# Detecting Anomalous Data Using Auto-Encoders

Jerone T. A. Andrews, Edward J. Morton, and Lewis D. Griffin

***Abstract*—The most general mode of detecting anomalous data would make no assumptions regarding them other than their atypicality. For such a system, choosing features, to best support the detection, is problematic. The hidden layer representation of auto-encoder artificial neural networks is a potential uncommitted solution to this. We have assessed these features on a range of problems derived from two image datasets, feeding the features into one-class Radial Basis Function (RBF) $\nu$-Support Vector Machine (SVM) classifiers. Our range of problems vary in diversity of the normal and anomalous classes. Assessed across the range, we find the best performing feature to be a late fusion of hidden layer activations, residual error vectors and the raw input signals. This improves upon the use of auto-encoder residual vector error magnitude, which has previously been proposed for anomaly detection.**

***Index Terms*—Anomaly detection, auto-encoders, one-class support vector machines.**

## I. Introduction

Anomalous data are loosely defined as data items that are atypical relative to a normal class. However, the specific scenario can influence which aspects of the data are considered in making an evaluation of atypicality. Consider, for example, a thermal imaging system used in a mass transportation-screening context. We may wish to detect people with abnormal body temperature patterns for disease control, or we may wish to detect abnormal temperature patterns across the body indicating concealed threats, or we may wish to detect people with any kind of abnormality as a way to direct visual inspection efficiently. Here we are concerned with this third type of ‘neutral’ anomaly detection.

Many of the successes in pattern recognition and classification are feature dependent, thus much effort is placed on the engineering of representative features. However, these approaches cannot be used when one only has normal data. Instead, we propose to use features that are directly learnt from the data without the need for labels. Within the field of unsupervised feature learning, there are two distinct routes one may follow: the first lies in probabilistic graphical models, and the second in computational models, namely neural networks [1]. The probabilistic approach requires a prior on the data distribution, while the neural network does not; for this reason, we consider the neural network approach.

Current solutions, fitting this description, typically employ an unsupervised auto-encoder feed-forward artificial neural network approach [2]-[9] and make anomaly assessments based on the magnitude of the residual vector (the difference between the network's input and output), as a feature to be compared to a threshold. A single-layered auto-encoder is composed of input and output layers of equal cardinality, and a hidden layer that attempts to recreate the inputs such that the outputs resemble the inputs. The justification for this use of auto-encoders, for anomaly detection, is that an auto-encoder trained solely on normal samples should find it difficult to reconstruct the input signal of an anomalous sample, hence yielding a large residual magnitude to such configurations, since one expects the latent characteristics of anomalous samples to deviate from that of normal samples. In our work, we only use auto-encoders with a single hidden layer. We do this in order to reduce the computational expense of our system, since ‘deeper’ networks tend to be prohibitive [10] due to the need for pre-training of each hidden layer separately. Furthermore, there is the issue of the non-trivial choice of network architecture selection which requires hyperparameter optimisation over a sizeable space of possible configurations, where the number of possible configurations grows exponentially as we increase the number of hidden layers. Moreover, it is shown in [11] that it is possible to attain state-of-the-art performance, in CIFAR-10 and NORB vision tasks, using only ‘shallow’ single-layered networks. In addition, empirical evidence is provided in [12] which shows shallower networks achieving recognition rates that are competitive with their deeper counterparts.

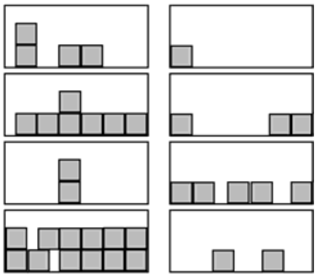

Fig. 1. High complexity images (left), and low complexity images (right).

Now, imagine an image with a small residual magnitude, without more qualification this may not be enough evidence that the image is normal. If for example, our training set is composed of images with high complexity, then it is possible that the auto-encoder will find it easy to reconstruct images that are anomalous in being of low complexity (see

Manuscript received October 26, 2015; revised January 12, 2016. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under CASE Award Grant 157760 and Rapiscan Systems Ltd.

J. T. A. Andrews is with the Department of Computer Science, Department of Statistical Science, and Security Science Doctoral Training Centre, University College London, 66-72 Gower Street, WC1E 6BT, London, UK (e-mail: jerone.andrews@cs.ucl.ac.uk).

E. J. Morton is with Rapiscan Systems Ltd., 2805 Columbia St, Torrance, CA 90503 USA (e-mail: emorton@rapiscansystems.com).

L. D. Griffin is with the Department of Computer Science, University College London, 66-72 Gower Street, WC1E 6BT, London, UK (e-mail: l.griffin@cs.ucl.ac.uk).

doi: 10.18178/ijmlc.2016.6.1.565

Fig. 1). That is, the magnitude of the residual vector of low complexity images may be within the range for normal samples.

A criticism of making anomaly assessments based solely on the magnitude of the residual vector is the fact that the residuals of each feature are summed to give a single value. This may be an imprudent thing to do, since we are disregarding which feature dimensions contribute to the magnitude of the residual vector, and residuals in some features may be more anomalous than in others. What is more, current approaches of this type fail to make use of the hidden representation, which has the potential to be a competent feature vector in itself since it is capable of learning first-order (low-level) features. Rather, we could instead employ a classifier in place of a threshold, such that we do not need to transform the residual vector into a single value. Moreover, this allows us to use the input signal vector, hidden representation, full residual vector, and their combinations. Our hypotheses are thus:

1) Using only the magnitude of the residual vector is under-utilising it for the task of anomaly detection: the full vector will give improved performance.
2) The hidden layer representations of an auto-encoder are effective ‘neutral’ features for anomaly detection.
3) Some anomalies may have abnormal hidden representations but normal residuals.

To test our hypotheses, we formed several test problems using two different datasets: low-resolution X-ray images of freight containers that may be empty or containing cargo, and MNIST handwritten digits [13]. For each of the datasets we constructed a range of test problems with combinations of tight and diverse, normal and anomaly classes. In all cases, we constructed the anomaly detection system on the basis of the normal class only. Our system does make minimal use of an anomaly set during classifier construction: we use it to choose a small number of hyperparameters. This is a weakness that we intend to remove in future work, but we note that it is quite different from using an anomaly set to choose features or a two-class feature space boundary.

Our results empirically support our hypotheses: employing the full residual vector gives an improved recognition rate over using the residual magnitude; the hidden representation has benefits above using the input signals and residuals; and the hidden representation together with the residuals and input signal is superior to any of those constituents alone. In addition, we compared a post-classifier fusion technique and pre-classifier feature vector concatenation. In the former, we built three separate classifier models trained on the input signals, hidden representations, and full residual vectors, and then combine their label outputs, for each sample, in order to give an anomaly score. Whereas in the latter, we concatenated the input signals, hidden representations, and full residual vectors to form new feature vector representations for classifier model construction. The results show the post-classifier fusion technique to be a workable alternative, attaining higher recognition rates, particularly when compared to pre-classifier feature vector concatenation.

## II. Datasets

We used two datasets (see Fig. 2) to assess our anomaly detection system: low-resolution X-ray images of freight containers that may be empty or containing cargo; and MNIST handwritten digits. For each dataset we constructed a range of test problems with combinations of tight and diverse, normal and anomaly classes. In all cases we constructed the anomaly detection system on the basis of the normal class only.

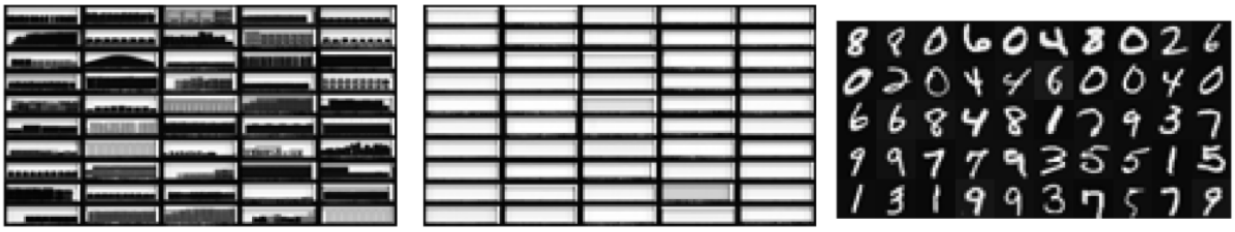

Fig. 2. Some typical samples from the two datasets: X-ray transmission images of non-empty freight containers (left), X-ray transmission images of empty freight containers (centre), and MNIST handwritten digits (right).

### A. X-Ray Transmission Images of Freight Containers

This dataset consists of X-ray transmission images of freight containers, obtained from a Rapiscan Eagle ®R60 rail car scanner. The scanner images individual rail cars moving at speeds of up to 60km/h with 5.6mm pixels. The dataset consists of 2560 images of freight containers containing cargo (non-empty) and 2560 images of freight containers containing no cargo (empty). All images vary due to small differences in freight containers and their furniture, while cargo images also vary in the cargo. We down-sampled the images, for ease of experiment, to 32×9 pixels.

### B. MNIST Handwritten Digits

This widely used dataset consists of handwritten digits 0 through 9, centred and scaled to similar size. We use the full dataset of 70000 samples. Images were resized to 14×14 pixels.

### C. Anomaly Detection Test Problems

Our six anomaly detection test problems derived from the two datasets were:

1) **Ftd.** Normal class: empty freight containers (*tight*); anomaly class: non-empty freight containers (*diverse*).
2) **Fdt.** Normal class: non-empty freight containers (*diverse*); anomaly class: empty freight containers (*tight*).
3) **Mtt.** Normal class: handwritten digits of {5} (*tight*); anomaly class: handwritten digits of {2} (*tight*).
4) **Mdt.** Normal class: handwritten digits of odds {1, 3, 5, 7, 9} (*diverse*); anomaly class: handwritten digits of {2}

(*tight*).

5) **Mtd.** Normal class: handwritten digits of {5} (*tight*); anomaly class: handwritten digits of evens {0, 2, 4, 6, 8} (*diverse*).
6) **Mdd.** Normal class: handwritten digits of odds {1, 3, 5, 7, 9} (*diverse*); anomaly class: handwritten digits of evens {0, 2, 4, 6, 8} (*diverse*).

For each of these test problems, we sampled without replacement the following sets:

- **Training set.** $2048$ images from the normal class.
- **Validation set.** $2048$ images from the anomaly class.
- **Testing set.** $512$ images from the normal class and $512$ images from the anomaly class.

## III. Anomaly Detection Framework

In this section, we will describe our anomaly detection framework. Although our approach is quite general we will focus on components suitable for images.

Our system performs the following steps given a set of $p$ training images $\mathbf{x}_1,\ldots,\mathbf{x}_p$:

1) Learn a feature encoding, of the training images, using an unsupervised sparse feed-forward neural network auto-encoder.
2) Extract features for each image using the trained sparse auto-encoder.
3) Train a one-class non-linear Radial Basis Function $\nu$-Support Vector Machine (RBF SVM) [14], [15] classifier to predict the label, normal or anomalous, given the computed features.

Next, we will go on to describing the steps of our system in finer detail.

### A. Unsupervised Sparse Auto-Encoder Feature Learning

A single-layered auto-encoder is a type of feed-forward artificial neural network with one hidden layer. An auto-encoder is trained to reconstruct its input signal by finding useful features from the input space. The auto-encoder learns a map from input to representation, where the representation consists of the activations of the $m$ hidden layer units. Concretely, given an input $\mathbf{x} \in R^n$, the auto-encoder computes an output $\mathbf{y} \in R^n$, via a hidden layer representation $f \in R^m$. The hidden layer activations are computed from the input according to $f(\mathbf{x}) = g(\mathbf{W}_1\mathbf{x}+\mathbf{b}_1)$, and the output layer from the hidden layer according to $\mathbf{y} = g(\mathbf{W}_2 f(\mathbf{x})+\mathbf{b}_2)$. Where $\mathbf{W}_1 \in R^{m\times n}$ and $\mathbf{W}_2 \in R^{n\times m}$ are weight matrices, $\mathbf{b}_1 \in R^m$ and $\mathbf{b}_2 \in R^n$ are bias vectors, and $g(\mathbf{z}) = 1/(1+\exp(-\mathbf{z}))$ is our chosen activation function applied to the vector $\mathbf{z}$ component-wise.

Auto-encoders apply back-propagation, for training $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1$, and $\mathbf{b}_2$, by means of gradient descent, in an attempt to achieve $\mathbf{y} \approx \mathbf{x}$ for the training data. If the hidden layer has dimensionality not less than the input and output layers ( $m \ge n$ ) then it is trivial for the auto-encoder to succeed in exactly reproducing its inputs, and nothing of use is learnt. However, if the hidden layer activations are encouraged to be sparse then its units are forced to learn significant structures within the data. Imposing sparsity, we have a minimisation problem of the form:

$$\text{minimise}\left\{\sum_{i=1}^{p}\|\mathbf{x}_i - \mathbf{y}_i\|^2 + \beta\sum_{j=1}^{m}\mathrm{KL}\left(\rho\|\hat{\rho}_j\right)\right\}, \tag{1}$$

where $\mathrm{KL}\left(\rho\|\hat{\rho}_j\right) = \rho\ln\left(\rho/\hat{\rho}_j\right) + \left(1-\rho\right)\ln\left(\left(1-\rho\right)/\left(1-\hat{\rho}_j\right)\right)$ is our sparsity penalty term, $\hat{\rho}_j$ is the average activation over the whole training set for hidden unit $j$, $\rho$ is our (desired) sparsity level parameter which we constrain $\hat{\rho}_j$ to approximate, and $\beta$ is used to control the weight of the sparsity penalty term.

TABLE I: Sparse Auto-Encoder Hyperparameter Values

| | Ftd | Fdt | Mtt | Mdt | Mtd | Mdd |
|---|---|---|---|---|---|---|
| $m$ | 576 | 144 | 392 | 392 | 98 | 98 |
| $\rho$ | 0.1 | 0.5 | 0.5 | 0.5 | 0.25 | 0.25 |
| $\beta$ | 4 | 16 | 4 | 4 | 4 | 4 |
| $\lambda$ | $10^{-1}$ | $10^{-3}$ | $10^{-5}$ | $10^{-5}$ | $10^{-5}$ | $10^{-5}$ |

Evidently, there are several hyperparameters associated with the simple single-layered sparse auto-encoders we employ: $m$ (number of units in the hidden layer, excluding the bias unit), $\lambda$ (weight decay) used during back-propagation, $\rho$ (sparsity level), and $\beta$ (weight of sparsity penalty term), which were all set to reasonable values based on preliminary results for each test problem. For reproducibility our chosen set of hyperparameter values, $\{m,\rho,\beta,\lambda\}$, are listed in Table I for each of the six anomaly detection test problems.

### B. Features

We use our sparse auto-encoder to compute several types of features for use in anomaly detection:

$$\text{Input (INP): } \mathbf{x}, \tag{2}$$

$$\text{Hidden Representation (HR): } f(\mathbf{x}), \tag{3}$$

$$\text{Scalar Residual Magnitude (SRM): } \|\mathbf{x}-\mathbf{y}\|_1, \tag{4}$$

$$\text{Signed Residual (R): } \mathbf{x}-\mathbf{y}, \tag{5}$$

$$\text{Absolute Residual (AR): } |\mathbf{x}-\mathbf{y}|, \tag{6}$$

$$\text{Squared Residual (SR): } (\mathbf{x}-\mathbf{y})^2, \tag{7}$$

$$\text{Normalised Signed Residual (NR): } (\mathbf{x}-\mathbf{y})/\sigma, \text{ and} \tag{8}$$

$$\text{Normalised Squared Residual (NSR): } (\mathbf{x}-\mathbf{y})^2/\sigma^2, \tag{9}$$

where $\sigma = \sqrt{(1/p)\sum_{k=1}^{p}(\mathbf{x}_k - \mathbf{y}_k)^2}$ is the vector of root-mean-square residuals between input and output across the training set. All of the features set out above are vectors of dimension $m$, except for the scalar residual magnitude feature.

### C. One-Class RBF SVM Classification

Consider our set of training samples $\mathbf{x}_1,\ldots,\mathbf{x}_p$ and suppose that these samples are drawn from a probability distribution $P$, in the feature space. We apply a non-linear one-class classification algorithm, namely a Radial Basis Function $\nu$ -Support Vector Machine, [14], [15] in an attempt to estimate the support of this distribution.

This one-class formulation, of the standard two-class SVM procedure, first transforms the feature vector via a non-linear RBF kernel, where the origin is viewed as the sole member of the unknown second class. The one-class SVM gives a function $h$ that outputs $+1$ in a region that encompasses most of the training samples, and outputs $-1$ everywhere else.

The objective function to separate the training samples from the origin of our one-class RBF SVM classifier is the following quadratic programming minimisation task:

$$\min_{\omega,\xi_i,q}\frac{1}{2}\|\omega\|^2+\frac{1}{\nu p}\sum_{i=1}^{p}\xi_i-q \tag{10}$$

subject to:

$$\left(\omega\cdot\phi(\mathbf{x}_i)\right)\geq q-\xi_i, \forall i=1,\ldots,p, \tag{11}$$

$$\xi_i\geq 0, \forall i=1,\ldots,p, \tag{12}$$

where $\phi$ is a kernel mapping of $\mathbf{x}_i$ into a dot product space $F$, $q$ is a bias term, and $\nu\in(0,1)$ is an upper bound on the fraction of the training samples that are considered to be out-of-class and a lower bound on the fraction of training samples used as Support Vectors (SVs).

Our decision rule, having solved the quadratic programming minimisation problem using Lagrange multipliers, is:

$$h(\mathbf{x})=\operatorname{sgn}\left(\sum_{i=1}^{p}\alpha_i\cdot K(\mathbf{x},\mathbf{x}_i)-q\right), \tag{13}$$

where the $\alpha_i$ are the Lagrange multipliers, and

$$K(\mathbf{x}_i,\mathbf{x}_j)=\exp\left(-\gamma\cdot\|\mathbf{x}_i-\mathbf{x}_j\|^2\right),\gamma>0, \tag{14}$$

is the RBF kernel with width parameter $\gamma$.

There is no clear means in which to differentiate between alternative one-class RBF SVM models, in the case of classification accuracy ties, when testing on the validation set. Therefore, we employ an ensemble composed of the winning models and take the majority output as the class label. Formally, our procedure for selecting hyperparameter values for $\nu$ and $\gamma$ is as follows:

1) Sample without replacement $1024$ normal samples from the training set.
2) Sample $512$ normal samples from the training set and $512$ anomalous samples from the validation set.
3) Perform a grid-search over the hyperparameters, and then select all (in the case of ties) hyperparameter tuples that give rise to the highest classification accuracy, for the samples in Step 2.
4) Go back to Step 1 and repeat this process twice more.

We now have a set of hyperparameter tuples $\{\nu,\gamma\}$. We will train an ensemble of one-class RBF SVMs (using the set of tuples $\{\nu,\gamma\}$), on the entire training set of $2048$ samples, such that they are now ready to predict the labels of unseen test samples. We combine the label outputs of each one-class RBF SVM, in the ensemble, by taking the majority output as the predicted label of an unseen test sample.

To be clear, we have made use of an anomaly set in order to select hyperparameter values for our one-class RBF SVMs. We emphasise that this is a deficiency within our system, however we do not use the anomaly set to choose features or a two-class feature space boundary. We aim to remove the need for anomalous samples, for classifier hyperparameter optimisation, in future work.

For this study, we utilised the readily available LIBSVM toolbox (Version 3.20) [16], which is an integrated piece of software with built-in distribution estimation (one-class SVM).

## IV. Experiments and Analysis

Using our anomaly detection framework outlined in Section III, the experiments that are reported in this paper are as follows:

1) We began by performing a comparison of the features, (2)-(9), across all six test problems specified in Section III.A. From this it will be possible to evaluate which of the features are better suited as representations for the concept of normality. Furthermore, we may assess whether anomalous samples, do in some cases, give rise to abnormal auto-encoder features but normal residuals, whilst others have abnormal residuals but normal auto-encoder features.
2) Lastly, we performed a comparison of two different ways of combining the best performing feature vectors, namely:
   - *Pre-classifier feature vector concatenation* where we combined the best feature vectors into a single new feature vector.
   - *Post-classifier fusion* whereby a different sparse auto-encoder is trained on each of the chosen feature vectors separately and then we combined the results of the various one-class RBF SVMs to determine whether a sample is anomalous or not.

### A. Comparison of Features

Our experiments first considered the usefulness of the features: (2)-(9). Table II shows the classification accuracy of each ensemble of one-class RBF SVMs, across the six test problems, on the unseen testing sets described in Section II.C. There are several notable observations that can be taken from Table II:

1) The hidden representation is on average the best

performing feature vector across all six test problems, and most notably it outperforms the raw input signal.

2) The best performing residual is the normalised signed residual vector, most importantly showing itself to be better than the scalar residual magnitude feature.
3) The normalised signed residual vector has superiority over the hidden representation when the normal class is tight or the anomaly class is diverse, exemplified in the comparison of their scores for Ftd with Fdt. For Ftd the hidden representation scores $94.43\%$ while the normalised signed residual vector scores $98.24\%$; for Fdt the scores are $92.99\%$ and $73.34\%$, respectively.

for comparison of features

### B. Feature Vector Combination

Having established that input features (2), hidden layer features (3), and reconstruction errors—encoded as normalised signed residuals (8)—are all sometimes effective, but each on different problems, we then considered the most effective means of combining them. We compared pre-classifier feature vector concatenation and post-classifier fusion using a combination of these three features.

#### 1) Pre-Classifier Feature Vector Concatenation (PRE)

1) Create combined feature vectors for the training data comprised of the features derived from (2), (3), and (8) by concatenation.
2) Scale feature dimensions so that each has zero-mean and unit-variance across the training set.
3) Retrieve the predicted class labels for the testing set given by the ensemble of one-class RBF SVMs trained on normal data, with the combined feature vectors, and take the mode of the predictions, for each sample, as the label.

#### 2) Post-Classifier Fusion (POST):

1) Retrieve the predicted class labels for the testing set given by the ensemble of one-class RBF SVMs trained on normal data with feature vectors of the form (2). Then compute the mean label, for each sample, given the predictions from the ensemble.
2) Repeat Step 1, but this time for feature vectors of the form (3).
3) Repeat Step 1, but this time for feature vectors of the form (8).
4) Compute the mean of the labels given by Steps 1--3, then label a sample as normal if the output is positive, otherwise label as anomalous.

The results in Table III show post-classifier fusion to be superior to using any of the stand-alone feature vectors: (2), (3), or (8). In contrast, pre-classifier feature vector concatenation is on average the worst performing approach, receiving the lowest classification accuracy.

TABLE II: TEST CLASSIFICATION ACCURACY (%) ON THE TEST PROBLEMS (BOLD INDICATES THE HIGHEST ACCURACY WITHIN A ROW)

| Test Problem | INP | HR | SRM | R | AR | SR | NR | NSR |
|---|---|---|---|---|---|---|---|---|
| Ftd | 98.24 | 94.43 | 95.02 | 98.24 | 98.93 | 98.54 | 98.24 | **99.22** |
| Fdt | 80.37 | **92.99** | 81.15 | 55.08 | 48.14 | 50.78 | 73.34 | 48.44 |
| Mtt | 91.11 | 91.21 | 84.96 | 86.91 | 88.77 | 83.59 | **91.80** | 90.43 |
| Mdt | **86.23** | 85.06 | 70.70 | 79.69 | 81.35 | 78.71 | 83.40 | 80.08 |
| Mtd | 85.74 | **88.77** | 87.60 | 50.00 | 50.00 | 86.91 | 86.33 | 83.01 |
| Mdd | 78.52 | 75.29 | **80.66** | 50.00 | 50.00 | 70.12 | 75.59 | 74.80 |
| Average | 86.70 | **87.96** | 83.35 | 69.99 | 69.53 | 78.11 | 84.78 | 79.33 |

TABLE III: TEST CLASSIFICATION ACCURACY (%) ON THE TEST PROBLEMS (BOLD INDICATES THE HIGHEST CLASSIFICATION ACCURACY WITHIN A ROW) FOR FEATURE VECTOR COMBINATION

| Test Problem | INP | HR | SRM | NR | PRE | POST |
|---|---|---|---|---|---|---|
| Ftd | **98.24** | 94.43 | 95.02 | **98.24** | **98.24** | 97.66 |
| Fdt | 80.37 | **92.99** | 81.15 | 73.34 | 51.86 | 89.55 |
| Mtt | 91.11 | 91.21 | 84.96 | 91.80 | 92.19 | **92.29** |
| Mdt | 86.23 | 85.06 | 70.70 | 83.40 | 82.91 | **87.01** |
| Mtd | 85.74 | 88.77 | 87.60 | 86.33 | 86.82 | **89.75** |
| Mdd | **78.52** | 75.29 | 80.66 | 75.59 | 72.95 | 78.22 |
| Average | 86.70 | 87.96 | 83.35 | 84.78 | 80.83 | **89.08** |

## V. SUMMARY

In this empirical study, on the detection of anomalous data using auto-encoders, we have conducted several experiments on a range of anomaly detection test problems. Our experiments compared a selection of different feature vectors derived from a sparse auto-encoder feed-forward neural network. The empirical results appear to support our hypothesis that there is indeed a better way to use residual errors than simply computing the magnitude, and this is most apparent when the normalised signed residual is employed. Furthermore, the results suggest that the hidden layer representation, as a stand-alone feature vector, is more than capable of characterising the fundamental attributes of normality. Its competence is shown through it having the highest average recognition rate amongst the stand-alone feature vectors. The robustness of the hidden layer representation is best illustrated in situations where the auto-encoder does not struggle to reconstruct anomalous images, and so gives rise to low residuals. For instance, in Fdt, where we have a diverse normal class of high-complexity non-empty cargo containers and an anomaly class of empty cargo containers. In this test problem the auto-encoder is able to reconstruct the low-complexity empty cargo containers, despite having never been trained to do so. However, the units of the hidden representation are activated in an abnormal fashion, and as such we are able to identify a greater number of anomalous images, as opposed to using the residuals to make a prediction. This differs from the converse test problem, Ftd, where the class labels have been swapped. We see in this case a performance increase across all the feature vectors, since the auto-encoder finds it difficult to encode a more-complex non-empty cargo container image having been trained on empty cargo containers, which are relatively homogeneous in appearance. Nonetheless, the use of the normalised signed residual, the hidden representation and the input signal work effectively when all three are used in tandem and combined using a post-classifier fusion. By doing this, we are able to benefit from their individual anomaly detection capabilities, that is, some anomalies may only be apparent in one of those constituents.

Our aim in this work was to consider the problem of anomaly detection without the knowledge of the anomaly class. We note that we deviated from this aim at one stage: we make use of an anomaly validation set to select hyperparameters for one-class RBF SVMs. Whilst the use of a validation set is not ideal, it is a component we aim to remove in future work so as to move towards a truly unsupervised system.

## Acknowledgment

J. T. A. Andrews thanks Emma F. Shapiro for her invaluable support and assistance in reviewing this work.

## References

[1] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798-1828, March 2013.

[2] N. Japkowicz, C. Myers, and M. Gluck, “A novelty detection approach to classification,” in *Proc. International Joint Conference on Artificial Intelligence*, 1995, pp. 518-523.

[3] J. M. Ko, J. M., Y. Q. Ni, J. Y. Wang, Z. G. Sun, and X. T. Zhou, “Studies of vibration-based damage detection of three cable-supported bridges in Hong Kong,” in *Proc. International Conference on Engineering and Technological Sciences*, *China*, 2000, pp. 105-112.

[4] L. Manevitz and M. Yousef, “One-class document classification via neural networks,” *Neurocomputing*, vol. 70, no. 7, pp. 1466-1481, 2007.

[5] H. Sohn, K. Worden, and C. R. Farrar, “Novelty detection under changing environmental conditions,” in *Proc. SPIE 8th Annual International Symposium on Smart Structures and Materials*, 2001, pp. 108-118.

[6] R. J. Streifel, R. J. Marks II, M. A. El-Sharkawi, and I. Kerszenbaum, “Detection of shorted-turns in the field winding of turbine-generator rotors using novelty detectors-development and field test,” *IEEE Transactions on Energy Conversion*, vol. 11, no. 2, pp. 312-317, 1996.

[7] C. Surace, K. Worden, and G. Tomlinson, “A novelty detection approach to diagnose damage in a cracked beam,” *Proceedings of SPIE*, vol. 3089, pp. 947-953, 1997.

[8] C. Surace and K. Worden, “A novelty detection method to diagnose damage in structures: an application to an offshore platform,” in *Proc. Eighth International Conference of Off-shore and Polar Engineering*, 1998, vol. 4, pp. 64–70.

[9] K. Worden, “Structural fault detection using a novelty measure,” *Journal of Sound and Vibration*, vol. 201, no. 1, pp. 85-101, 1997.

[10] A. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, “On random weights and unsupervised feature learning,” in *Proc. the 28th International Conference on Machine Learning*, 2011, pp. 1089-1096.

[11] A. Coates, A. Y. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in *Proc. International Conference on Artificial Intelligence and Statistics*, 2011, pp. 215-223.

[12] J. Ba and R. Caruana, “Do deep nets really need to be deep?” *Advances in Neural Information Processing Systems*, pp. 2654-2662, 2014.

[13] Y. LeCun, C. Cortes, and C. J. C. Burges. (1998). The MNIST database of handwritten digits. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[14] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, “Support vector method for novelty detection,” *NIPS*, vol. 12, pp. 582-588, 1999.

[15] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural Computation*, vol. 13, no. 7, pp. 1443-1471, 2001.

[16] C. C. Chang, and C. J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27, 2011.

**Jerone T. A. Andrews** has an MSci (Hons) in mathematics from King’s College London, UK and an MRes in security science from University College London, UK. He is currently working towards a PhD in applied mathematics at University College London, UK, jointly supervised by the Departments of Computer Science and Statistical Science. His main topics of interest are anomaly detection in computer vision, similarity learning, and density estimation.