ANOMALY DETECTION IN AIRCRAFT DATA USING RECURRENT NEURAL NETWORKS (RNN)

Anvardh Nanduri (M.S. Candidate), Lance Sherry (Ph.D.)

Center for Air Transportation Systems Research (CATSR) at George mason University (GMU), Fairfax, Virginia.

Abstract

Anomaly Detection in multivariate, time-series data collected from aircraft's Flight Data Recorder (FDR) or Flight Operational Quality Assurance (FOQA) data provide a powerful means for identifying events and trends that reduce safety margins. "Exceedance Detection" algorithms use a list of specified parameters and their thresholds to identify known deviations. In contrast, Machine Learning algorithms detect unknown unusual patterns in the data either through semi-supervised or unsupervised learning. The Multiple Kernel Anomaly Detection (MKAD) algorithm based on One-class SVM identified 6 of 11 canonical anomalies in a large dataset but is limited by the need for dimensionality reduction, poor sensitivity to short term anomalies, and inability to detect anomalies in latent features.

This paper describes the application of Recurrent Neural Networks (RNN) with Long Term Short Term Memory (LTSM) and Gated Recurrent Units (GRU) architectures which can overcome the limitations described above. The RNN algorithms detected 9 out the 11 anomalies in the test dataset with Precision = 1, Recall = 0.818 and F1 score = 0.89. RNN architectures, designed for time-series data, are suited for implementation on the flight deck to provide real-time anomaly detection. The implications of these results are discussed.

1 Introduction

Anomaly Detection in multivariate, time-series data collected from aircraft Flight Data Recorders (FDR) or Flight Operational Quality Assurance (FOQA) data provides a powerful means for identifying events, trends, or pre-cursors that reduce safety margins. For example, cluster analysis identified specific mountainous airports that experienced excessive Terrain Awareness and Warning System (TAWS) alerts [1], and some anomaly detection algorithms identified speed, altitude and airspace violations, as well as mode oscillations [2][3][4].

Exceedance Detection algorithms [5] use sets of rules to detect abnormalities in archived FDR data during various phases of flight. The performance of Exceedance Detection algorithm is dependent on the design of the "rules". In contrast, Machine Learning algorithms detect unusual (i.e. not normal) patterns in the input data by learning a "decision boundary." Li et al. [4] compared the performance of Multiple Kernel Anomaly Detection (MKAD) [2][3] and Cluster Anomaly Detection algorithms [4] by testing on the same set of data with 11 known anomalies. These two types of algorithms identified 6 of the 11 anomalies. Further, these two types of machine learning algorithms are limited by: (i) the need for dimensionality reduction, (ii) poor sensitivity to short term anomalies, and (iii) the inability to detect anomalies in latent features. Further they cannot be implemented for real-time anomaly detection on the flight deck.

This paper describes the application of Recurrent Neural Networks (RNN) for effectively detecting anomalies in flight data. Recurrent Neural Networks with Long Short Term Memory cells (RNN LSTM) and Recurrent Neural Networks with Gated Recurrent units (RNN GRU) are capable of handling multivariate sequential, time series data without dimensionality reduction, and can detect anomalies in latent features. RNNs can also be implemented for real-time anomaly detection on the flight deck.

The performance of RNN LTSM and RNN GRU results is compared with the performance of MKAD on 11 pre-defined anomalies in archived time series flight data generated for this study from simulated arrivals into SFO 24L and 24R [6]. The RNN LSTM and RNN GRU detected 9 out 11 anomalies in the test set (Precision = 1, Recall = 0.818 and F1 score = 0.89) compared to 6 out of 11 for MKAD (Precision = 1, Recall = 0.5, F1 score = 0.66).

RNNs demonstrated the ability to detect more subtle anomalies than MKAD such as Flight Director toggling, and one case of runway alignment anomaly. RNN, like MKAD failed to detect changes in pitch for short duration and subtler case of runway alignment change. In addition, by accepting sequences of multivariate data points as input and treating them as time series data, pre-trained RNNs models can be used for real-time anomaly detection on-board the aircraft.

This The rest of the paper is organized as follows: Section 2 describes previous research in flight data anomaly detection and a description of RNNs. Section 3 describes the source of data and the anomalies used for this study. Section 4 describes the results of the study. Section 5 provides Conclusions and Future Work.

2 Previous Research in Flight Data Anomaly Detection

Exceedance Detection

Exceedance Detection is the standard flight data analysis method used in the airline industry today[5], [7[. Flight parameters are checked for exceedance of predefined limits under certain conditions. The limits and conditions are defined in advance by domain experts based on airlines standard operating procedures. Example events include: pitch and speed during takeoff, flap retraction points, and various parameters for stabilized approach. Thus unknown and unspecified conditions cannot be detected by this method.

Multiple Kernel Based Anomaly Detection (MKAD)

Multiple Kernel Based Anomaly Detection (MKAD) uses Multiple Kernel Learning MKL [8][9] to simultaneously incorporate separate kernel functions for both discrete and continuous sequences.

One-class Support Vector Machines [9][10] are used as the anomaly detection method in MKAD. The One-class SVM is a semi-supervised method, that constructs an optimal hyperplane in the high dimensional feature space by maximizing the margin between the origin and the hyperplane. This is accomplished by solving an optimization problem [22] whose dual form is given as:

$$\text{minimize } Q = \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)$$

$$\text{subject to } 0 \le \alpha_i \le \frac{1}{l\nu}, \sum_i \alpha_i = 1, \rho \ge 0, \nu \in [0,1]$$

where α_i is Lagrange multiplier, ν is adjustable parameter gives upper bound on training error and lower bound on the fraction of training points that are support vectors. Solving this optimization problem yields at least νl training points whose Lagrange multipliers are greater than zero and these data points are called support vectors. ρ is a bias term and K is the composite kernel obtained by linear combination of discrete kernel K_d and continuous kernel K_c given as:

$$K(x_i, x_j) = \eta K_d(x_i, x_j) + (1-\eta) K_c(x_i, x_j)$$

Where η is a parameter which controls the contribution of both kernels.

The Support vectors $x_i : i \in [l], \alpha_i > 0$ are either marginal $\xi_m = i : 0 < \alpha_i < 1$ or non-marginal $\xi_{nm} = i : \alpha_i = 1$. Once support vectors $\alpha^{\cdot}$ are obtained, the following decision function given by following equation and is used to determine if a test data point is normal or anomalous. Data points with negative values are classified as anomalous and points with positive values are treated as normal.

$$f(x_j) = sign\left(\sum_{i \in \xi_m} \alpha_i k(x_i, x_j) + \sum_{\xi_{nm}} \alpha_i k(x_i, x_j - \rho)\right)$$

Clustering based Anomaly Detection (ClusterAD)

Clustering based Anomaly Detection (ClusterAD) [4] initially converts the raw data into time series data. In order to map data into comparable vectors in the high dimensional space, these time series data from different flights are anchored by a specific event to make temporal patterns comparable. Then every flight parameter is sampled at fixed intervals by time, distance or other reference from the reference event. All sampled values are arranged to form a high dimensional vector for each flight in the following form:

$$\left[x_{t_1}^1, x_{t_2}^1, \dots, x_{t_n}^1, \dots, x_{t_1}^m, x_{t_2}^m, \dots, x_{t_n}^m\right]$$

Where $x_{t_j}^{i}$ is the value of the ith flight parameter at sample time t_j, m is the number of flight parameters and n is the number of samples for every flight parameter.

Limitations of Current Methods

Need for Dimensionality Reduction

Both MKAD and ClusterAD pre-process that data to convert sequential data for the entire flight into a high dimensional time series data using complex dimensionality reduction techniques that map the high dimensional data to a low dimensional feature space. Specifically, MKAD uses Symbolic Aggregate Approximation (SAX), and ClusterAD uses Principal Component Analysis (PCA) as dimensionality reduction techniques. This is a restriction on the application of these algorithms for real time anomaly detection. Moreover, ClusterAD requires the feature vectors of multiple flights to be aligned with respect to a specific event for meaningful comparisons.

Poor sensitivity towards short term anomalies.

Past studies by [4] found that both MKAD and ClusterAD are not sensitive to anomalous patterns which occur for short durations. One of the reasons could be that, due to data compression during dimensionality reduction some of these nuances would have been lost.

Inability to detect anomalies in Latent Features

Li et al. [4] discuss that both MKAD and Cluster AD cannot detect anomalies in features that are not explicitly present in the feature vector, although these latent features are derivable from existing features. For example, both algorithms failed to detect abnormal pitch rate during landing as the pitch rate was not part of the feature vector. The dataset included pitch value as one of the features.

Recurrent Neural Networks

Recurrent Neural Networks (RNN) differ from standard neural networks by allowing the output of hidden layer neurons to feedback and serve as inputs to the neurons. In this way the network is able to use past history as a way to understand the sequential nature of the data. Two types of RNNs are used in this paper.

Long Short Term Memory (LSTM) facilitate learning long term dependencies in the input [12]. The LSTM architecture consists of a set of recurrently connected structures called memory blocks. Each block contains one or more self-connected memory cells each with an associated cell state. The memory block is built around the cell(s) which ensure constant error flow through them by using an identity function and always getting incoming unit weights. Thus these units solve the vanishing/exploding gradient problem [16] [17] commonly observed in traditional recurrent neural networks.

Gated Recurrent Units are special variants of LSTMs that merge the forget and input gates into a single update gate resulting in a simpler model than standard LSTM models [14].

Recurrent Neural Networks based on LSTM and GRU units do not have the limitations of MKAD or Cluster AD as RNNs are by definition capable of handling multivariate sequential input data without any modifications and treat it as time series data.

3 Flight Data Source for Analysis

The Distributed National FOQA (Flight Operations Quality Assurance) Archive (DNFA) contains data from flight data recorders of over two million flights that have been submitted by more than 10 participating carriers. All the data is de-identified and protected by data confidentiality, proprietary data clauses and security policies. Due to the data confidentiality, proprietary and security policies, this data is not publicly available to researchers developing innovative machine learning or data mining algorithms.

To overcome this limitation, large sets of FOQA-like data for development and testing of machine learning/data mining algorithms can be generated through simulated flights [6]. This data is not restricted and can be made publicly available.

For the purpose of this study, X-Plane Simulation, using the X-plane Software Development Kit (XSDK), was configured to run in a Monte Carlo shell to generate FOQA-like data [6]. External plugins were developed to manipulate the simulation set-up configuration, the pilot commands, and a Monte Carlo shell.

FOQA data was generated for a Boeing 777-200 ER aircraft in X-Plane for 500 approaches into San Francisco (KSFO) airport. Twenty one (21) continuous and discrete variables are recorded at the sampling rate of 2 Hz, which include aircraft state and automation state parameters for the approach phase of flight. The dataset included a total of 500 flights of which 485 are normal flights and 15 are anomalous flights.

The X-Plane simulation was able to generate 485 normal flights in approximately 72 hours, running on a T7500 machine with Intel® Xenon® CPU, with 6 GB memory on a windows 64 bit Operating System. See Nanduri & Sherry [6] for more information.

Anomalies

Eleven canonical anomalies from [2], [3], and [4] were introduced into the data set:

1) Very High Airspeed Approach (Rushed and Unstable Approach)
2) Landing Runway Configuration Change case 1
3) Landing Runway Configuration Change case 2
4) Auto Land without Full Flaps(Unusual Auto Land Configuration)
5) Auto Land with Single Auto Pilot (Unusual Auto Land Configuration)
6) High Energy Approach (Too High or Too Fast or both)
7) Recycling FDIR
8) Influence of Wind
9) High Pitch Rate for Short Duration
10) High Airspeed for Short Duration
11) Low Energy Approach

4 Results

This section describes the results of the analysis of training and testing RNN – LSTM, RNN- GRU, and MKAD algorithms on the FOQA-like data generated from the X-Plane simulation.

RNN Architectures

Recurrent Neural Networks used in this analysis includes 10 architectures of RNNs Long Short-Term Memory units (LSTM) and 10 architectures with Gated Recurrent Units (GRU). The following parameters are varied to generate different RNN architectures:

1. Number of iterations (epochs) training examples are presented to the network for learning
2. Number of hidden layers and number of hidden units in each layer
3. Number of time steps that RNNs are allowed to look into past
4. The dropout ratio which determines the percent of neurons randomly dropped at that layer before each iteration to improve generalization
5. Batch size which determines number of examples presented at a time to the network during training
6. Validation split which determines percent of training examples used to calculate the validation loss of the trained network

The *adam* optimizer[13] was used with default arguments. The loss or cost function used is Mean Squared Error in all the models. In the input layer and output layers consist of 21 neurons each to match the data used. In this way, the predicted feature vector for the next time step is the output of the RNN.

All the networks are designed and trained in Keras- a Theano based deep learning library. The details of RNN models using GRU are summarized in Table 1 and details of RNN models using LSTM are summarized in Table 2.

TABLE 1: Parameter Combinations for GRU RNN Models

Model	Time steps	Dropout	Config	Batch Size	Epochs	Validation
GRU1	60	0.2	30	30	40	0.2
GRU2	60	0.2	30	30	60	0.2
GRU3	60	0.2	30	30	90	0.2
GRU4	60	0.2	30	30	120	0.2
GRU5	60	0.2	60	30	120	0.2
GRU6	30	0.2	60	30	120	0.2
GRU7	60	0.1, 0.1	30, 30	30	120	0.2
GRU8	60	0.2, 0.2	30, 30	30	120	0.2
GRU9	60	0.2, 0.2	30, 30	30	120	0.3
GRU10	60	0.2, 0.2	60, 60	30	120	0.3

TABLE 2: Parameter Combinations for LSTM RNN Models

Model	Time steps	Dropout	Config	Batch Size	Epochs	Validation
LSTM1	60	0.2	30	30	40	0.2
LSTM2	60	0.2	30	30	60	0.2
LSTM3	60	0.2	30	30	90	0.2
LSTM4	60	0.2	30	30	120	0.2
LSTM5	60	0.2	60	30	120	0.2
LSTM6	60	0.2, 0.2	60, 60	60	40	0.2
LSTM7	60	0.2, 0.2	30, 30	60	40	0.2
LSTM8	90	0.2, 0.2	30, 30	60	40	0.2
LSTM9	60	0.2, 0.2	30, 30	30	40	0.3
LSTM10	60	0.2, 0.2	60, 60	30	40	0.3

Data for Training and Testing

The dataset from the X-Plane simulation is split into 478 training examples and 22 test samples. Among 22 test samples, 11 are positive anomalous examples and other 11 are normal examples.

Training the RNN

The dataset is normalized so that all the values are in the range [0,1]. The normalized feature vectors are sampled at regular time intervals (i.e. 2 secs) with 21 features per vector that are presented to RNNs sequentially. The algorithm is iterated through the training samples according to the number of times as specified by epochs parameter.

The output of the recurrent neural network at time t is the predicted value at next time step(s) $t+1$. The ideal output from the RNN at time t is the actual value (input) at time $t+1$. To calculate the error during training, the expected output at time t, $Y(t) = X(t+1)$, is the actual input for the next time step. It is important to note that since during the training phase the model has knowledge (both temporal and featural) about all the examples it can follow this methodology to calculate the training error.

In contrast, during testing, there is no requirement for $X(t+1)$ during time t. In fact, at time $t+1$, the model calculates the error of value predicted at time t. In this way during testing, if the resultant error is low, it signifies that the current value(s) are normal. On the other hand, if the resultant error is relatively high, it indicates the presence of anomaly.

Note that, during online/real time testing, as the network receives input values sequentially, ideally both point type and contextual anomalies can be detected.

Anomaly Detection Results

Both the MKAD and RNN algorithms were trained and tested on the same data for the approach into SFO. The overall results are summarized in Table 3.

All RNN models are able to detect 8 out of 11 anomalous cases. Anomalous flights with abnormal Pitch for short duration and the second case with a Runway change were not detected by either the MKAD or RNN models.

The MSE values for the 11 anomalies is summarized in Figure 2. The greater the MSE value, the less ambiguity in the anomaly detection.

TABLE 3: MKAD and RNN performance on 11 Canonical Anomalies

Anomaly	MKAD1	MKAD2	MKAD3	LSTM	GRU
PTCH	No	No	No	No	No
FDIR	No	No	No	Yes	Yes
SHORT	Yes	Yes	Yes	Yes	Yes
HENG	No	Yes	No	Yes	Yes
FLAP	Yes	Yes	Yes	Yes	Yes
1AP	Yes	Yes	Yes	Yes	Yes
RW1	No	No	No	Yes	Yes
RW2	No	No	No	No	No
VHSPD	Yes	Yes	Yes	Yes	Yes
WIND	Yes	No	No	Yes	Yes
LENG	No	Yes	No	Yes	Yes

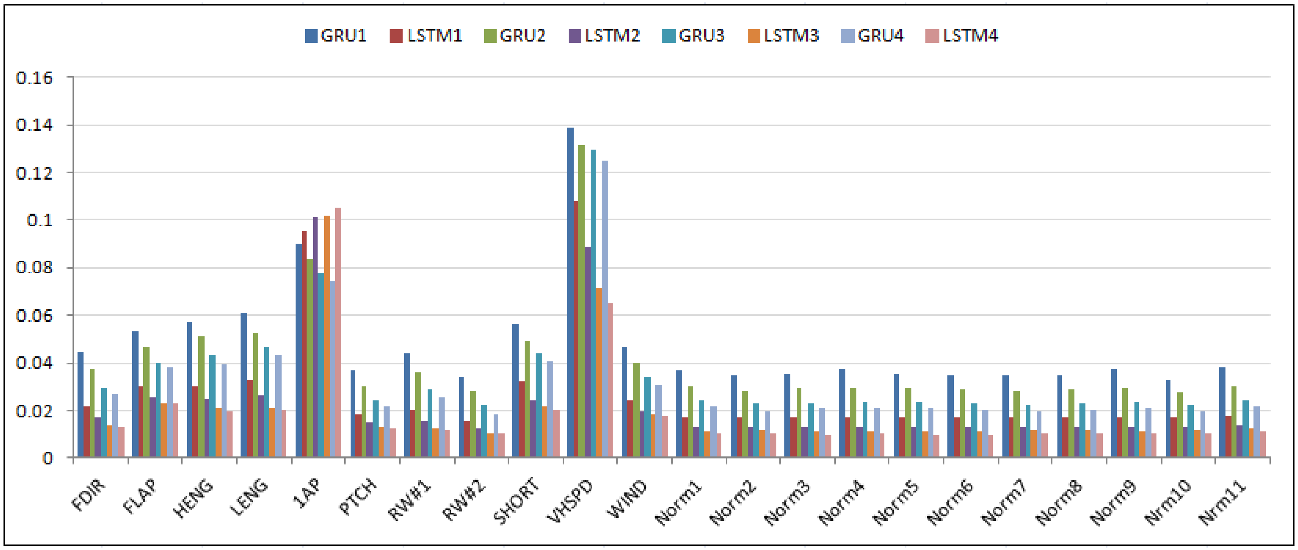

FIGURE 2: RNN: Reconstruction errors on test data (RMSE values)

Normal flights were also part of the test set. Both the RNN and MKAD algorithms successfully classified them as negative. In this way, the RNNs did not exhibit any false positives yielding an ideal *precision* value equal to 1 (Table 4).

Since the majority of anomalies are classified correctly, they also resulted in high *Recall* values and thus high F_1 scores. The details are presented in Table 4. All models using RNNs with LSTM units as well as GRU units performed equally well. All the models with various configurations yielded similar results with high precision and high recall values.

Since $F1$ score considers both precision and recall values, it best represents the overall performance of the models. As shown in Table 4 RNN-LSTM and RNN-GRU outperformed MKAD in terms of all three metrics. Though MKAD is able to achieve high precision values, as a result of false negatives their overall performance was poor.

For more details on this study see [15].

5 Conclusions

This paper describes the application of Recurrent Neural Networks (RNN) in detecting 11 canonical anomalies found in the literature [2], [3] and [4]. The analysis was done using a FOQA-like data set generated from an X-Plane simulation. The MKAD algorithm [2], [3] and the RNNs were trained and tested on this data set.

RNNs were able to unambiguously detect 8 out of the 11 anomalies. One additional anomaly was close to detection. The MKAD algorithm detected 6 out of the 11 anomalies.

RNNs are designed to handle multivariate time series data such as FOQA data. Further, they do not need dimensionality reduction, they are more sensitive to short-term anomalies, and can detect anomalies in latent failures.

RNNs are also suitable for implementation on the flight deck for real-time anomaly detection.

TABLE 4: Performance of MKAD and RNNs

Model	Precision	Recall	F1 score
MKAD1	**1**	0.454	0.624
MKAD2	**1**	0.545	0.706
MKAD3	**1**	0.363	0.534
RNN-LSTM	**1**	**0**.818	**0**.899
RNN-GRU	**1**	**0**.818	**0**.899

Future Work

Continue to refine the RNNs to detect runway change configuration and abnormal pitch anomalies. Experiments with varying feature combinations may be valuable in assessing the performance of recurrent neural networks in detecting even the subtlest anomalies in the dataset. Also expand the data set beyond 21 parameters, and evaluate the performance of proposed models using various feature combinations.

References

[1] Lee, J., (2012) The TAWS Model And Its Applications. In Proceedings 31st Digital Avionics Systems Conference. 10/17/2012

[2] Das, S. B. L. Matthews, A. N. Srivastava, and N. C. Oza, (2010) "Multiple kernel learning for heterogeneous anomaly detection: Algorithm and aviation safety case study," In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10, 2010, pp. 47–56. [Online]. Available: http://doi.acm.org/10.1145/1835804.1835813

[3] Das, S., B. Matthews, and R. Lawrence, "Fleet level anomaly detection of aviation safety data," In *Prognostics and Health Management (PHM), 2011 IEEE Conference on*, June 2011, pp. 1–10.

[4] Li, L., M. Gariel, R. Hansman, and R. Palacios, "Anomaly detection in onboard-recorded flight data using cluster analysis," in *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*, Oct 2011, pp. 4A4–1–4A4–11.

[5] Civil Aviation Advisory Publication (2011) Guidance on the establishment of a Flight Data Analysis Program (FDAP) – Safety Management Systems (SMS). CAAP SMS-4(0)

[6] Nanduri, A., L. Sherry (2016) Generating Flight Operations Quality Assurance (FOQA) Data from the X-Plane Simulation**.** In Proceedings *2016 Integrated Communications, Navigation, Surveillance (ICNS) Conference, Dulles, Va.*April 19-2, 2016

[7] Treder, B., & B. Crane (2004) Application of Insightful Corporations Data Mining Algorithms to FOQA Data at JetBlue Airways. Flight Safety Foundation Report. Available at http://flightsafety.org/files/FOQA_data_mining_report.pdf.

[8] Bach, F.R., G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in Proceedings of the Twenty-first International Conference on Machine Learning, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 6–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015424

[9] Lanckriet, G. R. G., N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," J. Mach. Learn. Res., vol. 5, pp. 27– 72, Dec. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1005332.1005334

[10] Tax, D. M. J. and R. P. W. Duin, "Support vector domain description," Pattern Recognition Letters, vol. 20, pp. 1191–1199, 1999.

[11] Scholkopf, B., J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Comput., vol. 13, no. 7, pp. 1443–1471, Jul. 2001. [Online]. Available: http://dx.doi.org/10.1162/089976601750264965

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[13] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," CoRR, vol. abs/1409.1259, 2014. [Online]. Available: http://arxiv.org/abs/1409.1259

[14] Kingma, D. P., and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[15] Nanduri, S. K. A (2015) Anomaly Detection in Aircraft Performance Data. Masters Thesis for the Degree of Master of Science Computer Science. George Mason University. Thesis Advisors Dr George Tecuci and Dr. Lance Sherry.

[16] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[17] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," Trans. Neur. Netw., vol. 5, no. 2, pp. 157{166, Mar. 1994. [Online]. Available: http://dx.doi.org/10.1109/72.279181

Acknowledgements

The authors acknowledge Dr George Tecuci (GMU), Dr. Xu (GMU) who served as thesis committee members for Mr. Nanduri. Dr Mathews (NASA) provided the MKAD algorithms and patiently answered many questions. Also Dr. Michael Feary (NASA), Dr. Immanuel Barshi (NASA), Dr Robert Mauro (Decision Research Inc.), Mrs Julia Trippe (University of Oregon), Dr John Shortle (GMU), Dr. George Donohue (GMU) Houda Kourdali-Kerkoub (GMU), Zhenming Wang (GMU), Bob Mathews (retired FAA). This research was funded by internal funds from the George Mason University Research Foundation.

Email Addresses

snandusri@masonlive.gmu.edu

lsherry@gmu.edu

2016 Integrated Communications Navigation and Surveillance (ICNS) Conference
April 19-21, 2016