^{-07}$	0.0	0.0	✓	
19:32:15	58.5864	20.0672	456	36000	0.9323	$-4.6043 \cdot 10^{-07}$	0.0	0.0	✓	✓
19:36:15	58.9727	19.4734	448	36000	0.8937	$-4.4832 \cdot 10^{-07}$	$-2.0542 \cdot 10^{-07}$	0.0	✓	✓
19:40:15	59.3615	18.8578	452	36000	0.8545	$-4.5446 \cdot 10^{-07}$	$1.0316 \cdot 10^{-07}$	0.0	✓	✓
19:44:15	59.7473	18.2276	444	36000	0.8153	$-4.5478 \cdot 10^{-07}$	$-2.0725 \cdot 10^{-07}$	0.0	✓	✓
19:48:15	60.1261	17.5894	448	36000	0.7765	$-4.5019 \cdot 10^{-07}$	$1.0409 \cdot 10^{-07}$	0.0	✓	✓
19:52:15	60.5042	16.9319	448	36000	0.7375	$-4.5334 \cdot 10^{-07}$	0.0	0.0	✓	✓
19:56:15	60.8755	16.2662	452	36000	0.6988	$-4.4855 \cdot 10^{-07}$	$1.0316 \cdot 10^{-07}$	0.0	✓	✓
20:00:47	61.2186	15.6306	444	36000	0.6627	$-4.1913 \cdot 10^{-07}$	$-2.0733 \cdot 10^{-07}$	0.0	✓	✓
20:04:47	61.6485	14.8065	452	36000	0.6170	$-5.3020 \cdot 10^{-07}$	$2.0725 \cdot 10^{-07}$	0.0	✓	✓
20:08:47	62.021	14.0649	444	32450	0.5770	$-4.6519 \cdot 10^{-07}$	$-2.0725 \cdot 10^{-07}$	$-1.2038 \cdot 10^{-06}$	✓	✓
20:12:47	62.3755	13.3361	456	26630	0.5385	$-4.4649 \cdot 10^{-07}$	$3.0950 \cdot 10^{-07}$	$-2.2866 \cdot 10^{-06}$	✓	✓
20:16:47	62.697	12.6515	396	22450	0.5031	$-4.1045 \cdot 10^{-07}$	$-1.6346 \cdot 10^{-06}$	$-1.9769 \cdot 10^{-06}$	✓	✓
20:20:47	62.9955	12.0668	372	14080	0.4741	$-3.3722 \cdot 10^{-07}$	$-7.2539 \cdot 10^{-07}$	$-5.3186 \cdot 10^{-06}$	✓	✓
20:24:52	63.3423	11.7122	264	7800	0.4599	$-1.6432 \cdot 10^{-07}$	$-3.9419 \cdot 10^{-06}$	$-6.6628 \cdot 10^{-06}$	✓	✓
20:28:52	63.4562	11.3417	172	3900	0.4408	$-2.2170 \cdot 10^{-07}$	$-4.8980 \cdot 10^{-06}$	$-7.7375 \cdot 10^{-06}$	✓	✓

Table B.7: Analysis of the diverted flight from Antalya to Skellefteå.