Data Min Knowl Disc (2012) 24:443–471
DOI 10.1007/s10618-011-0240-z

Greener aviation with virtual sensors: a case study

Ashok N. Srivastava

Received: 4 June 2010 / Accepted: 21 September 2011 / Published online: 28 October 2011

Abstract The environmental impact of aviation is enormous given the fact that in the US alone there are nearly 6 million flights per year of commercial aircraft. This situation has driven numerous policy and procedural measures to help develop environmentally friendly technologies which are safe and affordable and reduce the environmental impact of aviation. However, many of these technologies require significant initial investment in newer aircraft fleets and modifications to existing regulations which are both long and costly enterprises. We propose to use an anomaly detection method based on Virtual Sensors to help detect overconsumption of fuel in aircraft which relies only on the data recorded during flight of most existing commercial aircraft, thus significantly reducing the cost and complexity of implementing this method. The Virtual Sensors developed here are ensemble-learning regression models for detecting the overconsumption of fuel based on instantaneous measurements of the aircraft state. This approach requires no additional information about standard operating procedures or other encoded domain knowledge. We present experimental results on three data sets and compare five different Virtual Sensors algorithms. The first two data sets are publicly available and consist of a simulated data set from a flight simulator and a real-world turbine disk. We show the ability to detect anomalies with high accuracy on these data sets. These sets contain seeded faults, meaning that they have been deliberately injected into the system. The second data set is from real-world fleet of 84 jet aircraft where we show the ability to detect fuel overconsumption which can have a significant environmental and economic impact. To the best of our knowledge, this is the first study of its kind in the aviation domain.

Responsible editor: Katharina Morik, Kanishka Bhaduri and Hillol Kargupta.

A. N. Srivastava (✉)
Intelligent Systems Division, Intelligent Data Understanding Group, NASA Ames Research Center, Moffett Field, CA 94035, USA
e-mail: ashok.n.srivastava@nasa.gov

Keywords Ensemble learning · Gaussian process · Aviation · Environmental systems · Anomaly detection

1 Introduction

Although modern aircraft are more fuel efficient than ever before, they use a significant amount of fossil fuels which account for a large percentage of the total operating costs of a commercial airline. Some reports indicate that these costs are as high as 30% of the total expenditures for a major airline. The overall 'carbon footprint' of the aviation system is substantial, although it is overshadowed by other transportation and energy production technologies in terms of its impact on the environment. However, an improvement in the efficiency of these aircraft in terms of fuel usage can have a significant environmental and economic benefit to many parties, including the airline operators, airframe and engine manufacturers, and the public at large. Figure 1 shows the fuel consumption of different aircraft over a fifty year period. Expressed as a percent reduction of the fuel consumption of the Comet 4, we can see a dramatic reduction in engine fuel consumption as well as the aircraft fuel burn per seat. The latter metric takes the aircraft's passenger capacity into account. For example, the Boeing 777–200 consumes about 40% less fuel than the Comet 4 with about 70% less fuel consumed per seat.

Because fuel consumption represents a significant operating cost for an airline, it is monitored and controlled very carefully through a number of procedures. These procedures essentially compare the total fuel consumed on a particular leg of a trip against other trips which have the same origin, destination, take-off weight, make-and-model of aircraft (and engine), flight time and durations of various phases of flight (such as take-off, cruise, descent, and landing), and other contextual factors. If a particular aircraft uses more fuel than expected, it may be brought in for maintenance purposes and further investigation. As we will see later in the paper, there can be a wide degree of variability in the overall fuel consumption for a particular aircraft on a given trip even after taking these factors into account. This variability poses a challenge and results in a classic signal-to-noise issue: is the observed high fuel usage on a specific flight significant or not given the observed data?

1.1 Environmental impact of aviation

Jet engines can be a significant source of carbon emissions. A typical Boeing 747 can carry nearly 184,000 liters of fuel on a given flight. With a conservative assumption that 90% of the fuel is converted into carbon dioxide for a long range flight, and assuming a fuel density of 2.76 kg/l of CO_2, we can estimate that a single Boeing 747 can emit as much as 457,000 kg of carbon dioxide into the high atmosphere. It is clear that when such emissions are multiplied by the number of aircraft in world-wide operation (estimated to be around 15,000 aircraft according to the IPCC, Penner et al. 1999) the environmental impact is substantial. The Intergovernmental Panel on Climate Change Special Report on Aviation and the Global Atmosphere (Penner et al. 1999) describes the impact of these emissions at high altitude in their report, a synopsis of which is

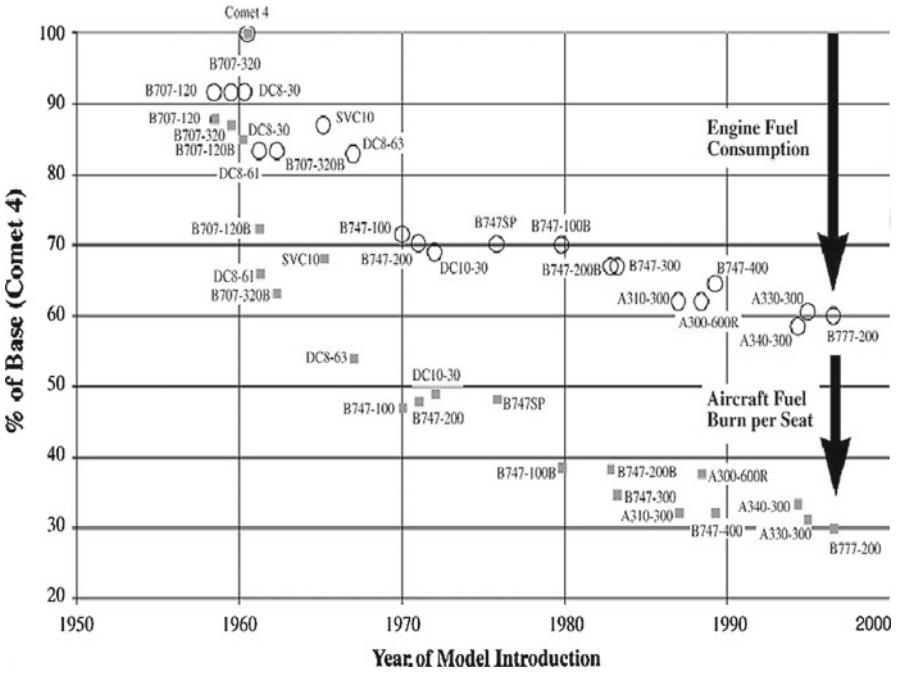

Fig. 1 This figure is part of the IPCC report on Aviation and the Global Atmosphere. It shows a significant reduction in fuel consumption and Aircraft Fuel Burn per Seat as a function of time. This reflects the continuous improvement in efficiency of both the aerodynamic and the engine designs. The capability for increased stage lengths and operations at higher altitudes could also have been significant factors. Notice that the most efficient aircraft in the recent years are the wide-bodies that are used primarily for international flights and long periods of cruise at high altitude. The data are normalized with the Comet 4 in 1960 being the base of comparison (Penner et al. 1999)

quoted here: "Aircraft emit gases and particles directly into the upper troposphere and lower stratosphere where they have an impact on atmospheric composition. These gases and particles alter the concentration of atmospheric greenhouse gases, including carbon dioxide (CO_2), ozone (O_3), and methane (CH_4); trigger formation of condensation trails (contrails); and may increase cirrus cloudiness-all of which contribute to climate change." A study by the National Research Council (Committee on Aeronautics Research and Technology for Environmental Compatibility 2002) indicates "although aircraft fuel consumption is small relative to fuel consumption by other sectors, aircraft emissions are of increasing concern because they are deposited at altitudes where, with the exception of CO_2, they affect the environment differently than ground-based emissions."

1.2 Contribution

In this paper we present a novel method to detect an anomaly in any system which has time-series measurements of some continuous variable of interest and measurements of related discrete and continuous variables. This general framework is applied to the

problem of detecting anomalies such as overconsumption of fuel (also known as an overage) in modern aircraft using data that is already measured and monitored on such aircraft. This data, known as Flight Operational Quality Assurance (FOQA) data is used for numerous purposes including improving safety and efficiency of commercial and business transport aircraft. The approach we discuss here differs significantly from the traditional approach of comparing actual fuel consumption against averages for a given flight or aircraft. Such computations do not sufficiently control for the context of the flight and may not reveal more subtle performance issues. The potential impact of this work for the development of environmentally friendly technologies is significant, because we show a methodology for detecting overages in energy consumption that could be applied in a variety of application domains, including construction, automotive, locomotive, and computing platforms.

The key question that we face is how to develop models of typical fuel consumption given data from a set of flights. Our goal is to provide a cost-effective, scalable, and accurate method to discover fuel overconsumption to help detect aircraft with potential issues in fuel consumption. These issues could point to maintenance or piloting issues, and if left unresolved, could lead to a safety issue. For example, if the control surfaces on an aircraft are in an improper configuration for cruise it could lead to higher drag which could lead to an increase in fuel consumption. Aircraft rigging ages with time and can also lead to an increase in fuel burn. Another source of increase in fuel consumption could be due to the way the airplane is flown. It is possible that the combination of control surface configurations, speed, and other engine settings could lead to an increase in instantaneous consumption even though all systems are individually operating in their nominal regime.

We approach this problem using a method based on ensembles of regression models. Regression models are designed to predict the value of one continuous variable given a set of other discrete and continuous variables. These models are used in numerous applications including finance, the social sciences, medicine, engineering systems, and related areas. The predictions are based on a model that is learned from training data that can then be applied to a test data set. Using a regression model, we predict the instantaneous fuel consumption of an aircraft given a vector of continuous and discrete variables that are measured concurrently. [1] This input vector represents the instantaneous state of the aircraft. The intuition behind this model is that the fuel consumption may be a complex function of the state that can be learned from the data given an appropriate regression model. An issue that arises in this procedure is that we assume that the FOQA data used for training the models are from aircraft operating in nominal conditions. However, it could be that some subset of the training data could be from aircraft operating in off-nominal conditions. Throughout this text, we refer to nominal conditions as those that are according to plan or design, in consonance with the language used in aerospace engineering. We address this problem by requiring that we have a large number of flights for the same make and model of aircraft for a given city pair (origin and destination airport). The assumption that we make is that while a subset of the data may have off nominal characteristics, the vast majority would be

[1] Although this paper presents the use of a concurrent state vector, the approach presented here generalizes to state vectors formed with information from the current time as well as past times.

operating in a nominal condition. We believe this assumption is valid because of the high degree of emphasis on tracking fuel consumption in modern fleets. Thus, any anomalies detected would be present in a small minority of flights.

A key requirement of our approach is that we need an estimate of the uncertainty in the prediction. This uncertainty estimate is used to provide bounds on the expected fuel consumption at a moment in time. To produce these estimates we build an ensemble of regression models. Each regression model is built off of a bootstrap sample of the training data. The mean of these regression models is used as the prediction of the fuel consumption, whereas the standard deviation of the predicted fuel consumptions is used as a measure of uncertainty. The overall approach described here is called *Virtual Sensors*, because we are developing an estimator of one sensor measurement (fuel consumption) given other potentially nonlinearly correlated sensor measurements. This approach is an example of an ensemble learning method where multiple models are built on bootstrap samples to improve prediction accuracy (Breiman 1996).

We contrast this approach with simply monitoring fuel burn and computing the total fuel burned during a flight. This approach leads to a measure of total fuel burned but does not address the issue of whether, at a given moment during flight, the instantaneous fuel consumption is higher than expected.

Key contributions of this research is given below. Earlier work describing Virtual Sensors and their applications to other domains can be found in Srivastava et al. (2005), Way and Srivastava (2006), and Srivastava and Das (2009). In this article, we use Virtual Sensors to demonstrate the detection of overconsumption of fuel and also show that they can detect anomalies in other domains related to aviation systems through the use of two real-world data sets. The paper has the following contributions:

1. A description of a novel application of ensemble regression techniques to detecting overconsumption of energy in real-world aircraft based on instantaneous flight data. To the best of our knowledge this is the first attempt to model instantaneous energy consumption using data already available on aircraft. Indeed, this is a sharp contrast to the current approach which simply measures the amount of fuel used on a particular flight and compares that with past flights of the same make and model of aircraft. Our approach does not require extra data, new equipment, or costly changes to regulation or manufacturing standards.
2. A demonstration, using the Virtual Sensors technique, that it is possible that instantaneous fuel consumption can exceed a preset threshold while the total fuel consumed on a particular flight may be well within expected bounds. While this result may not be surprising from a statistical standpoint, it has a significant impact on the aviation domain and its associated environmental impact. Such a result indicates that although a particular flight falls within the bounds of expected fuel consumption, it could be possible to further reduce the fuel consumption, thus increasing efficiency and decreasing the environmental impact.
3. A case study on a real-world data set from a modern commercial fleet of 84 aircraft covering about 40,000 flights to demonstrate the effectiveness of our approach. Because maintenance data on these aircraft cannot be obtained due to data access restrictions, we also demonstrate our technique on the same real-world data set with artificially injected increases in fuel consumption in one aircraft.

The ensemble regression techniques that we employ build on a substantial body of research that shows that these methods can give highly accurate predictions but also can give a useful measure of certainty. The approach has been previously validated on an anomaly detection problem regarding the Space Shuttle Main Engine (Matthews and Srivastava 2010). We also provide results on two publicly available data sets.[2] The first data set, from the NASA FLTz (Oza 2010) simulator, consists of simulated data of randomly generated circular flight paths starting from San Francisco International Airport end ending at the same location. We show the results of this algorithm in predicting subtle injected anomalies in the roll acceleration given other information about the state of the aircraft. The second publicly available data set is data from a real-world turbine disk (Abdul-Aziz et al. 2010). The data are taken from a capacitive probe sensor technology that measures variations in the distance between two conductive surfaces on the turbine disk. A small crack is induced in the disk which gives rise to an imbalance measurable at high speeds. These two data sets are used to demonstrate the versatility of Virtual Sensors to detect anomalies. Our technique reliably and accurately detects these anomalies.

2 Related work

Our work is based upon an extensive literature regarding regression models and ensemble learning and studies that show the application of these techniques to real-world physical systems. We apply these techniques to build Virtual Sensors for the purpose of anomaly detection. There is a large body of research on anomaly detection methods for continuous and discrete variables which we also reference. Furthermore there is a significant amount of work in model-based approaches to state estimation in the Control Systems literature. Although the Control Systems literature discusses model-based approaches to state estimation and fault detection, in this work we assume that the model is unknown, thus leading us to a data-driven approach. We are not aware of any work directly comparable to the work presented here in the aviation community for the detection of fuel consumption related anomalies. We conclude this section with a discussion of the methods currently employed to detect fuel overconsumption.

We begin with a survey of Virtual Sensors which are nonlinear regression models applied to the specific problem of estimating the value of one sensor measurement given a set of other, possibly nonlinearly correlated measurements (Srivastava et al. 2005). This approach has been successfully employed in a variety of domains including astrophysics (Way and Srivastava 2006) and detecting anomalies in the Main Propulsion System of the Space Shuttle (Matthews and Srivastava 2010). We note that in the latter study, we were able to detect problems in the fuel lines of the Space Shuttle using real data and a method similar to the one described in this paper. In the case of the Space Shuttle, the critical issue was not overconsumption of fuel– it was the fact that there was a crack in the fuel line that could lead to a catastrophic failure. This issue,

[2] These data sets, sample code, and papers are available on our website at https://c3.ndc.nasa.gov/dashlink/resources/.

however, lead to an anomalous pressurization in the fuel line which can be detected using Virtual Sensors.

Subsequent studies have employed Virtual Sensors for estimating sensor values appropriate to vehicle tracking (Petrovskaya and Thrun 2009). In this context, sensor representations refer to the estimated values of certain sensor measurements using statistical regression techniques. There are numerous approaches to address linear and nonlinear regression problems. In this paper, we focus on Gaussian Process Regression, Bagged Neural Networks, and a regularized linear method known as Elastic Nets. A key issue that arises in applying kernel-based models such as Gaussian Processes is that the scalability of the regression method becomes critical. Collobert and Bengio (2001), Chandola and Vatsavai (2010), and Foster et al. (2009) provide directions to increase the scalability of kernel-based techniques for regression. Although kernel methods can be preferable for some regression problems, there are numerous other ways to perform regression which are surveyed well elsewhere (Seber and Wild 1989).

Although it is not necessary to build an ensemble of regression functions for Virtual Sensors, we have found it advantageous to do so to improve prediction accuracy and model stability across different data samples. In this work we perform bagging, or bootstrap-aggregation, with our models (Breiman 1996). This allows us to measure the confidence in the predictions based on the variation of the predictions of the ensemble.

Once a Virtual Sensor is estimated for anomaly detection, one measures the estimation error between the actual sensor value and the predicted sensor value from the Virtual Sensor. If the estimation error is high, it can indicate a change in the underlying system dynamics. This is the standard approach taken in the Control Systems community and can be directly transfered to this formulation. This approach of estimating a Virtual Sensor is a supervised learning approach and the anomaly detection is a direct result of inspection of the residuals. It is possible to perform unsupervised learning to discover anomalies. In fact, for many applications this is the method of choice. An excellent compendium on the subject can be found at Chandola et al. (2009). These approaches often build internal models of nominal (or normal) behavior and then compare the observed behavior against the internal model. An anomaly is detected if the deviation is larger than a pre-specified value.

The problem of detecting increases in fuel consumption on jet engines is not new. Some early reports on this problem domain written by engine manufacturers such as GE Wulf (1980) and Sallee (1980) describe an approach of studying fuel consumption on aircraft that is still used today. The idea is to take snapshots of the fuel consumption during the cruise phase of the flight and compare it with baseline performance. These studies reveal that jet engines can experience a reduction in efficiency of about 2% over 4000 hours of operation due to engine degradation and wear. In the GE report (Wulf 1980) a snapshot of performance during cruise shows a linear increase in fuel consumption with flight cycles as well as a corresponding linear increase in engine exhaust gas temperature for several different engine models. With appropriate condition-based maintenance procedures they indicate in their report that in 1980, "this represents a potential reduction in fuel consumption of 26 million gallons and savings to the airlines of 16.6 million dollars based on projected flight hours." Similarly, the Pratt and Whitney report indicates that the engine deterioration can be as high as 3.8% over 2000 engine cycles with a corresponding increase in engine exhaust temperature

from 5 to 33 degrees Celsius. The combination of higher engine exhaust temperatures and higher fuel consumption can be indicative of engine degradation.

While it is clear that monitoring engine degradation over time can lead to an improved maintenance cycle, thereby reducing the operational costs and reducing the environmental impact, such studies are performed over very long periods of time with data captured at very low frequency. Many modern commercial aircraft are equipped with flight data recorders which capture data at 1 Hz from potentially hundreds of sources. These data comprise the so-called Flight Operational Quality Assurance (FOQA) programs at major carriers and form the basis of the studies conducted in this paper.

3 Overview of the approach

3.1 System architecture

Figure 2 shows the system architecture that we propose for anomaly detection from heterogeneous data sources as part of an overall approach to improving the safety and efficiency of a complex system such as an aircraft (Statler 2007). Although the primary goal of this work is to show the use of Virtual Sensors in discovering fuel overconsumption, it also has an eventual safety benefit. In some cases the source of the overage may be due to a maintenance issue that could lead to a safety problem. The architecture shows the flow of data from multiple, heterogeneous sources through a comparison with performance standards. These performance standards could be drawn from a standard operating procedure (SOP) or from a learning model such as Virtual Sensors. The identification of potential hazards (of which a fuel overage may be a subset) can be made either autonomously or with the aid of a human in the Evaluation phase.

In some cases, an intervention strategy may need to be developed. For example, in the case of a fuel overage, it may be that an engine needs to be exchanged on a particular aircraft; in other scenarios, the source of the overage may be due to improper adjustment of the flight control surfaces. These considerations would be made by safety and maintenance experts perhaps aided by models and simulations. The resulting intervention can be deployed in the field as determined by the experts as shown in the Intervention phase. Virtual sensors would be one approach to comparing observed data against performance standards. They may also be used in the Evaluation phase, where we perform statistical analysis to determine the frequency and severity of an event such as a fuel overage.

3.2 Notation

This section outlines the notation used for the remainder of this paper. The bold-faced variables correspond to vector quantities, whereas the non-bold correspond to scalar quantities. For a Virtual Sensors model, we assume that the inputs and outputs are observed quantities and that the system state is also observable.

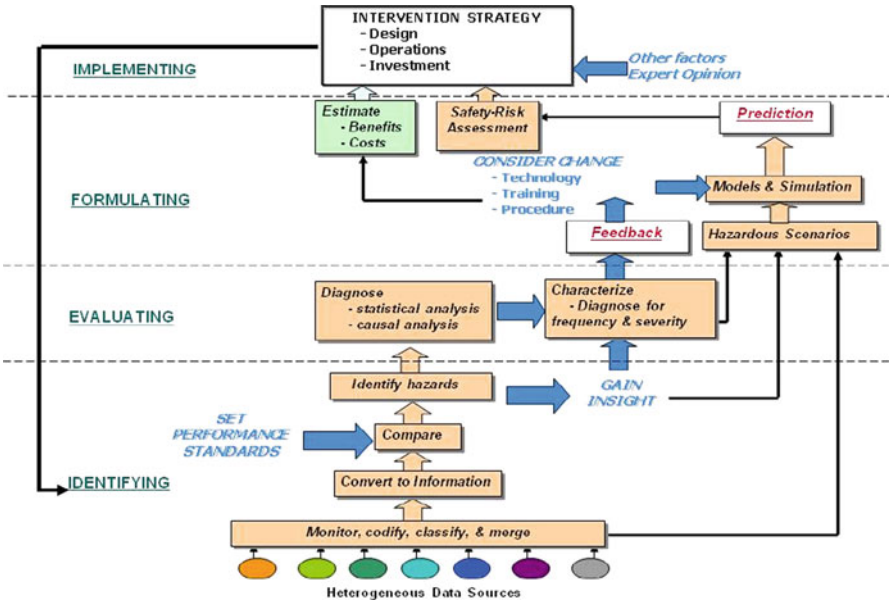

Fig. 2 This diagram shows a conceptual architecture for the use of Virtual Sensors and other anomaly detection methods in the context of improved maintenance with the end goal of improving aircraft safety. Although methods to reduce fuel consumption can have direct environmental and economic impact, it can also have potential public safety benefits (Statler 2007). Virtual sensors can play multiple roles in this conceptual architecture, both at the identification phase and the evaluation phase

- Γ corresponds to a function that maps the previous hidden state to the next hidden state.
- $\mathbf{h}_t$ corresponds to the value of a hidden state at time t. The hidden state can represent different modes of the data generating process.
- Ψ corresponds to a potentially nonlinear function that gives the current observed state of the data generating process given the past observed state, the past hidden variable, and the current inputs to the system.
- $\mathbf{x}_t$ gives the N-dimensional current observed state of the system.
- $\mathbf{u}_t$ gives the current input to the system.
- Ω maps the current observed state to the observed output of the system. In this work we assume that this function is the identity map.
- ε_t represents measurement noise.
- y_t represents the observed output of the system. For the purposes of this study it refers to the fuel consumed at time t.
- t corresponds to the time index.
- $\mathbf{c}$ represents the gross, time independent factors of the flight such as the city pairs, the make and model of the aircraft, etc.
- $P(Y|\mathbf{c})$ is the distribution of total fuel consumed given the flight context.
- $P(y_t|\mathbf{x}_t, \mathbf{c}, \theta)$ is the distribution of the fuel consumed at time t given the inputs, the flight context, and the model parameters.

– α is a threshold multiplier on the number of standard deviations that the observed value must be greater than the mean prediction of the ensemble of regression models.
– Variables with a star superscript, such as $\mathbf{h}^*_{t-1}$ are variables that contain a preset number of lagged values of the time series.

3.3 Discovering fuel overconsumption: a comparison with current methods

We approach the challenge of discovering overconsumption of fuel, or an overage, as an anomaly detection problem in a multi-dimensional multivariate time series. We detect anomalies in a scalar variable given multivariate data series containing both discrete and continuous channels. We assume that we are given data from a data generating process that can be functionally described by the following equations:

$$\mathbf{h}_t = \Gamma(\mathbf{h}^*_{t-1}) \tag{1}$$
$$\mathbf{x}_t = \Psi(\mathbf{x}^*_{t-1}, \mathbf{h}^*_t, \mathbf{u}_t, \mathbf{c}) \tag{2}$$
$$y_t = \Omega(\mathbf{x}_t) + \varepsilon_t \tag{3}$$

We assume that the function Γ determining the evolution of the hidden system state $\mathbf{h}_t$ is unknown. In our fuel consumption study, we assume that there are two possible values of $\mathbf{h}_t$: either the system is in a nominal state with respect to fuel consumption or it is in an off-nominal (high) state. We also assume that the function Ψ, which governs the evolution of the continuous state vector is also unknown. We assume that the vector $\mathbf{x}$ is an N dimensional observed state vector. The quantity $\mathbf{u}_t$ is the observed system input (such as throttle commands from the pilot), and y_t is the observed fuel consumed at time t. We model the measurement noise as $N(0, \sigma^2)$. We assume that we are given the set $(\mathcal{U}, \mathcal{X}, \mathcal{Y})$ for a fleet of aircraft covering many legs and different operating conditions. In this paper, we do not distinguish between the exogeneous inputs $\mathcal{U}$ and the state variables $\mathcal{X}$. Thus, for the remainder of this paper we lump the input and the state variables together into $\mathcal{X}$ to simplify the notation. The variable $\mathbf{c}$ represents the gross, time-independent, factors of the flight such as the city pairs, the make and model of the aircraft, the pilots and flight crew, etc.

The approach used currently by the aviation community compares the total fuel consumed on a flight Y against a distribution $P(Y|\mathbf{c})$ where $\mathbf{c}$ is the flight context and $Y = \sum_t y_t$. If $Y > E_P(P(Y|\mathbf{c})) + \tau$ where τ is a distribution dependent and time independent threshold (such as a fixed multiple of standard deviations), the aviation expert would indicate that the flight had a significant overage.

Our approach differs from this procedure as follows. Rather than treating the entire flight as a monolithic entity, we take advantage of the fact that at each instant in time (about 1 Hz on the aircraft under study) a data recording system is storing about 84 parameters of information, including information about instantaneous fuel usage in the engines, navigational information, wind speed and direction, altitude, attitude, and other positional information, and various latitudinal and longitudinal accelerations. We attempt to model $P(y_t|\mathbf{x}_t, \mathbf{c}, \theta)$ where θ are the parameters of our model. Our

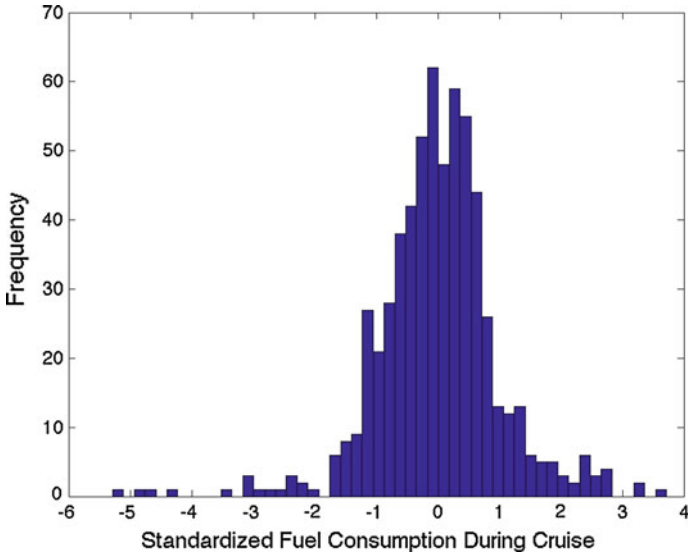

Fig. 3 This figure shows the variability in fuel consumption for a real-world commercial jet aircraft for a given city pair. The data have been standardized to have zero mean and unit variance to maintain the confidentiality of the data

procedure is then to estimate the fuel consumed at time t against $P(y_t|\mathbf{x}_t, \mathbf{c}, \theta)$ and set a flag if:

$$y_t > E_P(P(y_t|\mathbf{x}_t, \mathbf{c}, \theta)) + \alpha\sqrt{Var_P(P(y_t|\mathbf{x}_t, \mathbf{c}, \theta))} \tag{4}$$

where α is a constant threshold multiplier selected based on a desired detection rate and other criteria. If, for a given flight of duration T the total number of times the anomaly flag is activated is large, we indicate that there is a potential anomaly on the flight with respect to fuel consumption. For a two-sided anomaly, we simply modify Eq. 4 so that $|y_t - E_P(P(y_t|\mathbf{x}_t, \mathbf{c}, \theta))|$ must be less than the specified threshold.

Figure 3 shows the variability in fuel consumption in the form of a histogram for one city pair. The data have been standardized to have zero mean and unit variance to give an idea of the degree of total variation in the fuel consumed and also to protect proprietary information. Notice that the unconditional distribution of fuel consumption varies by as much as 4 standard deviations thus making it difficult to determine which flights consumed more fuel than expected for the given operating conditions. This is the aim of our study.

Thus, although there is variability in fuel consumption, we hypothesize that given these contextual factors, there will still be a small minority of aircraft that are consuming more fuel than warranted in a particular time dependent flight context. In order to test this hypothesis, we must model the time-dependent conditional distribution $P(y_t|\mathbf{x}_t, \mathbf{c}, \theta)$ rather than the time-independent conditional distribution $P(Y|\mathbf{c})$ shown in Fig. 3.

The next section discusses Virtual Sensors algorithms. We then discuss the specific requirement of estimating conditional probability distributions and illustrate how the ensemble methods address this requirement. Having established the algorithms we discuss the specific case study of estimating the fuel consumption of aircraft in real-world conditions. We then discuss our experimental results and validate the results using various statistical tests as well as injected anomalies. We conclude the paper by discussing the potential impact this approach can have on improving the efficiency of fleets of aircraft, the subsequent reduction in the carbon footprint by addressing these inefficiencies, and describe plans for future research.

4 Virtual sensor algorithms for anomaly detection

This section begins with a discussion of the Virtual Sensor algorithm and then describes its use with two specific regression functions: the Elastic Net and Gaussian Process regression with a new variant known as stable GP.

4.1 Virtual sensors

Virtual Sensors algorithms are designed to create an estimator for a quantity y_t given other correlated information $\mathbf{x}_t$ and $\mathbf{u}_t$. In the simplest case, this can be accomplished by building a regression model to estimate the desired quantity. In some cases such as the one under consideration in this paper we need to not only estimate y_t but also obtain a local confidence measure on the prediction of y_t. Regions in the input-output space where there is high uncertainty in the prediction need to be characterized differently than regions where there is low uncertainty in the prediction. For example, in this case study, during the climb phase of flight there is a higher uncertainty in the prediction of the fuel consumption than in the cruise phase of the flight. Rather than encoding the phase-of-flight in the model we instead have the algorithm discover these different operating regimes using data alone.

The uncertainty in the estimate can be characterized by $Var_P(P(y_t|\mathbf{x}_t, \theta))$, assuming that we have an efficient method to compute this quantity. In the actual computations we use the standard-deviation of this quantity, but discuss the variance of the estimator to maintain the consistency of the discussion. Thus, rather than building a single regression model, we are building an ensemble using the bootstrap aggregation technique of Breiman (1996) known as bagging. With bagged predictors, we sample the data set m times with replacement and build m models. The mean of these predictions becomes our estimate of $E_P(P(y_t|\mathbf{x}_t, \theta))$ and the standard deviation of the predictions becomes $Var_P(P(y_t|\mathbf{x}_t, \theta))$. We assume that we have chosen a particular flight context and drop the associated c variable from further discussion. These time-dependent quantities form the basis of our anomaly detection approach. Since the residuals are assumed to be Gaussian we choose a detection envelope of three standard deviations about the predicted mean to be the anomaly threshold, although on two of our examples we do an exhaustive study of the accuracy of the model as a function of the number of standard deviations used for the detection envelope.

We contrast the measure $Var_P(P(y_t|\mathbf{x}_t, \theta))$ with a measure solely obtained from the variance of the fuel consumed across multiple flights. The variance of fuel consumed across multiple flights is well known (see Fig. 3) and can easily be measured. In our measure of variance we compute the uncertainty in the estimate of the fuel consumption at a specific point in time during the flight of a single aircraft.

The base regressors used in this study were neural networks (Nabney 2001), regression trees using Matlab [15], Elastic Nets which are generalized linear models with l_1 and l_2 norms (Hastie et al. 2009), and Gaussian Processes (Rasmussen and Williams 2006). We begin by reviewing the likelihood function of the model and then discuss the Gaussian Process formulation and show that this method also models the appropriate conditional distribution. Suppose that we are taking a total of m bootstrap samples from our training data set. For bootstrap sample k, a target, and an input variable, we assume that we have a Gaussian noise distribution in accordance with Eq. 3:

$$P(y_t|\mathbf{x}_t, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_t - \hat{y}_t(\mathbf{x}_t, \theta))^2}{2\pi\sigma^2}\right) \tag{5}$$

where $\hat{y}_t(\mathbf{x}_t, \theta)$ is the prediction of the model given the data input $\mathbf{x}_t$ and the model parameters θ. Note that in this equation, we have suppressed the use of k in this notation. However, we note that the model parameters, the inputs, the target, and the model prediction all depend on the bootstrap sample k. In this formulation, $\hat{y}_t(\mathbf{x}_t, \theta)$ is the estimate of the mean of the distribution for the current bootstrap sample. Thus, as we draw further bootstrap samples the distribution of $\hat{y}_t(\mathbf{x}_t, \theta)$ will begin approximating the joint distribution of the inputs and outputs as discussed in Breiman (1996). For a given bootstrap sample, if we assume independence of the samples we can obtain the standard likelihood function whose negative logarithm leads to the familiar squared-error cost function.

$$\mathcal{L} = \prod_{t=1}^{T} P(y_t|\mathbf{x}_t, \theta) \tag{6}$$

$$= \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - \hat{y}_t(\mathbf{x}_t, \theta))^2}{2\pi\sigma^2}\right) \tag{7}$$

Taking the negative logarithm and computing the average log-likelihood, we obtain the per-time-step cost function:

$$\mathcal{C} = \frac{1}{T} \sum_{t=1}^{T} \left(-\frac{(y - \hat{y}_t(\mathbf{x}_t, \theta))^2}{2\pi\sigma^2}\right) + q \tag{8}$$

where q is a constant term with respect to the optimization parameters. Optimization of this cost function is model-dependent and results in a regression function $g(\mathcal{X}_k, \theta_k)$. In this paper we used neural networks, regression trees, and linear regression models as the regression function (Hastie et al. 2001). Once we obtain the estimates for a given bootstrap sample, the procedure is repeated until m models have been built.

The mean and variance of the predictive distribution $P(y_t|\mathbf{x}_t, \theta)$ is formed by taking the mean and variance of the ensemble of predictions from the m models:

$$E_P(P(y_t|\mathbf{x}_t, \theta)) = \frac{1}{m}\sum_{k=1}^{m} g(\mathbf{x}_t, \theta_k) \tag{9}$$

$$Var_P(P(y_t|\mathbf{x}_t, \theta)) = \frac{1}{m}\sum_{k=1}^{m} (g(\mathbf{x}_t, \theta_k) - E_P(P(y_t|\mathbf{x}_t, \theta)))^2 \tag{10}$$

Finally, we perform anomaly detection by computing the fraction of time per flight where the observed fuel flow y_t is greater than $\alpha\sqrt{Var_P}$. In this manner, each flight is given a score, and the flights are sorted so that the flights with the highest fraction of time where the observed fuel flow is higher than the model expects are at the top of the list. Algorithm 1 gives the entire Virtual Sensors for Anomaly Detection Algorithm. We contrast the Virtual Sensors approach with one in which we simply perform anomaly detection on continuous time series by determining when the time series crosses a pre-defined threshold. In that case, the fixed threshold is applied to a single time series without regard to other inputs. While that approach is valid for some applications the Virtual Sensors approach models the dependency between input signals and a pre-specified target and has thresholds that adapt to the underlying uncertainty in the model predictions.

Algorithm 1: Virtual Sensors for Anomaly Detection

Input: $(\mathcal{X}, \mathcal{Y}, \mathcal{C}, \alpha, m, n)$, representing state variables, the target variable, the cost function for minimization, a multiplier on the number of standard deviations to use as the anomaly detection threshold, the number of models, the number of bootstrap samples, respectively.
Output: Sorted list of anomalies $List$
Initialization: Standardize inputs and outputs to have zero mean and unit variance;
begin
  for $k = 1$ *to* m **do**
    Draw bootstrap replicate with n **samples**: $(\mathcal{X}_k, \mathcal{Y}_k)$
    minimize cost function C to obtain estimate: $\hat{\mathcal{Y}}_k = G(\mathcal{X}_k, \theta_k)$;
  Compute mean and standard deviation of the estimates for the m models;
  Compute the percentage of the test data for a given flight that is larger than the mean + α standard deviations;
  Return rank ordered list $List$ of anomalous flights.
end

4.2 Virtual sensors with elastic nets

Since the base regression algorithms for neural networks and regression trees are well-known, we refer the interested reader to Hastie et al. (2009), Friedman et al. (2010), and Friedman et al. (2007) for more information. In this section we explore Elastic Nets, which are a form of linear regression with l_1 and l_2 regularization performed simultaneously (Friedman et al. 2007). We discuss Gaussian Process Regression in

the next section and compare that method with Elastic Nets and the other models discussed in this paper.

Due to the relatively large number of input variables in our model and the fact that there is a high degree of intercorrelation between the input variables we chose to use Elastic Nets as a method for simultaneously performing variable selection and regression.[3] They also offer the advantage of being highly scalable compared with the other methods tested. The Elastic Net is a generalized linear regression model where we assume that $E(y_t|x_t) = \beta_0 + x_t^T\beta$ is the regression function. Assuming that $x \in \mathbb{R}^p$, we perform the following minimization as discussed in Friedman et al. (2007):

$$C = \min_{\beta_0,\beta}\left(\frac{1}{2T}\sum_{t=1}^{T}(y_t - \beta_0 - x_t^T\beta)^2 + \lambda Q_\gamma(\beta)\right) \tag{11}$$

where

$$Q_\gamma(\beta) = (1-\gamma)\frac{1}{2}||\beta||_{l_2}^2 + \gamma||\beta||_{l_1} \tag{12}$$

$$= \sum_{j=1}^{p}(1-\gamma)\frac{1}{2}\beta_j^2 + \gamma|\beta|_j \tag{13}$$

This cost function has two components that can be traded off against each other by setting the parameter γ. When $\gamma = 0$, this results in ridge regression, and $\gamma = 1$ results in lasso regression. Ridge regression can help in situations with high collinearity, and lasso regression can be a form of variable selection. Intermediate values of this parameter allow a tradeoff between these two extremes. Friedman et al. (2007) show that this cost function yields a model that performs variable selection even in situations where the input variables are highly correlated. The authors go on to describe an extremely efficient method of performing the optimization above using a coordinate descent method. The interested reader is referred to their paper for more information. In the study discussed here, we used the Elastic Net code (Hastie et al. 2009) available on their website.

The assumptions behind this model are highly applicable to the study discussed in this paper, and especially for many engineering systems where one would want to employ Virtual Sensors. Engineering systems often have sensors with built-in redundancy as well as an associated control system; as such, one would expect that the sensors and output variables have moderate to high degrees of correlation. In some applications the number of variables may be large based on the complexity of the system. Thus, Elastic Nets provide several unique capabilities which we employ in this study.

[3] We note that the typical situation where this method is used is where p>>T.

4.3 Virtual sensors with gaussian process regression

The fourth approach we explored in this paper is to use Gaussian Process Regression (Rasmussen and Williams 2006) as the base model for Virtual Sensors. GPs predict the mean of the conditional distribution and also provide a principled way of estimating the variance of the conditional distribution. We briefly review the mathematics of the GP and tie it to our previous discussion regarding the conditional distribution of the target given the inputs, following the arguments given in Rasmussen and Williams (2006). Equation 5 gives the likelihood of the target given the inputs and the model parameters. For simplicity, for a given bootstrap sample if we assume that we have a linear model, i.e., $\hat{y}_t(\mathbf{x}_t, \theta) = x_t^T \theta$ and assume that the time samples are independent, we find that the conditional distribution is multivariate Gaussian as shown in the likelihood function in Eq. 7. The key step that occurs in a GP is that we also assume a prior distribution on the weights of the model θ that assumes a Gaussian distribution with zero mean and a covariance matrix Σ_θ. This matrix is of size $p \times p$ and, through the use of Bayes Rule, we obtain the following joint distribution of the weights given the inputs and target:

$$p(\theta|X, y) = N\left(\frac{1}{\sigma^2} A^{-1} X y, A^{-1}\right) \tag{14}$$

where $A = \frac{1}{\sigma^2}\left(X^T\right) + \Sigma_\theta^{-1}$. When a new input vector x_t is given, the predictive distribution can be obtained by averaging over all model parameters weighted by their posterior probability. This gives rise to the following two equations for the predictive distribution for an input $\mathbf{x}_t$ as shown in Rasmussen and Williams (2006):

$$P\left(\hat{y}_t|\mathbf{x}_t, X, y\right) = \int p(\hat{y}_t|\mathbf{x}_t, \theta) p(\theta|X, y) d\theta \tag{15}$$

$$= N\left(\frac{1}{\sigma^2} \mathbf{x}_t A^{-1} X y, \mathbf{x}_t A^{-1} \mathbf{x}\right) \tag{16}$$

This results in a predictive distribution that is no longer a function of the model parameters, which is, in a sense, the best estimate of the distribution assuming the Gaussian prior on the model weights. Rasmussen shows that this process can be generalized to nonlinear models in the input space through the use of the kernel trick. We note in passing that Gaussian Process Regression requires the inversion of a large matrix which can lead to computational and numerical stability problems. Recent results from Foster et al. (2009), known as stable GP, show a method to perform this computation efficiently with high numerical stability and similar or better performance than the standard method. We test a standard GP implementation and stable GP in this paper as the fifth regression method.

As in the case for the other models, we perform bootstrapping on m models and then average the resulting estimates of the mean of the predictive distribution together. In related work, the estimated variances are added together along with a term that includes the empirical mean squared error as discussed in Chen and Ren (2009). However, to

maintain our ability to compare performance across algorithms, we compute the uncertainty in the estimate for the GP and stable GP algorithms in the same manner as for the other algorithms.

Real-world applications may generate a significant amount of data for Virtual Sensors. For example, we are in the process of obtaining data from an airline with nearly 175,000 flights of about 100 aircraft taken over a 2 year period with approximately 100 recorded parameters taken at 1 Hz. This will amount to nearly 1.5TB of data. Given that major air carriers may have as many as 3,000 *flights a day* the data volumes are growing at an extremely fast pace. It is necessary to analyze this data in a timely fashion if we are to detect and address anomalies as they arise in the system. Some of these anomalies may be regarding safety issues and others may point to issues that have an economic impact. The Virtual Sensors method depends on bagging, which is highly scalable as noted by its inventor Leo Breiman, "…bagging is almost a dream procedure for parallel computing. The construction of a predictor on each $\mathcal{L}^{\{B\}}$ [bootstrap sample] proceeds with no communication necessary from the other CPU's" (Breiman 1996). We also note that the memory requirements for the method, particularly when the Elastic Nets are used, is relatively small. The associated convex optimization problem has only p parameters, and the summation eliminates the requirement for storing the entire data set in memory.

Based on the algorithms described in the previous sections, we now turn our attention to using Virtual Sensors to detect anomalies on two publicly available data sets and implementing Virtual Sensors on real-world data obtained from an airline. [4] The next section discusses the data sets used, their preparation, the experimental set up, and the results.

5 Data

We study Virtual Sensors for anomaly detection on three data sets: one public data set from the FLTz flight simulator, one from a real-world turbine disk, and a proprietary data set. The proprietary data set is from a fleet of 84 jet aircraft from a US-based carrier.

5.1 FLTz simulator data

The data set was obtained by generating 100 flights with varying flight conditions in the FLTz simulator. These lead to different altitudes, Mach numbers, and turbulences. The turbulence levels vary from none to mild and the empirical maximum and minimum were computed and rounded before picking the normalization ranges. Table 1 shows the list of parameters as well as a justification for the range values given which is given in the Comments column. The range values were used for normalization in

[4] We cannot divulge the name of the airline or the data itself due to the agreement between NASA and the carrier. The data are normalized to zero mean and unit variance to help in computations and to protect information that could reveal the data source. We also do not state the city pairs we studied for the same reasons.

some studies. Note that Parameters 1–19 are inputs and 20–25 are outputs. Parameter 26 is the mass used in a varying mass model that was used in related work (Chu et al. 2010). The FLTz simulator does not provide a value for fuel consumption, we show the same approach for detecting anomalies in the roll-acceleration, thereby demonstrating that the Virtual Sensors method is not specific to a single application. We injected a bias in the form of a linear ramp function in the data stream for the roll acceleration that increases from 0 to 0.1 over about 100 flights. This represents a change of 20% of the range of the roll acceleration over 100 flights. The algorithm was set up to predict the roll acceleration given the other input information shown in Table 1 under both no turbulence and high turbulence settings. A training data set was prepared using 100 flights with no injected fault and two test data sets were prepared (one for no turbulence and one for high turbulence) using the injected fault data. The mean and standard deviation of the training data set was used to standardize the training and the test data sets.

5.2 Turbine engine disk spin test data

This data set is taken from a real-world experiment on a spinning metallic engine disk. In the experimental setup, researchers took data under three configurations using a capacitive sensor probe. The first condition represented nominal behavior in which a disk with no known defects was tested at 3000, 4000, and 5000 revolutions per minute (rpm). The capacitive sensors measured the variation in the blade-to-edge clearance on 32 blades (Abdul-Aziz et al. 2010). The algorithm was set up to predict the variation in blade 7 given the variations measured in the remaining 31 blades and the disk speed at 3000, 4000, and 5000 rpm. The nominal data set was used to train the algorithm and it was tested on data sets with a small notch and a large notch deliberately placed in the disk. The mean and standard deviation of the training data set was used to standardize the training and the test data set.

5.3 Real-world aircraft flight and fuel consumption data

The proprietary data set we obtained is from real-world flights of modern passenger aircraft. We obtained 1938 flights of data from a US-based carrier taken during the period from 2004–2005. This data includes over 80 parameters taken at a 1 Hz sampling rate for the duration of the flight. Based on a discussion with two aviation experts, we narrowed the number of variables down to 35. These included variables such as: Altitude, Acceleration Load Factor, Lateral Acceleration, Longitudinal Acceleration, N1 (low compressor speed) and N2 (high compressor speed) for both aircraft engines, Throttle Position for both engines, Airspeed, Pitch Angle, Roll Angle, Vertical Speed (based on inertial measurement), Flap Positions, and Air Temperatures. Although we could have included all 80 parameters in the model, many of them are clearly unrelated and were discarded to simplify the model training. Each flight has a recorded tail number which is an identifier of the plane itself. We do not know when specific tail numbers may have had engine changes or changes in the engine configuration. However, previous studies of Virtual Sensors on the Space Shuttle Main Engine have shown

Table 1 This table shows the inputs and outputs of the FLTz simulator (Oza 2010) along with the units and range of the data values

	Parameter	Range	Units	Comments
1.	Aileron	$[-0.5, 0.0]$	degrees	Mild turbulence data
2.	Differential aileron	$[-5.0, 5.0]$	degrees	Mild turbulence data
3.	Elevator	$[-2.0, 2.0]$	degrees	Mild turbulence data
4.	Rudder	$[-5.0, 5.0]$	degrees	Mild turbulence data
5.	Stabilizer	$[-2.0, -1.0]$	degrees	Mild turbulence data
6.	Roll angle	$[-0.1, 0.1]$	radians	About 5°; more in strong turb.
7.	Yaw angle	$[-3.0, 3.0]$	radians	Relative to earth-axis, between $[-\pi, \pi]$
8.	Pitch angle	$[0.0, 0.05]$	radians	Plane doesn't pitch much in cruise
9.	Angle-of-attack	$[0.0, 0.05]$	radians	Greater than 0; steady in cruise
10.	Sideslip angle	$[-0.02, 0.02]$	radians	About 1°; more in strong turb.
11.	Mach number	$[0.7, 0.9]$	–	Less than 1; range simulation
12.	Dynamic pressure	$[200, 300]$	Pascals	Approx. min, max Mach
13.	Engine thrust	$[20000, 33000]$	lbs	Mild turbulence data
14.	Long. velocity	$[700, 900]$	ft / s	Mild turbulence data
15.	Lat. velocity	$[-10, 10]$	ft / s	Mild turbulence data
16.	Vertical velocity	$[0.0, 40]$	ft / s	Mild turbulence data
17.	Roll rate	$[-0.1, 0.1]$	rad / s	Mild turbulence data
18.	Pitch rate	$[-0.005, 0.005]$	rad / s	Mild turbulence data
19.	Yaw rate	$[-0.01, 0.01]$	rad / s	Mild turbulence data
20.	Forward accel.	$[-0.5, 0.5]$	ft / s^2	Mild turbulence data
21.	Lateral accel.	$[-10, 10]$	ft / s^2	Mild turbulence data
22.	Vertical accel.	$[-10, 10]$	ft / s^2	Mild turbulence data
23.	Roll accel.	$[-0.25, 0.25]$	rad / s^2	Mild turbulence data
24.	Pitch accel.	$[-0.05, 0.05]$	rad / s^2	Mild turbulence data
25.	Yaw accel.	$[-0.05, 0.05]$	rad / s^2	Mild turbulence data
26.	Mass	$[4680, 5800]$	slugs	Simulation setup

The comments give justifications for the range values. Parameters 1–19 are inputs and 20–25 are outputs. Parameter 26 is the mass used in a varying mass model. This table is copied with permission from the authors of Chu et al. (2010)

that the method can reveal faults and also different engine configurations (Matthews and Srivastava 2010). There are 84 tail numbers in our data set for the chosen *city pairs*, which refers to an origin and destination pair of cities. The true airspeed took into account the aircraft heading and the wind speed and direction.

The data were divided into a training set for one city pair $A \rightarrow B$, and then two test sets $B \rightarrow A$ and $B \rightarrow C$. This represents the worst case scenario with respect to the algorithm, since the algorithm is being tested on flight directions and city pairs that do not overlap with the training data. We standardized each training bootstrap replicate to have zero mean and unit variance. For this proof-of-concept study, we chose to develop our models by training data on flights between a city pair (say, A $\rightarrow$ B) and testing on the reverse trip, B $\rightarrow$ A, as well as another trip with the same origination airport

B → C. We chose this experimental design for several reasons. First, we hypothesized that the context of a flight from one city to another, when taken over a year, should constitute a reasonable set of flight events: weather delays, ground issues, air traffic control issues, etc. Since the jet stream plays a critical role in fuel consumption (because the jet stream can 'push' the aircraft forward if its wind direction is in the same direction as the plane's flight path), we also hypothesized that the worst-case scenario would be for us to train on data taken from one direction and then test on the other direction. If the model incorrectly captures the relationship between the wind-speed, its direction, and flight path, that would result in a high root-mean-squared (RMS) error thus invalidating the model. Our results indicate, however, that we are able to obtain low RMS errors on the test set. The RMS errors we observed vary from 0.34 to 0.88 with the standardized test set.

For each flight, we selected the cruise phase of flight as the period for model training and testing. The selection of the cruise phase avoids variability during take-offs and landings that are due to city-specific and, possibly, time specific air traffic patterns. We defined the cruise-phase of flight as the 1 hour and 40 minutes duration from the time after the landing gear are retracted. Because the data are sampled at a high frequency compared with the rate of fuel consumption and airplane dynamics, we decimated the input and output time series by a factor of 10. Some flights were discarded due to bad data or insufficient data due to recording or other errors. In the end, we had about 658 flights for training and about 1280 flights for testing (including both legs for each city pair). [5] Because we do not have data regarding overages in fuel consumption, we chose to inject one tail number which had progressively higher fuel consumption with time. This tail number was generated by taking a tail number from our data set and artificially increasing its fuel consumption linearly by about 5% per flight. The first flight had a fuel consumption of about 90% of the average fuel consumption for the city pair.

During the model building process, the training data were divided into 22 sets of approximately 30 flights each. For each set of 30 flights a 90% sample was drawn three times with replacement. This resulted in 66 models being generated for training. We tested different methods of performing the bootstrap sampling and found that our results did not vary significantly with different configurations. The models used in this study were neural networks (Nabney 2001), regression trees from Matlab, generalized linear models with L_1 norm (Hastie et al. 2009), Gaussian Processes (Rasmussen and Williams 2006) and stable GP (Foster et al. 2009). The means and variances of the output of the regression functions were combined as discussed earlier. The regression functions are set up to predict the combined fuel consumption in both engines as a function of the inputs described above.

6 Experimental results

The discussion of the experimental results is divided into three sections. The first section overviews the results for applying Virtual Sensors to detect anomalies in the roll

[5] We give approximate numbers of flights because some flights were discarded due to bad data.

acceleration of an aircraft using data from a flight simulator. The second section gives the results for applying Virtual Sensors to detect anomalies in a real-world data set from a turbine disk spin test, and the final section gives the results of detecting fuel overconsumption anomalies.

6.1 Results for FLTz simulated data

The FLTz data set has a single training set and two test sets. The training set has no turbulence and no faults injected, whereas the test sets have a fault injected in the roll acceleration the form of a ramp function under both no-turbulence and turbulence settings. The added turbulence increases the difficulty of detecting the fault in the roll acceleration and provides for additional validation of the methods described here.

Figure 4 shows the results of the Virtual Sensors method in detecting faults in the roll acceleration for all five algorithms: the regression tree (tree), Elastic Network (glm), neural network (nnet), Gaussian Process (gp), and stable Gaussian Process (stable GP) regression methods. The curves show the area under the receiver operator characteristic (ROC) curve as a function of the detection threshold multiplier α described in Eq. 4. The left side of the figure shows the results for the case with no turbulence while the right side of the figure shows the results for the high turbulence case. A perfect algorithm would have an area of one for some value of detection threshold multiplier. We can see that all algorithms except for the Elastic Net approach this value in the no-turbulence case. The performance of the Elastic Net as shown in this figure is as expected since this linear model will have low variability across bootstrap samples in low-noise environments such as the one in this case. Thus, its variation in predictions would be smaller than other models, requiring a higher detection threshold to achieve the same area under ROC curve as for other algorithms. The detection area under the ROC curve increases steadily for the Elastic Net as a function of threshold multiplier. For the high turbulence case, the maximum area of the ROC curve is 0.8, which is less than the case with no turbulence as expected. The algorithms perform similarly with respect to each other for the high turbulence and the no turbulence settings. Simon (2010) shows the development of a three-dimensional ROC curve for simultaneously detecting and diagnosing faults. This innovation can be applied in the event that we have additional causal information regarding the faults.

While this example is not directly related to detecting overconsumption of fuel, it is included to demonstrate the effectiveness of the proposed method on a related problem in both no turbulence and high turbulence environments. The results show that the approach is robust to noise and can achieve high detection rates in noisy environments. Further discussion of this data set and other approaches to anomaly detection on this set are given in Chu et al. (2010).

6.2 Results for the turbine engine disk spin test

The results for the turbine engine disk spin test data are shown in Fig. 5. As in the case of the FLTz simulator data, we know the times during the test at which the faults are injected, thereby giving us the ability to construct ROC curves. Figure 5 shows

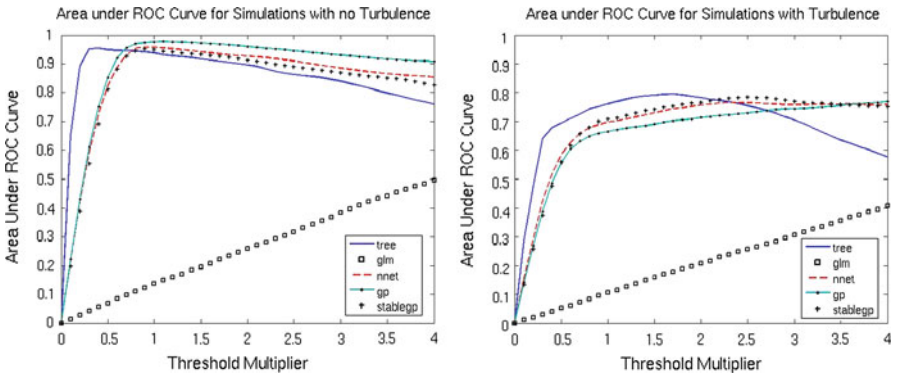

Fig. 4 This figure shows the area under the ROC curve as a function of the threshold multiplier α described in Eq. 4. The left panel shows the area under the ROC curve for the no turbulence setting, whereas the right panel shows the area under the ROC curve for the setting with high turbulence. The figures indicate that most algorithms perform well on this task for both settings. The glm algorithm (Elastic Net) shows performance increasing as a function of threshold multiplier as expected

the area under the ROC curve for the five regression methods as a function of the threshold multiplier α discussed in Eq. 4. The algorithms detect the anomalies in the data very well, with over 90% area under the ROC curve captured by the GP for a large interval of the threshold multiplier. These results show that in real-world noise environments with realistic seeded faults the Virtual Sensors method described here can detect anomalies with high accuracy. Other work (Abdul-Aziz et al. 2010) shows the use of unsupervised anomaly detection methods to detect these anomalies. As in the case of the FLTz simulator, this example is not directly related to overconsumption of fuel but is included to show the versatility of the proposed method on publicly available data. We turn our attention in the next section to the problem of detecting overconsumption of fuel on real-world aircraft.

6.3 Results for detecting overconsumption of fuel consumption in real aircraft

In this section we discuss the results for detecting overconsumption of fuel on real-world aircraft. Because we do not have validated data that show the onset of anomalies or any other indication of overconsumption we cannot present the area under ROC curves as a metric for the validity of the results. Instead we show the internal consistency of the results for different algorithms and compute the probability of the degree of observed consistency using a Monte-Carlo simulation.

In order to test the amount of overfitting occurring in the models, we did one study in which we trained on the data from City Pair A $\rightarrow$ B and tested on the same city pair in the same direction. We used 50% of the data for training and the remaining 50% of the data for testing. We computed the normalized root mean squared error (NRMSE) between the predictions and the actual values of fuel flow for the training data set and the testing data set. The NRMSE is defined as the root mean squared error divided by the standard deviation of the actual value. We found that the ratio of the training to

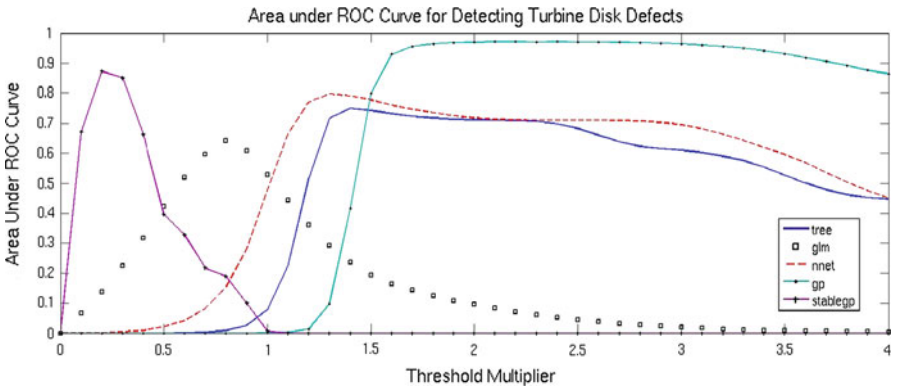

Fig. 5 This figure shows the area under the ROC curve as a function of the threshold multiplier α described in Eq. 4. Each curve represents the performance of a different regression method on detecting anomalies in a turbine engine disk spin test. The performance of all algorithms is acceptable, with the best performance achieved by the Gaussian Process regression method. This method has a high accuracy for a large interval of the threshold multiplier

test NRMSE varied consistently between 85–90% across all the four models tested, thus indicating a slight amount of overfitting.

Once the Virtual Sensors were trained on the data from City Pair A $\rightarrow$ B, we tested them on the hold-out data set for two different city pairs: (1) City Pair B $\rightarrow$ A and (2) City Pair B $\rightarrow$ C. We also trained models on City Pair A $\rightarrow$ B and tested them on City Pair A $\rightarrow$ B (i.e., for training and testing on data from flights in the same geographical direction). We show the results of the algorithms on worst-case scenarios in which the algorithm is trained on one set of city pairs but tested on different sets of city pairs. Good performance in this setting implies that the Virtual Sensors are not somehow specialized for a specific city pair.

In essence the process of computing the percentage of time where the actual fuel is significantly greater than the estimated fuel provides a score which can be used to rank order the list of flights that occurred between the two City Pairs. We average the anomaly scores by Tail Number as a post processing step and then analyze the top Tail Numbers which have the highest average anomaly scores. This reveals that out of the 84 Tail Numbers in our data set, some Tail Numbers have higher than expected fuel consumption over a very long period of time.

Table 2 shows the ten tail numbers with the highest percentage of outlying fuel consumption in the test set corresponding to the City Pair B $\rightarrow$ A using the Elastic Net algorithm (labeled glm, or generalized linear model, in subsequent tables). Note that the model is trained on the data corresponding to the flights in the reverse direction, namely from A $\rightarrow$ B. Each row in the table corresponds to a different tail number. The first column shows the average percentage of cruise time that the tail number was consuming more fuel than expected for the flights between City Pair B $\rightarrow$ A. The two adjacent columns are similarly defined. The last column shows the number of flights for the tail number in the test set. Thus, for example, the first row corresponds to a tail number which flew 8 times from City Pair B $\rightarrow$ A. For the cruise portions of the

Table 2 This table shows the output of the Elastic Net for testing data for the B → A city pair

High (%)	Low (%)	Within bounds (%)	Number of flights
59.3	8.6	32.1	8
55.0	22.3	22.7	11
53.4	18.3	28.3	3
52.5	8.92	38.6	2
50.5	17.8	31.7	11
50.3	15.9	33.8	6
48.4	15.4	36.2	5
47.3	14.2	38.5	8
46.2	18.2	35.6	8
44.0	23.6	32.5	7

Each row represents one tail number shows the top 10 most anomalous tail numbers as discovered in the testing data set. The first column represents the percentage of cruise time in which Virtual Sensors estimates that the aircraft was consuming more than the expected amount of fuel. The two adjacent columns are similarly defined. The last column shows the number of flights for that tail number in the data we analyzed. These outliers represent about 3.5% of the flights for this City Pair, which is a small fraction consistent with anomalies seen in real-world operations

flight, it used more fuel than expected based on the Virtual Sensors 59.3% of the time, was below expected consumption about 8.6% of the time and was within bounds the remainder of the time. Our studies show that the most prominent outlier (not shown on the table) is for the fictitious tail number which we inserted into the data set.

The outliers shown in Table 2 are taken across 1938 flights, which represents an anomaly rate of approximately 3.5%. This rate, of course, is dependent on the threshold multiplier that is chosen. For this study we used a value of 3. However, we know from Fig. 4 that the accuracy of the method is dependent on the value of the threshold multiplier and the algorithm used. We show the results using the Elastic Net since its performance characteristics are different than the other algorithms tested, based on the previous two examples. Once we obtain validation information from an airline operator we can choose the appropriate algorithm and the associated threshold. We also point out that the anomaly rate described here is not inconsistent with the variations expected by operators based on numerous discussions we have had with them and related aviation experts regarding this matter.

Figure 6 shows the model prediction for a different City Pair (B → C) using the Elastic Net algorithm. This tail number corresponds to the same one shown in the first row of Table 2 for a different City Pair. Each flight occurred in December 2005. The green bands show the threshold for Virtual Sensors based on three standard deviations of the distribution of the model predictions. The red band line shows the actual consumption. Whenever the actual consumption falls above the green bands, an anomaly is tagged. The blue lines above the curves indicate each time step where an overage is detected.

Although we are able to detect these anomalies, it is possible that the rank ordered lists of Tail Numbers produced by the various algorithms and the City Pair

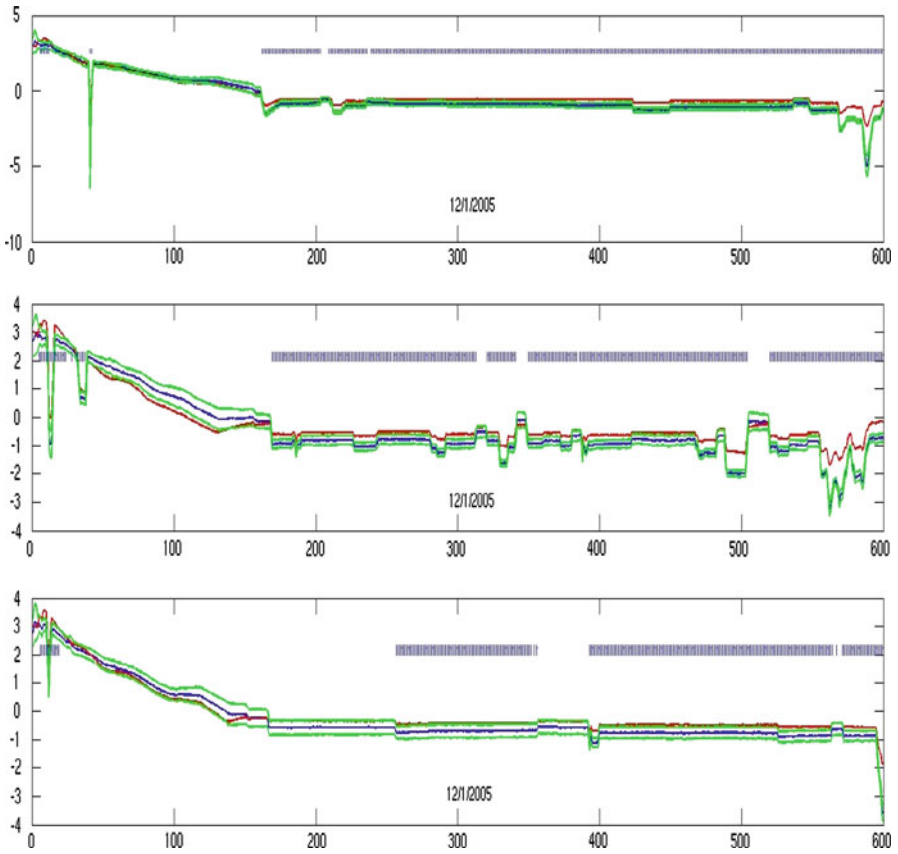

Fig. 6 This figure shows one of the top outliers with respect to fuel consumption on the city pair B → C for a specific tail number. Each panel represents one flight of a specific tail number and is labeled with the month and year of the flight. The predicted mean and standard deviations are shown for the bagged Elastic Net. The bars at the top of a panel show the regions where the actual fuel consumption exceeds the threshold of the 3 standard deviations above the mean

combinations may in fact be random, i.e., it is possible that the Virtual Sensors methodology may not be generating consistent rank orderings of the Tail Numbers for different algorithms and City Pair combinations. If this were the case, it would indicate a significant issue in the methodology used to generate the anomaly scores. Table 3 shows a pair-wise comparison of algorithms and City Pair combinations for all four algorithms tested. Certain trends are evident including the fact that the models tend to have higher overall agreement on the B → A City Pair rather than the B → C City Pair.

In order to test this hypothesis, for the 84 Tail Numbers in the data base, we randomly simulated 100,000 rank orderings, each generated randomly and computed the expected overlap in the top quartile of the lists. If the rank orderings are truly random from different algorithm and City Pair combinations, then we would expect to see an overlap between the two rank ordered lists near the modal number of overlaps in

Table 3 This table shows the agreement between the five algorithms (sgp = Stabilized Gaussian Process, glm = Elastic Net, nnet = Neural Network, tree= Tree, and gp = Gaussian Process Regression) regarding the top 20 outlying Tail Numbers for the two city pairs considered for testing

		B → A					B → C				
		sgp	glm	nnet	tree	gp	sgp	glm	nnet	tree	gp
B → A	sgp		12	18	12	14	10	10	7	15	11
	glm	12		11	15	9	15	5	8	10	15
	nnet	18	11		13	15	10	8	6	16	12
	tree	12	15	13		10	14	6	8	12	14
	gp	14	9	15	10		9	9	7	15	9
	sgp	10	15	10	14	9		5	8	10	12
B → C	glm	10	5	8	6	9	5		8	7	5
	nnet	7	8	6	8	7	8	8		7	6
	tree	15	10	16	12	15	10	7	7		11
	gp	11	15	12	14	9	12	5	6	11	

Each cell corresponds to a city pair and algorithm combination. For example, the number 12 in the first column and second row shows that in the top 20 outlying Tail Numbers as detected by the Virtual Sensors based on Elastic Nets (glm) 12 of the top 20 outliers agree with the top 20 outliers as generated by the stablegp-based Virtual Sensors. The table shows extremely high agreement in the rank ordering of outliers across different algorithms. The Gaussian Process based Virtual Sensors algorithm shows the highest agreement with the stable gp Virtual Sensors on the B → C city pair

randomly generated rank ordered lists. Our simulations revealed that the algorithm and City Pair combinations are remarkably consistent for the algorithms used in this study. In fact, the same Tail Numbers tend to appear in the top quartile of the list regardless of the algorithm or City Pair combination that is used. Figure 7 shows the probability that two lists will have the overlap shown on the x-axis using the simulations mentioned above. For example, we see that the two lists share 11 of the same Tail Numbers in the top 20 Tail Numbers in the rank ordered lists with probability of approximately 0.001 which indicates that the chance that these two rank orderings were generated randomly is very small. As shown in Table 3 the degree of overlap can be as high as 18 for some pairs of algorithms.

7 Implications for aviation carbon footprint

As we have seen in this paper, the carbon footprint of the aviation system is significant. A single Boeing 747 may emit over 100,000 kg of carbon into the atmosphere on a single flight. Data mining technologies can be used to potentially reduce these emissions by revealing situations where aircraft expend more fuel than is expected. In some cases operational changes may be possible that can help reduce fuel consumption. To detect these situations, this requires the development of a method to detect the expected fuel burn, which is the subject of this paper. This approach is not unique to aviation in that it can be applied to numerous other situations where streaming data

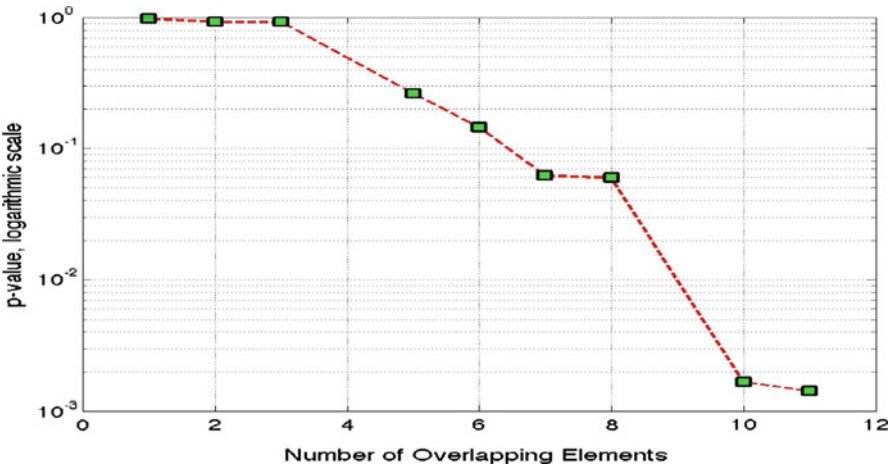

Fig. 7 This figure shows the probability that two different Virtual Sensors algorithms would have a given number of matches in the top 20 Tail Numbers from a set of 84 possible Tail Numbers. It shows that an agreement of 11 has a probability of nearly $p = 0.001$. This shows that the Virtual Sensors algorithms are generating rank ordered lists that are similar, thus validating the methods for different algorithms and city pair combinations

is available and fuel consumption is measured. For example, the automotive industry may benefit from the approach discussed here.

It is evident that the airlines are extremely diligent in their monitoring of fuel consumption for economic and environmental reasons. However, analyses performed using snapshots of data and the aggregate fuel consumed on a flight can reveal larger deviations. More subtle deviations must be detected using higher frequency data. Our research indicates that it is possible for an aircraft to have modal fuel consumption and yet have periods of time where the fuel consumption is higher than expected. If those periods can be addressed through some intervention, it is likely that the aircraft efficiency can improve and the carbon footprint can be reduced.

8 Conclusions and future work

As of the writing of this paper we do not have validated labeled data indicating fuel anomalies or other conditions on the aircraft. Those tests must be performed on much larger data sets with the assistance of the airline. It is possible that the outliers we observe in this study are due to slightly different aircraft or engine configurations. Even if that were the case, we have demonstrated the ability to detect these engine configurations and their impact on fuel consumption. Thus, further validation in an operational environment is essential for this proof-of-concept study. We plan to study these methods on data sets with at least 100,000 flights in a large scale, distributed computing environment and perform the required validation in a real aircraft setting. Other future activities could include the development of online algorithms for Virtual Sensors to further help address the scaling issue.

Another area of research, assuming that the results thus far are validated further on larger data sets, is the development of techniques to identify the causal factors for the excess fuel consumption. It may be that the larger deviations from expected behavior are due to uncontrollable environmental or external factors. However, if we are able to determine the underlying causal factors and they are actionable, these technologies could significantly impact the overall approach to fuel analysis in the aviation domain. A third area of research that we plan to explore is developing combined supervised and unsupervised anomaly detection methods where the underlying distribution of nominal data can be used to inform the supervised model of potential outliers, independent of the predictions of the ensemble of regressors. This work may allow us to improve the quality and interpretability of the results even further.

Acknowledgements The author would like to thank Irving Statler for extremely thoughtful and useful discussions throughout this project. The author also thanks Timothy Woodbury for valuable discussions and Don Simon of NASA Glenn Research Center for providing key references and analytical advice. He would also like to acknowledge Nikunj Oza for valuable discussions and the reviewers for their constructive feedback. The author also thanks our airline partner in providing access to their flight operational data to enable this study. This research was conducted with the support of the NASA Aviation Safety Program's System-Wide Safety and Assurance project. The code used for many of the algorithms in this paper is available as opensource and can be found on DASHlink at https://c3.nasa.gov/dashlink/projects/7/.

References

Abdul-Aziz A, Woike M, Oza N, Matthews B, Baakilini G (2010) Propulsion health monitoring of a turbine engine disk using spin test data. Smart structures and materials and nondestructive evaluation and health monitoring

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Chandola V, Vatsavai R (2010) Scalable hyper-parameter estimation for Gaussian process based time series analysis. In: Proceedings of the SIGKDD workshop on large-scale data mining, Washington DC

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3)

Chen T, Ren J (2009) Bagging for Gaussian process regression. Neurocomputing 72(7–9):1605–1610

Chu E, Gorinevsky D, Boyd SP (2010) Detecting aircraft performance anomalies from cruise flight data. In: Proceedings of the AIAA infotech aerospace conference, Atlanta, GA

Collobert R, Bengio S (2001) SVMTorch: support vector machines for large-scale regression problems. J Mach Learn Res 1:143–160

Committee on Aeronautics Research and Technology for Environmental Compatibility (2002) For greener skies: reducing environmental impacts of aviation. Tech report, National Academies Press

Foster L, Waagen A, Aijaz N, Hurley M, Luis A, Rinsky J, Satyavolu C, Way M, Gazis P, Srivastava AN (2009) Stable and efficient Gaussian process calculations. J Mach Learn Res 10:857–882

Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Softw 33(1):1–22

Friedman J, Hastie T, Höfling H, Tibshirani R (2007) Pathwise coordinate optimization. Annals Appl Stat 1:302–332

Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning: data mining, inference, and prediction. Springer, New York

Hastie T, Friedman JH, Tibshirani R (2009) Glmnet for Matlab. Technical report, Stanford University

Litt JS, Frederick DK, DeCastro JA (2007) User's guide for the commercial modular aero-propulsion system simulation (c-mapps). NASA technical report

Matlab Statistics Toolbox (2010) Ver. 7.4, The MathWorks, Inc., Natick

Matthews B, Srivastava AN (2010) Adaptive fault detection on liquid propulsion systems with virtual sensors: algorithms and case study. In: Proceedings of the joint army navy NASA air force conference on propulsion

Nabney IT (2001) NETLAB-algorithms for pattern recognition. Springer, New York

Oza N (2010) FLTz simulated data. https://c3.ndc.nasa.gov/dashlink/resources/142
Penner JE, Lister DH, Griggs DJ, Dokken DJ, McFarland M (eds) (1999) Global aviation and the atmosphere. Technical report, Intergovernmental Panel on Climate Change
Petrovskaya A, Thrun S (2009) Model based vehicle detection and tracking for autonomous urban driving. Autonomous robots. Springer, NEW York
Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. MIT Press, Cambridge
Sallee GP (1980) Performance deterioration based on in-service engine data. NASA technical manuscript CR-159525
Seber GAF, Wild CJ (1989) Nonlinear regression. Wiley, New York
Simon DL (2010) A three-dimensional receiver operator characteristic surface diagnostic metric. IN: Annual conference of the prognostics and health management society
Srivastava AN, Das S (2009) Detection and prognostics on low dimensional systems. IEEE Trans Syst Man Cybernet C 39(1)
Srivastava AN, Oza NC, Stroeve J (2005) Virtual sensors: using data mining techniques to efficiently estimate remote sensing spectra. IEEE Transactions on Geoscience and Remote Sensing 43(3)
Statler IC (2007) The aviation system monitoring and modeling (ASMM) project: a documentation of its history and accomplishments: 1999 to 2005. NASA technical publication TP-2007-214556
Way MJ, Srivastava AN (2006) Novel methods for predicting photometric redshifts from broad-band photometry using virtual sensors. Astrophys J 647:102–115
Wulf R (1980) Engine diagnostics program: Cf-60 engine deterioration. NASA technical report CR-159867