# Anomaly Detection in Aviation Data using Extreme Learning Machines

Vijay Manikandan Janakiraman
UARC/NASA Ames Research Center
Moffett Field, CA 94035, USA
Email: vjanakir@mail.nasa.gov

David Nielsen
MORI Associates/NASA Ames Research Center
Moffett Field, CA 94035, USA
Email: david.l.nielsen@nasa.gov

***Abstract*—We develop fast anomaly detection algorithms using extreme learning machines (ELM) to discover operationally significant anomalies in large aviation data sets. Anomaly detection (aka one-class classification or outlier detection) is an active area of research to identify safety risks in aviation. Aviation data is characterized by high dimensionality, heterogeneity (continuous and categorical variables), multimodality and temporality. To address these challenges, NASA Ames has developed several anomaly detection algorithms including MKAD, the present state of the art [1]. MKAD's computational complexity is quadratic with respect to the number of training examples which makes it time consuming (and sometimes infeasible) for mining very large data sets. In this paper, we utilize ELM's fast training and good generalization properties to develop scalable anomaly detection algorithms for very large data sets. We adapt unsupervised ELM algorithms such as the autoencoder and embedding models to perform anomaly detection. The unsupervised models capture the nominal data distribution and by choosing a desired strength of detection that defines the upper bound of outliers in the training data, the anomaly decision boundary is determined. The autoencoder model detects anomalies as the ones that have a large reconstruction error while the embedding model detects anomalies as the ones that lie outside a hypersphere in the embedded space. The proposed algorithms are applied to a real aviation safety benchmark problem and the results show that the ELM based algorithms are comparable to MKAD in detection while training is made faster by two orders of magnitude.**

## I. Introduction

The US national Airspace System (NAS) is among the safest in the world transporting more than 2 million passengers a day [2]. With increasing traffic demands, new technology is being introduced by the Federal Aviation Administration (FAA), transforming the airspace system with the Next Generation Air Transportation System (NextGen) [3]. The NAS is a highly complex system with multiple interacting elements and ensuring safety is a top priority. One of the goals of NextGen is to mine massive data that is being collected in the NAS to observe and study present and future safety issues so that air safety can be improved. NASA, in partnership with the FAA and industry is continuing to develop new technologies to address this need.

Anomaly detection using data has been an active area of research to discover and study safety events in the NAS. Anomaly detection (aka one-class classification, outlier detection) refers to the task of identifying abnormal patterns in data which sometimes reveal the weak links in the NAS. Anomaly detection for aviation problems has been studied at various levels ranging from simple exceedance based methods to sophisticated kernel based methods [4]. The most common industrial practice is to detect anomalies using domain defined exceedances (simple threshold on some critical parameters). For example, a flight's airspeed $\geq 150kts$ at $1000ft$ altitude before landing may be considered an exceedance. As exceedances are simple and are defined by domain experts, when applied to new data, they often detect previously known and simple anomalies while missing the ones that involve a combination of several parameters.

With advances in data mining, more sophisticated methods were developed such as Morning Report, Orca, IMS, SequenceMiner and MKAD for detecting aviation anomalies [4]. Morning report uses multivariate cluster-analysis to group flights by similarity along flight signatures derived from parameter values, calculates an atypicality score for each flight, and provides a plain-language description of what makes targeted flights atypical [5]. Inductive Monitoring System (IMS) [6] is another distance based unsupervised algorithm to find anomalies using incremental clustering. IMS uses data from nominal operation of the system to build a model and tests a given data sample for its closeness to the nominal model. While Morning Report and IMS operate only on continuous data, Orca [7] works on both continuous and discrete features using a variant of the k-nearest neighbors algorithm. The similarity metric in Orca is calculated based on Euclidean distance and hamming distance for continuous and discrete features respectively. In order to consider the temporal aspect of the data, Sequence Miner [8] was introduced. Sequence Miner works by first using an unsupervised clustering algorithm to cluster the sequences using the normalized longest common subsequence (nLCS) as a similarity metric. Once the clusters are defined, anomalies can be detected using the nLCS as the distance measure.

Discovering anomalies in modern aviation data is challenging because of several factors including high dimensionality (a typical flight operational quality assurance data has about 350 time series variables), heterogeneity (data consists of continuous and categorical variables), multimodality (data consists of a mix of different aircraft types, runways and airports, each having its own unique characteristics) and temporality (data is in the form of long time series). To simultaneously

address these challenges, multiple kernel anomaly detection (MKAD) was developed [1]. MKAD builds separate kernels for continuous and categorical variables taking the temporal aspect into account and combines the different variables in the kernel space. A one-class SVM model is then learned which separates the nominal data from the anomalous data using a hyperplane. MKAD achieves state of the art results in anomaly detection and is being assessed by the FAA for implementation. A major challenge with MKAD is that the kernel building step is time consuming and memory intensive. MKAD's computational complexity is quadratic with respect to the number of training examples which makes it time consuming (and sometimes impossible) for mining very large data sets [9]. With increasing volume of data and NextGen's focus towards realtime safety monitoring, a method that can perform anomaly detection faster is required.

In this paper, we explore anomaly detection algorithms using extreme learning machines (ELM) to take advantage of ELM's fast training and good generalization performance. An ELM is a single hidden layer feed-forward model whose input layer parameters are assigned using random numbers and fixed during training. Thus, ELM training involves solving a linear least squares problem and so training is several times faster compared to SVM and backpropagation neural nets. ELM tends to achieve superior results on many classification and regression problems [10].

Although ELM based algorithms are very popular in supervised learning, little work has been done in ELM based anomaly detection. When prior labels are available, the problem of anomaly detection is converted to a binary classification problem and solved using ELM [11], [12], [13], [14]. Although one-class ELM was recently introduced [15], [16], they require that the models are trained purely on the target (nominal) class data. Such methods may be useful for problems such as system fault detection where it may be very easy to obtain the target class data by running the system in normal mode. A model that is built using this data can be used to detect faults in the test data. In many challenging cases, as in the case of aviation data, it is not trivial to collect purely nominal data for training. For instance, flight data in our domain may be identified as nominal only after a subject matter expert (SME) confirms it, which takes several hours of expert's time. Thus flight anomaly detection algorithms such as MKAD operate on data whose labels are not required to be known a priori. In other words, training data is expected to contain some anomalies. The algorithms that we discuss in this paper works in this setting; i.e., training data may be a mixture of nominal and anomalous data. By choosing a desired strength of detection that defines the upper bound of anomalies in the training data, the decision boundary can be defined. We adapt unsupervised ELM algorithms such as ELM autoencoder [17], [16] and ELM embedding [18] to perform anomaly detection. We explored both a sparse autoencoder model ($L_1$-ELMAD) and a non-sparse $L_2$ regularized autoencoder model ($L_2$-ELMAD) to compare if sparsity helps find better features for anomaly detection. The autoencoder model uses a reconstruction error as the anomaly score while the embedding model (Em-ELMAD) uses the Euclidean distance metric for determining the anomaly score. The three models are applied to an aviation safety benchmark data set and are compared against the state-of-the-art MKAD algorithm. We observe from the experiments that the proposed ELM based algorithms are comparable to MKAD in detection performance with $L_2$-ELMAD slightly better, while being faster by two orders of magnitude.

## II. ELM Based Anomaly Detection

In this section ELM principles are used in developing fast and scalable algorithms for aviation anomaly detection. In most cases, the labels on anomalies are not available a priori and thus anomaly detection falls in the category of unsupervised learning. We develop three anomaly detection algorithms by adapting ELM based unsupervised learning methods; two models based on ELM autoencoding and one model based on ELM embedding. In all the models, the general idea is the following: Given a training data set, obtain a model capturing the data distribution. Based a user-defined expected rate of anomalies, set a threshold on the anomaly scores of the training data. Using the threshold, evaluate the test data set for anomalies. Based on these steps, simple but efficient anomaly detection algorithms are developed.

### A. Autoencoder Based Anomaly Detection

An autoencoder [19], [20] is model that maps the training data onto itself by identifying an efficient latent representation that reconstructs the data well. The encoded representation usually captures prominent features of the data that often results in better classification and regression. For anomaly detection where the data is predominantly nominal with very few anomalies, the autoencoder model captures the features of the nominal data that helps reconstruct it. When the learned features are used for reconstruction, the model finds it hard to reconstruct the anomalies which can then be identified using the reconstruction error.

Consider a training data set $\{x_1, x_2, ..x_N\}$ where $N$ is the number of training data and $x \in \mathbb{R}^d$. The labels $y_i = 1$ (or 0) if $x_i$ is anomalous (or nominal) are not known a priori. The optimization problem that solves an autoencoder training is

$$\min \sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2 \tag{1}$$

where $\hat{x}_i$ is the reconstruction corresponding to the data $x_i$. Using an ELM autoencoder model, we obtain a standard ELM optimization problem given by

$$\min_W \{\|HW - X\|^2 + \lambda_p \|W\|_p\}. \tag{2}$$

where $p = 1, 2$ corresponds to $L_1, L_2$ norm respectively, $\lambda_p$ represents the regularization coefficient and $H = H(x)$ represents the coding function that transforms the training data to some coded representation. $W$ is the output parameters of the ELM model (aka the decoding function) that needs to be

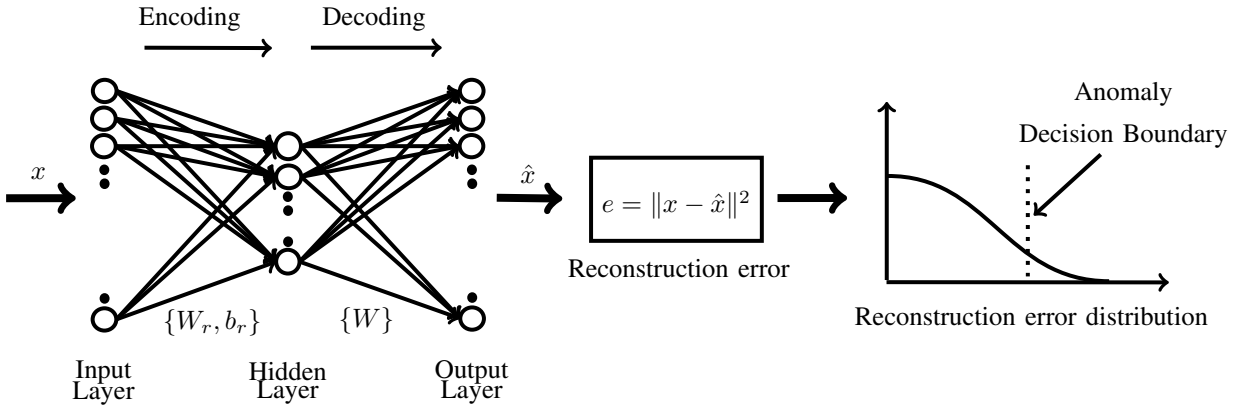


Fig. 1. ELM autoencoder based anomaly detection.

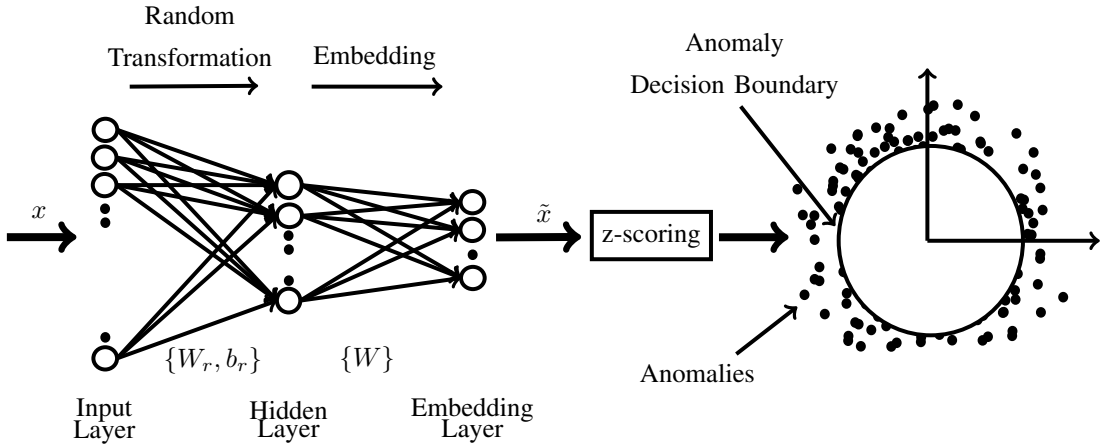


Fig. 2. ELM embedding based anomaly detection.

learned from data. Here we consider both $L_1$ and $L_2$ norms for autoencoding because each has its own advantages and disadvantages. For instance, $L_1$ norm encourages sparsity in $W$ and sometimes provides good feature learning [21] but the optimization problem becomes non-smooth and an iterative algorithm needs to be designed. An $L_2$ norm on the other hand results in a much simpler (closed form) solution to the optimization problem but gives a non-sparse solution which may not be desirable for some applications.

*1) $L_2$-ELMAD:* Using $L_2$ regularization, equation (2) can be modified as

$$\min_W \left\{ \|HW - X\|^2 + \lambda_2 \|W\|^2 \right\}, \tag{3}$$

where $\lambda_2$ represents the $L_2$ regularization coefficient and $H$ in this case is given by

$$H^T = \phi(W_r^T X + b_r) \in \mathbb{R}^{n_h \times N}. \tag{4}$$

where $n_h$ represents the number of hidden neurons of the ELM model, $\phi$ represents the hidden layer activation function (any piecewise continuous function such as sigmoidal, radial basis functions), $b_r$ represents the input bias, $W_r, W$ represents the input and output layer weights respectively. Based on ELM theory, $W_r$ and $b_r$ can be assigned based on any continuous random distribution without even seeing the training data, and is fixed during training. Thus, the nonlinear optimization problem in equation (3) is converted to a linear least squares problem. The learned parameters of ELM autoencoder is given by

$$W^* = \left(H^T H + \lambda_2 I\right)^{-1} H^T X. \tag{5}$$

Using the above ELM autoencoder, the reconstruction $\hat{x_i}$ corresponding to any $x_i$ is given by

$$\hat{x}_i = W^{*T} \phi(W_r^T x_i + b_r). \tag{6}$$

*2) $L_1$-ELMAD:* For the $L_1$-ELMAD, we adapt the ELM based sparse autoencoder for multilayer perceptron developed in [17]. Using $L_1$ regularization, equation (2) can be modified as

$$\min_W \left\{ \|HW - X\|^2 + \lambda_1 \|W\| \right\}, \tag{7}$$

where $\lambda_1$ represents the $L_1$ regularization coefficient and the other variables as defined earlier. If $A(W) = \|HW - X\|^2$ and $B(W) = \lambda_1 \|W\|$, the solution to the problem in (7) is obtained using FISTA (A fast iterative shrinkage-thresholding algorithm) [21] as follows

1) Calculate the Lipschitz constant $\gamma$ of the gradient of smooth convex function $\Delta A$.
2) Begin the iteration by taking $y_1 = W_0, t_1 = 1$ as the initial conditions. Then, for $j(j \geq 1)$, the following holds.
   - $W_j = s_\gamma(y_j)$ where $s_\gamma$ is given by

$$s_\gamma = \underset{W}{\operatorname{argmin}} \left\{ \frac{\gamma}{2} \|W - (W_{j-1} - \frac{1}{\gamma} \Delta A(W_{j-1}))\|^2 + B(W) \right\}.$$

   - $t_{j+1} = \frac{1+\sqrt{1+4t_j^2}}{2}$.
   - $y_{t+1} = W_j + \frac{t_j - 1}{t_{j+1}}(W_j - W_{j-1})$.

The reconstruction function of the sparse ELM autoencoder remains the same as in equation (6).

From equation (6), it can be seen that $\phi$ is the encoding function with parameters $W_r$ and $b_r$ while the decoding function is linear with learned parameters $W$. Using the reconstruction error as the anomaly score, the above algorithms can be adapted to perform anomaly detection. Let the reconstruction error $e_i$ for data sample $x_i$ be defined as

$$e_i = \|x_i - W^T \phi(W_r^T x_i + b_r)\|^2, \qquad i = 1, 2, ..N. \tag{8}$$

After ELM training, a set of reconstruction error values $\mathcal{E} = \{e_1, e_2, ..e_N\}$ can be obtained. For anomaly detection, the data having the top largest reconstruction errors (note from equation (8) that the reconstruction errors are positive) are identified as anomalies. As in any unsupervised anomaly detection such as one-class SVM, a strength of detection is specified that defines the upper bound of outliers in the training data. Here, the percentile $\tau \in [0, 100]$ on the reconstruction error distribution is considered as the strength of detection. A threshold on reconstruction error $e_\delta$ is computed corresponding to the $\tau$-th percentile as follows

$$e_\delta = percentile(E_{tr}, \tau), \tag{9}$$

where $E_{tr}$ contains the ordered values of $\mathcal{E}$. A data sample $x_i$ is considered an anomaly if

$$e_i = \|x_i - W^T \phi(W_r^T x_i + b_r)\|^2 > e_\delta. \tag{10}$$

It can be noted that if $\tau = 100$, no anomalies are detected while if $\tau = 0$, all the training data are detected as anomalies. Figure 1 shows the idea of using reconstruction error to detect anomalies. The anomaly decision boundary $e_\delta$ (shown by a vertical dotted line) is fixed based on $\tau$ defined in training.

### *B. Embedding Based Anomaly Detection*

Extreme learning machines were recently extended to perform semi-supervised and unsupervised learning tasks [18] using a manifold regularization framework. For anomaly detection, we make use of the unsupervised learning algorithm (US-ELM) where the input data is mapped to a random hyperspace by ELM transformation and spectral embedding is performed. For a clustering task, the data in the embedded space can be clustered using a k-means algorithm. Our hypothesis for anomaly detection is that the ELM embedded space discriminates the anomalies from the nominal data better, and that the anomalies are separated from the origin of the embedded space.

The formulation of US-ELM [18] is given by

$$\min_W \frac{1}{2}\|W\|^2 + \frac{\lambda}{2} \sum_{i,j} a_{i,j} \|f_i - f_j\|^2, \tag{11}$$

where $a_{i,j}$ is the pair-wise similarity between two patterns $x_i$ and $x_j$, $f_i = h(x_i)W$. The second term in the above equation is the manifold regularization term that says if two patterns $x_i$ and $x_j$ are similar, then they are similar in the embedding space as well. Expressing in vector form, the US-ELM is given by

$$\min_W \frac{1}{2}\|W\|^2 + \frac{\lambda}{2} Tr(W^T H^T L H W) \qquad s.t. (HW)^T HW = I_{n_e} \tag{12}$$

where $L$ is the graph Laplacian and $n_e$ the chosen dimension of the embedded space. It was shown in [18] that an optimal solution to the above problem is obtained by choosing $W$ as the matrix whose columns are the normalized eigenvectors corresponding to the first $n_e$ smallest eigenvalues of the below generalized eigenvalue problem

$$(I_{n_h} + \lambda H^T L H)v = \gamma H^T H v. \tag{13}$$

If $v_1, v_2, ..., v_{n_e+1}$ are the eigenvectors corresponding to the $(n_e+1)$ smallest eigenvalues of equation (13), then $W$ is given by

$$W = [\tilde{v}_2, \tilde{v}_3, ..., \tilde{v}_{n_e+1}], \tag{14}$$

where $\tilde{v}_i = v_i / \|Hv_i\|, i = 2, ..., (n_e + 1)$ are the normalized eigen vectors. The embedding for any data sample $x_i$ is given by

$$\tilde{x}_i = h(x_i)W. \tag{15}$$

The idea of adapting the US-ELM for anomaly detection is shown in Figure 2. Using ELM, we aim to find an embedding space where the nominal data are separated well from the anomalous data. A simple rule to define a separating boundary is as follows: the embedded data is z-scored to center the data at the origin and have unit variance. The Euclidean norm of the data $\mathcal{D} = \{d_1, d_2, ..d_N\}$ is determined and is considered as the anomaly score. Similar to the autoencoder based anomaly detection models, a strength of detection is specified that defines the upper bound of outliers in the training data. Here, the percentile $\tau \in [0, 100]$ on the distribution of Euclidean norm of the samples is considered as the strength of detection. A threshold $d_\delta$ is computed corresponding to the $\tau$-th percentile as follows

$$d_\delta = percentile(D_{tr}, \tau), \tag{16}$$

where $D_{tr}$ contains the ordered values of $\mathcal{D}$. A data sample $x_i$ is considered an anomaly if

$$\|Z(h(x_i)W)\|^2 > d_\delta, \tag{17}$$

where Z(.) represents the z-scoring function. As we have centered the data to the origin, the Euclidean norm corresponds to the Euclidean distance of the data sample from the origin. Thus, we define a hypersphere of radius $d_\delta$ that encompasses the nominal data in the embedded space. Any data that falls outside the hypersphere is considered an anomaly. Similar to previous discussions, a $\tau$ of 100 doesn't detect any anomalies while if $\tau = 0$, all the training data are detected as anomalies.

## III. Experiments

In this section, we apply the ELM based anomaly detection algorithms to a real aviation problem and evaluate its ability to detect anomalies as well as its training speed. We use MKAD as the baseline algorithm as MKAD is the present state-of-the-art in aviation anomaly detection and has worked extremely well in practice detecting operational significant anomalies significantly better than other methods [1], [22], [23].

The aviation data considered in this work include radar measurements recorded at the Denver Terminal Radar Approach Control Facility (TRACON), collected and stored by the Performance Data Analysis and Reporting System (PDARS) program which is managed by the FAA's Air Traffic Organization Office of System Operations Services. PDARS not only provided trajectory data but also delivered additional capabilities to enhance this research such as runway detection and computing flight separation features. Each data record is in the form of a time series consisting of aircraft trajectories such as latitude, longitude and altitude. In addition, we also consider a separation parameter that measures the distance to the nearest neighbouring aircraft in the space. This feature is included to consider anomalies caused by loss in separation between flights in the air. Although other data such as ground speed, airport data (runway configuration, counts and rates of departure and arrival, total air and taxi delays) and meteorological data were available, we restrict ourselves to the above 4 parameters for this study for which we have ground truth anomaly labels. Our goal is to compare the proposed ELM anomaly detection algorithms against MKAD on a dataset with known anomalies.

For this benchmark study, we setup the data as follows. The training data is a mix of flights landed at Denver (DEN) airport (all runways included, see Figure 3 for the runway orientation at DEN) during the months of March and July 2015. Based on our previous work [23], we identified and validated about 57 anomalous flights with subject matter experts (SME) who found them to be operationally significant during this time period. Each of the operationally significant events were either due to high speed, overshooting their final turn before landing, overtaking other landing aircraft, or a combination. These constitute the anomaly labels. We also labeled an equal number of nominal flights. These two sets of data (about 115 flights in total) constitutes our test data set. Our training data consists of about 43000 flights. We consider times series data representing the final 60 miles of flight before landing, sampled at 1 mile intervals. Thus our training data matrix is sized 240x43000. The data is shown in Figure 4 in multiple views to understand

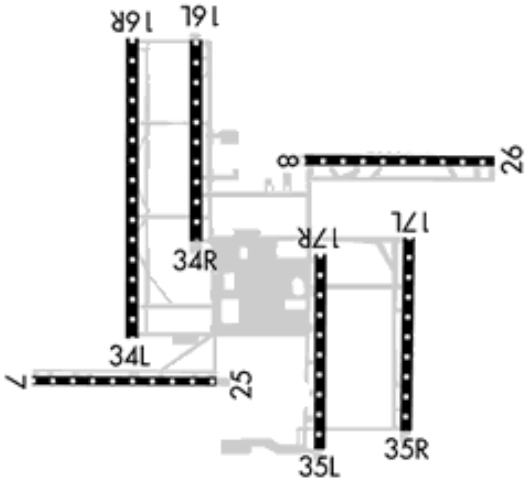


Fig. 3. A sketch of the runways at Denver airport showing the runway numbers and orientation.

the multimodal, high variability patterns in flights landing at DEN airport. The top left plot shows the top view of the airport runways (also see Figure 3) where the flight data begins from the sides of the plot window and converges to the center where the runway area is shown in a rectangular box. The trajectories are colored to distinguish between nominal flights (green) and anomalies (red). A couple of side view plots are also shown to visualize the descent patterns of the flights.

The experiment setup of the baseline MKAD algorithm is as follows: For each of the trajectories which includes latitude, longitude, altitude and separation, the similarity between flights is calculated using the radial basis function

$$\kappa(\vec{x_i}, \vec{x_j}) = exp(-\frac{\|\vec{x_i} - \vec{x_j}\|}{2\sigma^2})$$

where $\sigma$ is chosen by an unsupervised grid search on a subset of the data using the method described in [23]. These similarities are used to build an $N \times N$ kernel matrix for each variable with $N$ equalling the number of training flights. The kernels are then linearly combined using the procedure described in [1], [23] and the below one-class SVM optimization problem is solved using quadratic programming.

$$\min \frac{1}{2} \sum_{i=0}^{N} \sum_{j=0}^{N} \alpha_i \kappa(\vec{x_i}, \vec{x_j}) \alpha_j$$

subject to

$$0 \leq \alpha \leq \frac{1}{N\nu}, \sum_{i=0}^{N} \alpha_i = 1, \ 0 \leq \nu \leq 1,$$

where $\nu$ is a hyperparameter selected to represent the percentage of examples expected to be anomalous. The above problem is solved for optimal $\alpha_i$. The non-zero $\alpha_i$ define the support vectors, which forms the hyperplane that separates the

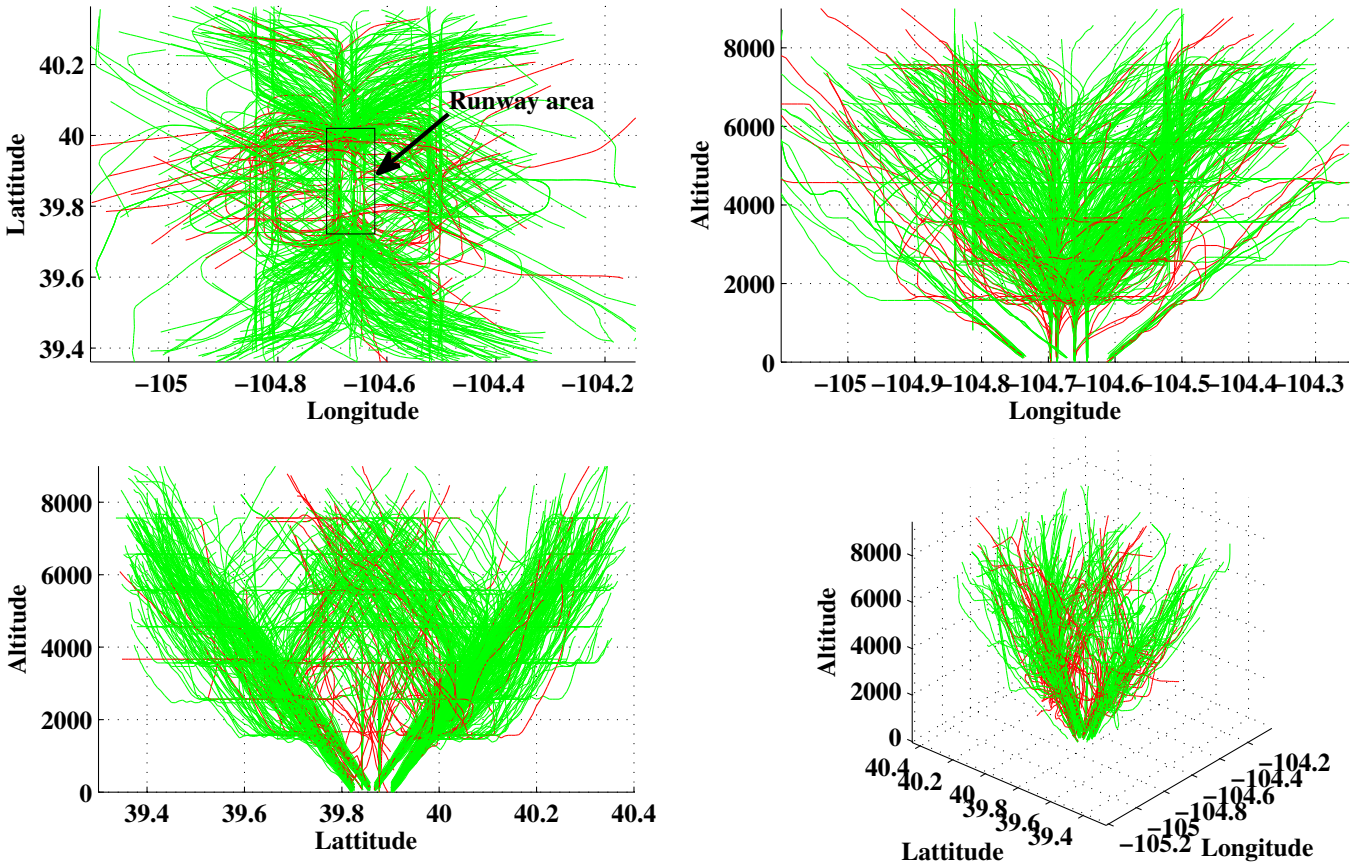


Fig. 4. Flight landing patterns at Denver airport (all runways included, see Fig. 3 for the runway orientation at DEN airport) for the months of March and July 2015. The figure on the top left shows the top view while the figures on top right and bottom left shows two side views. The bottom right figure shows the 3-D view of the flight landing patterns. To help interpret the figures, each trajectory corresponds to a flight which starts from the sides of the plot window and converge to the center (near the runway area) and simultaneously descend for landing. The trajectories colored in red are anomalous flights while the ones in green are nominal flights.

anomalies from the nominals in the training data. Using the SVM model, a data sample $x_i$ is considered anomalous if

$$\sum_{i}^{N} \alpha_i \kappa(x, x_i) - \rho < m_\delta, \tag{18}$$

where $\rho$ is the bias term of the SVM hyperplane, $m_\delta$ is a threshold that can be tuned to achieve a tradeoff between detection accuracy and false alarms.

The ELM experiments are performed as follows. The same training data that is used for MKAD is used to train the models for the three cases - $L_1$-ELMAD, $L_2$-ELMAD and Em-ELMAD. For $L_1$-ELMAD and $L_2$-ELMAD, we used 1000 and 900 hidden nodes respectively while for Em-ELMAD, we used 2000 nodes and 1000 dimensions for the embedded space. The regularization coefficients for the models include $\lambda_2 = 1E-2$, $\lambda_1 = 1E-5$, $\lambda = 4$ and 4 nearest neighbors for calculating the graph Laplacian (in equation (12)). The FISTA algorithm for $L_1$-ELMAD is run for 50 iterations. In all ELM models, the sigmoidal activation function is used. For each model including MKAD, the decision threshold is varied to obtain a receiver-operating characteristic (ROC) curve that shows the variation of true positive rate (detection accuracy) with respect to false positive rate (false alarm rate). The ROC is a popular way to compare the performance of anomaly detection algorithms. An algorithm is said to be better if the ROC curve approaches the top left corner of the plot where it detects all anomalies with no false alarms. It can be seen from Figure 5 that the ROC of the ELM based algorithms are comparable to that of MKAD. Further, it can be observed that $L_2$-ELMAD and Em-ELMAD have a better detection rates with zero false alarms compared to MKAD, while $L_1$-ELMAD is worse. However, the ELM algorithms does not detect all anomalies without a very high false alarm rate (more than 70%) compared to MKAD which detects all anomalies with about 50% false alarms.

To quantitatively compare the algorithms, we use the area under the ROC curve (AUC) that gives a single value summarizing the ROC curve. A high value of AUC indicates a better algorithm. The AUC values of the ELM algorithms along with that of MKAD are summarized in Table I. The training and testing times for the different algorithms, and reconstruction errors for the autoencoder based models are

TABLE I

PERFORMANCE COMPARISON OF ELM ANOMALY DETECTION ALGORITHMS WITH MKAD SHOWING AREA UNDER THE ROC CURVE (AUC), TRAINING AND TESTING TIME, RECONSTRUCTION ERRORS FOR AUTOENCODER MODELS. A HIGHER VALUE OF AUC INDICATES A BETTER ALGORITHM.

| | Training time (seconds) | Testing time (seconds) | Average training reconstruction error | Average testing reconstruction error | AUC |
|---|---|---|---|---|---|
| MKAD | 680.053 | 31.575 | - | - | 0.8922 |
| $L_1$-ELMAD | 8.4629 | 0.1058 | 0.0016 | 0.0058 | 0.8513 |
| Em-ELMAD | 49.33 | 0.2646 | - | - | 0.8912 |
| $L_2$-ELMAD | **2.0153** | **0.0269** | 0.0005529 | 0.0009264 | **0.9105** |

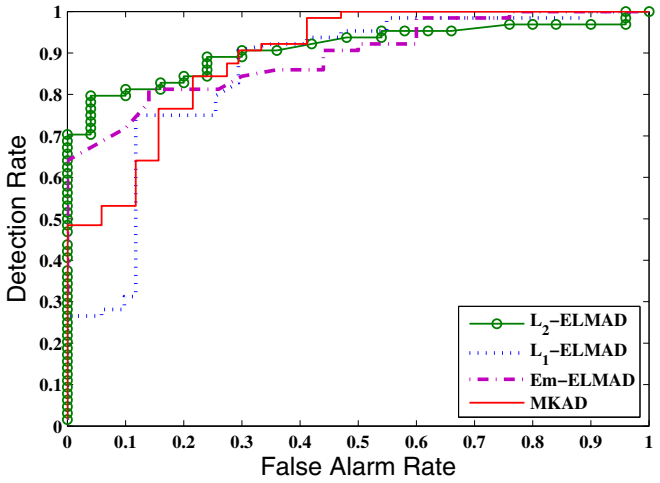


Fig. 5. Comparison of receiver-operating characteristic (ROC) curves of the ELM based algorithms with the baseline MKAD algorithm.

also shown in Table I. We observe that the AUC of the ELM based algorithms are comparable[1] to that of MKAD with $L_2$-ELMAD slightly better. Further, the training of ELM based algorithms are much faster[2] compared to that of MKAD. The Em-ELMAD is about 13 times faster and $L_1$-ELMAD is about 80 times faster while $L_2$-ELMAD is about 330 times faster compared to MKAD's training. Clearly, the ELM's random feature mapping helps achieve a high accuracy with a significantly low (about 2 orders of magnitude less) training times.

We note that MKAD's real benefit lies in the fact that it can handle heterogenous data sources such as continuous, discrete sequences, text etc., very elegantly by having a specialized kernel for each data source. However, it is not difficult to observe that MKAD's slowness is not only because of the kernel building step but also because of solving the resulting quadratic programming problem. In our present version of ELM, we use only the random kernel (the random feature mapping of the input data) and that gives us good results. However, if need be, any of MKAD's kernels may be used with ELM models [10]. By doing so, the time consuming kernel building step may be present but ELM solves the resulting optimization problem more efficiently and would still be faster than MKAD.

[1]In this paper, we only report the AUC comparison of the algorithms. To understand more about the type of anomalies that we found in this work, the reader is referred to [23].

[2]All the experiments are performed on Intel(R) Core(TM) i7-2600 CPU @ 3.40 GHz with 16GB of RAM running on 64-bit operating system.

## IV. CONCLUSIONS

In summary, ELM based anomaly detection is explored and three extensions are proposed namely the $L_1$-ELMAD, $L_2$-ELMAD and Em-ELMAD. The algorithms are tested on a benchmark aviation safety problem where the results of ELM looks promising. The $L_2$-ELMAD algorithm outperforms MKAD in detection accuracy and is more than 300 times faster in training.

Although the proposed algorithms perform well on the considered aviation problem, more work needs to be done to benchmark the performance on other data sets as well as with respect to other popular state-of-the-art algorithms and will constitute our work for the future.

## REFERENCES


[1] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "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. New York, NY, USA: ACM, 2010, pp. 47–56.

[2] D. Smallen, "Summary 2014 U.S.-Based Airline Traffic Data," *U.S. Department of Transportations Bureau of Transportation Statistics (BTS)*, 2015. [Online]. Available: http://www.rita.dot.gov/bts/press_releases/bts015_15

[3] "FAA Nextgen Implementation Plan 2014," 2014. [Online]. Available: www.faa.gov/nextgen/library/media/nextgen_implementation_plan_2014.pdf

[4] S. Das, 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.

[5] T. R. Chidester, "Understanding normal and atypical operations through analysis of flight data," in *Proceedings of the 12th International Symposium on Aviation Psychology*, Dayton, OH, 2003.

[6] D. L. Iverson, "Inductive system health monitoring," in *Proceedings of the 2004 International Conference on Artificial Intelligence (IC-AI'04)*, Las Vegas, NV, 2004.

[7] S. D. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03. New York, NY, USA: ACM, 2003, pp. 29–38.

[8] S. Budalakoti, S. Budalakoti, A. Srivastava, M. Otey, and M. Otey, "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 1, pp. 101–113, Jan 2009.

[9] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*. ACM, 2013, pp. 8–15.

[10] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32 – 48, 2015.

[11] Y. Wang, D. Li, Y. Du, and Z. Pan, "Anomaly detection in traffic using l1-norm minimization extreme learning machine," *Neurocomputing*, vol. 149, Part A, pp. 415 – 425, 2015.

[12] R. Singh, H. Kumar, and R. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8609 – 8624, 2015.

[13] C. Cai, H. Pan, and G. Cheng, "Fusion of bvm and elm for anomaly detection in computer networks," in *Computer Science Service System (CSSS), 2012 International Conference on*, Aug 2012, pp. 1957–1960.

[14] V. M. Janakiraman, X. Nguyen, J. Sterniak, and D. Assanis, "Identification of the Dynamic Operating Envelope of HCCI Engines Using Class Imbalance Learning," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 98–112, Jan 2015.

[15] V. Ravi and P. Singh, "Auto-associative extreme learning factory as a single class classifier," in *Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on*, Dec 2014, pp. 1–6.

[16] C. Gautam and A. Tiwari, "On the construction of extreme learning machine for one class classifier," in *Proceedings of ELM-2015 Volume 1*, ser. Proceedings in Adaptation, Learning and Optimization, J. Cao, K. Mao, J. Wu, and A. Lendasse, Eds. Springer International Publishing, 2016, vol. 6, pp. 447–461.

[17] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[18] G. Huang, S. Song, J. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *Cybernetics, IEEE Transactions on*, vol. 44, no. 12, pp. 2405–2417, Dec 2014.

[19] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," *Unsupervised and Transfer Learning Challenges in Machine Learning, Volume 7*, p. 43, 2012.

[20] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1096–1103.

[21] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[22] S. Das, 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.

[23] B. Matthews, D. Nielsen, J. Schade, K. Chan, and M. Kiniry, "Automated discovery of flight track anomalies," in *Digital Avionics Systems Conference (DASC), 2014 IEEE/AIAA 33rd*, Oct 2014, pp. 4B3–1–4B3–15.