Generative Adversarial Network Based Autoencoder: Application to fault detection problem for closed-loop dynamical systems*

I. Chakraborty[1,◇] **, R. Chakraborty**[2,◇] and **D. Vrabie**[1]
[1]Optimization and Control Group
Pacific Northwest National Laboratory, Richland, WA, USA
[2]CVGMI, University of Florida, FL, USA
[1]{indrasis.chakraborty, draguna.vrabie}@pnnl.gov [2]rudrasischa@gmail.com
◇ These authors contributed equally to this work

Abstract

The fault detection problem for closed-loop, uncertain dynamical systems is investigated in this paper, using different deep-learning based methods. The traditional classifier-based method does not perform well, because of the inherent difficulty of detecting system-level faults for a closed-loop dynamical system. Specifically, the acting controller in any closed-loop dynamical system works to reduce the effect of system-level faults. A novel generative-adversarial-based deep autoencoder is designed to classify data sets under normal and faulty operating conditions. This proposed network performs quite well when compared to any available classifier-based methods, and moreover, does not require labeled fault-incorporated data sets for training purposes. This network's performance is tested on a high-complexity building energy system data set.

1 Introduction

Fault detection and isolation enables safe operation of critical dynamical systems, along with cost effective system performance and maximally effective control performance. For this reason, fault detection and isolation research is of interest in many engineering areas, such as aerospace systems (e.g., [1; 2; 3; 4]), automotive systems (e.g., [5; 6; 7; 8; 9; 10; 11]), photovoltaic systems (e.g., [12; 13; 14; 15; 16; 17; 18]), and building heating and cooling systems (e.g., [20; 21]). For feedback-controlled dynamical systems subjected to exogenous disturbances, fault detection and isolation becomes challenging because the controller expends effort to compensate for the undesired effect of the fault.

In this paper, we will focus on the fault detection problem. The objective will be to successfully distinguish data sets collected under faulty operating conditions from data sets representative of normal operating conditions. We will only investigate physical faults that affect the system dynamics. One can classify the approaches to fault detection problems based on the assumption regarding system dynamics, namely linear or nonlinear systems, and based on the use of a system model for fault detection, either model-driven or data-driven. Model-driven methods use a model for the dynamical system to detect the fault, whereas the data-driven methods do not make explicit use of a model of the physical system. Next we provide a brief overview of the available literature in all these categories.

The fault detection problem for linear systems was first formulated in [22] and [23]. Both papers developed Luenberger-observer based approaches, where the observer gain matrix decouples the effects of different faults. The observer-based approach was extended in [24] to include fault identification by solving the problem of residual generation by processing the inputs and outputs of the system. A model- and parameter-estimation based fault detection method is developed in [25]. An observer-based fault detection approach, where eigenstructure assignment provides robustness to the effects of exogenous disturbances, is demonstrated in [26]. Sliding-mode observers are used in [27] and [28], who also provide fault severity estimates. Isermann and Balle [29] provide an overview of fault detection methods developed in the 1990s, including state and output observers, parity equations, bandpass filters, spectral analysis (fast Fourier transforms), and maximum-entropy estimation.

For nonlinear systems, fault detection methods primarily use the concept of unknown input observability. Controllability and observability Gramians for nonlinear systems are defined in [30]. De Persis and Isidori [31] develop a differential geometric method for fault detection and isolation. They use the concept of an unobservability subspace, based on the similar notion for linear systems (see [32]). The method guarantees the existence of a quotient subsystem of a given system space, which is only affected by the fault of interest. Martinelli [33] develops a generalized algorithm to calculate the rank of the observable codistribution matrix (equivalent to the observability Gramian for linear systems) for nonlinear systems, and demonstrates its applicability for several practical examples, such as motion of a unicycle, a vehicle moving in three-dimensional space, and visual-inertial sensor fusion dynamics.

For a model-based fault detection problem, Maybeck et al. and Elgersma et al. used an assemble of Kalman filters to match a particular fault pattern in [34] and [35], respectively. Boskovic et al. [36] and [37] develop a multiple model method to detect and isolate actuator faults, using multiple hypothesis testing. In [9], a nonlinear observer-based fault identification method has been developed for a robot manipulator, which shows an asymptotic convergence of the fault observer to the actual fault value. Dixon et al. [5] develop a torque filtering based fault isolation for a class of robotic manipulator systems. In [38], a model-based fault detection and identification approach is developed, by using

*This work was supported by the U.S. Department of Energy's Building Technologies Office through the Emerging Technologies, Sensors and Controls Program.

a differential algebraic and residual generation method.

Data-driven approaches such as [39] and [40], use system data to identify the state-space matrices, without using any knowledge of system dynamics. In [41], for a class of discrete time-varying networked systems with incomplete measurements, a least-squares filter paired with a residual matching (RM) approach is developed to isolate and estimate faults. This approach comprises several Kalman filters, with each filter designed to estimate the augment signal, composed of the system state and a specific fault signal, associated with it. An adaptive fault detection and diagnosis method is developed in [42], by implementing a clustering approach to detect faults. For incipient faults, Harmouche et al. in [43] used a principal component analysis (PCA) framework to transform a data set with faulty operating conditions into either principal or residual subspaces. For nonlinear systems, although data-driven approaches are effective in many fault identification scenarios, the quality of fault detection greatly depends on the quality of available training data and the training data span. Zhang et al. [4] proposed merging data-driven and model-based methods in a Bayesian framework. In [44], sparse global-local preserving projections are used to extract sparse transformation vectors from given data set. The extracted sparse transformation is able to extract meaningful features from the data set, which results in a fault related feature extraction, as shown in [44].

Generative adversarial networks (GANs) were introduced in [45] as data generative models in a zero-sum game framework. The training objective for a GAN is to increase the error rate of the discriminative network that was trained on an existing data set. Since their introduction, GANs have been used to augment machine learning techniques to do boosting of classification accuracy, generate samples, and detect fraud [46; 47; 48; 49; 50; 51; 52; 53; 54; 55]. GAN has been proposed as an alternative to variational autoencoders [56; 57]. Several research publications propose algorithms that can distinguish between "true" samples and samples generated by GANs [58; 59; 60; 61].

The remainder of the paper is organized as follows. We provide a mathematical description of the fault detection problem along with the proposed approach in Section 2. In Section 3 we explain the architecture of an autoencoder and we propose a GAN to generate and classify data sets with normal and faulty operating conditions. A novel loss function, suitable for the proposed GAN based autoencoder network, is developed in Section 3. In Section 4, we first train and test a support vector machine (SVM) based classifier, on labeled data sets; (labeling is done based on both faulty and normal operating conditions). Subsequently, we demonstrate a way to improve the performance of the designed SVM, by training a GAN based autoencoder on a Gaussian random data set, which represents data sets with faulty operating conditions, for training the proposed GAN based network. In Section 4, we show further improved performance of our proposed GAN based network architecture using a representative data set with faulty operating conditions generated by taking linear combinations of vectors that are orthogonal to the principal components of the normal data set space. Finally, we summarize our findings in Section 5.

2 The problem and the proposed method

2.1 Problem description

Figure 1 shows a schematic diagram of a closed-loop dynamical system. The dynamics of the system can be mathematically defined as

$$\begin{aligned} \dot{x} &= f(x,u,d) \\ y &= g(x,u,d) \end{aligned} \tag{1}$$

where $x : [0,\infty) \to \mathbb{R}^n$ is the n-dimensional vector containing system states, $u : [0,\infty) \to \mathbb{R}^m$ is the m-dimensional vector containing control inputs, $d : [0,\infty] \to \mathbb{R}^p$ is the p-dimensional vector of exogenous disturbances, $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$ is an unknown nonlinear mapping that represents the system dynamics, $y : [0,\infty] \to \mathbb{R}^q$ is a vector of measurable system outputs, and $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^q$ is an unknown nonlinear mapping that represents the relationship of input to output.

Now we define a fault detection problem for the dynamics in (1) as follows. Given any data set $\mathcal{S}$ containing sample measurement pairs of u and y, identify an unwanted change in the system dynamics. In order to further generalize the fault detection problem, we will only use the data set representing normal operating conditions. This restriction uses the fact that having a data set that incorporates faulty operating conditions indicates either having the capabilities of inserting system-level faults in the dynamics or having a known system dynamics f (as in (1)). Developing either of these aforementioned capabilities involves manual labor and associated cost. We will further assume that the observable part of the system described in (1) can be sufficiently identified from the available data set with normal operating conditions.

2.2 Proposed method description

For the fault detection problem described in Section 2.1, we develop a GAN based deep autoencoder, which uses data set with normal operating conditions (let us define this data space as S_0), to successfully identify the presence of faulty operating conditions in a given data set. In order to do that, we take the principal components of S_0 and use the orthogonals to those principal components to define a vector space S_1. Now, the training objective of our proposed GAN is to "refine" S_1, to calculate $S_2 \subseteq S_1$, such that S_1 becomes a representative of the data set that contains system-level faulty operating conditions. The purpose of our deep autoencoder is to learn the data structure of S_0, by going through the process of encoding and decoding. Upon selecting an encoding dimension, we map the GAN generated space S_2 to the selected encoding dimension space. Let us designate the encoded representation of S_0 as S_0^*, and S_2 as S_2^*. Our final step is to design a classifier, which takes S_0^* and S_2^* for training. Furthermore, this entire training process, of both GAN and the deep autoencoder, is done simultaneously by defining a cumulative loss function.

In order to motivate the requirement of doing an orthogonal transformation on S_0 to define S_1, we demonstrate a case of defining S_1 using Gaussian random noises, and follow the aforementioned training process of the proposed network. Moreover, a single SVM based classifier is trained on labeled normal and fault-incorporated data sets, to compare performance with our proposed network for two different cases (orthogonal transformation and Gaussian-noise based prior selection).

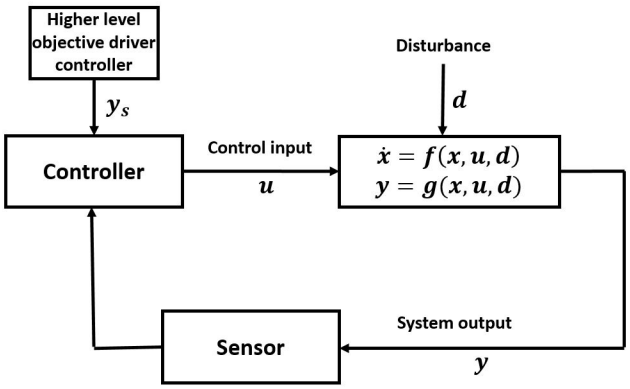

Figure 1: A closed-loop dynamical system.

3 Proposed deep-learning based method

3.1 Overview of autoencoder networks

Autoencoders are multilayer computational graphs used to learn a representation (encoding) for a set of data, for the purpose of dimensionality reduction or data compression. In other words, the training objective of our proposed deep autoencoder is to learn the underlying representation of a data set with normal operating conditions while going through the encoding transformations. A deep enough autoencoder, in theory, should be able to extract a latent representation signature from the training data, which can then be used to better distinguish normal and faulty operation. An autoencoder comprises two different transformations, namely encoding and decoding. The architecture of an autoencoder was first introduced and described by Bengio et al. in [62]. The encoder takes an input vector $\mathbf{x} \in \mathbf{R}^d$ and maps it to a hidden (encoded) representation $\mathbf{x}_e \in \mathbf{R}^{d'}$, through a convolution of deterministic mappings. The decoder maps the resulting encoded expression into a reconstruction vector $\mathbf{x}'$. We will use the notation $\mathcal{E}$ and $\mathcal{G}$ for the encoder and decoder of the autoencoder respectively.

Let the number of layers in the autoencoder network be $2n+1$, and let $\mathbf{y}_i$ denote the output for the network's i^{th} layer. Then

$$\mathbf{y}_0 = \mathbf{x}, \mathbf{x}' = \mathbf{y}_{2n+1}$$
$$\mathbf{y}_i = \sigma_i(\mathbf{w}_i^t \mathbf{y}_{i-1} + b_i), \forall i \in [1, 2n+1]\,.$$

Let $\boldsymbol{\theta}_i = \{\mathbf{w}_i, b_i\}$ denote the parameters of the i^{th} layer and $\sigma_i : \mathbf{R} \rightarrow \mathbf{R}$ be the activation function selected for each layer of the autoencoder. Let us also define $\boldsymbol{\theta} = \{\boldsymbol{\theta}_i\}_{i=1}^{2n+1}$.

$\boldsymbol{\theta}$ defined for this autoencoder is optimized to minimize the average reconstruction error, given by

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \frac{1}{m\sqrt{d}} \sum_{i=1}^{m} L(\mathbf{x}_i, \mathbf{x}_i') \tag{2}$$

where L is square of Euclidean distance, defined as $L(\mathbf{x}, \mathbf{x}') \triangleq \|\mathbf{x} - \mathbf{x}'\|^2$, and $m \in \mathbb{N}$ is the number of available data points.

Our proposed autoencoder is trained on the normal data set, mentioned in Section 4.1. 90% of the normal data (data span one year, with 5 minute resolution) is used to train the autoencoder, and the rest 10% is used for testing. Figure 2 shows both training and testing performance of our autoencoder with encoding dimension 100, with increase in training epochs. Furthermore, selecting the proper encoding dimension is crucial for the following classifier to perform optimally. Figure 2 also demonstrates that the true positive accuracy rate from the classifier decreases when we decrease the encoding dimension. This signifies the loss of valuable information, if we keep decreasing the encoding dimension. For our application, we selected an encoding dimension of 100.

In the next subsection, we give the formulation of our proposed generative model. Our generative model is in the spirit of the well-known GAN [45]. Our proposed model will essentially generate samples that are not from the training data population. So clearly, unlike GAN, here the objective is not to fool the discriminator but to learn which samples are different. We will first formulate our proposed model and then comment on the relationship of our model with GAN in detail.

3.2 Our proposed generative model

We will use $\mathbf{x}_1$ to denote a sample of the data from the normal class, i.e., the class for which the training data is given. Let p_{data} be the distribution of the normal class. We will denote a sample from the abnormal class by $\mathbf{x}_2$. In our setting, the distribution of the abnormal class is unknown, because the training data does not have any samples from the abnormal class. Our generative model will generate sample $\mathbf{x}_2$ from the unknown distribution, p_{noise}. *Note, here we will use the terminology "data" and "noise" to denote the normal and abnormal samples*. Let p_z be the prior of the noise in the encoding space, i.e., $\mathbf{x}_2 \sim p_{\text{noise}} = \mathcal{G}(p_z)$. Furthermore, let $\mathcal{D}$ be the discriminator (a multilayer perceptron for binary classification), such that

$$\mathcal{D}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \sim p_{\text{data}} \\ 0, & \text{if } \mathbf{x} \sim p_{\text{noise}} \end{cases}$$

We will solve for $\mathcal{E}$, $\mathcal{G}$, and $\mathcal{D}$ in a maximization problem with the error function V as follows:

$$V(\mathcal{D}, \mathcal{E}, \mathcal{G}) = \mathbf{E}_{\mathbf{x}_1 \sim p_{\text{data}}} \left((1 - L(\mathbf{x}_1, \mathcal{G}(\mathcal{E}(\mathbf{x}_1)))) + \log(\mathcal{D}(\mathbf{x}_1)) \right) + \mathbf{E}_{\mathbf{z} \sim p_z} \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}))) \tag{3}$$

Note that here, L is as defined in Eq. 2. Furthermore, L is normalized in $[0, 1]$. Now, we will state and prove some theorems about the optimality of the solutions for the error function V.

Theorem 1. *For fixed $\mathcal{G}$ and $\mathcal{E}$, the optimal $\mathcal{D}$ is*

$$\mathcal{D}^*(\mathbf{x}) = \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_{noise}(\mathbf{x})} \tag{4}$$

Proof. Given $\mathcal{G}$ and $\mathcal{E}$, $V(\mathcal{D}, \mathcal{E}, \mathcal{G})$ can be written as:

$$\begin{aligned}\bar{V}(\mathcal{D}) &= \mathbf{E}_{\mathbf{x}_1 \sim p_{\text{data}}}(\log(\mathcal{D}(\mathbf{x}_1))) + \mathbf{E}_{\mathbf{z} \sim p_z} \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}))) \\ &= \int_{\mathbf{x}} (p_{\text{data}}(\mathbf{x}) \log(\mathcal{D}(\mathbf{x})) + p_{\text{noise}}(\mathbf{x}) \log(1 - \mathcal{D}(\mathbf{x})))\end{aligned}$$

The above function achieves the maximum at $\mathcal{D}^*(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\text{noise}}(\mathbf{x})}$. □

Theorem 2. *With $\mathcal{D}^*$ and fixed $\mathcal{E}$, the optimal $\mathcal{G}$ is attained when $\mathbf{x}_1 \sim p_{data}$ and $\mathbf{x}_2 \sim p_{noise}$ has zero mutual information.*

Proof. Observe, from Eq. 3, the first term is maximized if and only if the loss, L, is zero. Hence, when $\mathcal{D} = \mathcal{D}^*$ and $\mathcal{E}$ are fixed, the objective function, V reduces to

$$\begin{aligned}\mathbf{E}_{\mathbf{x}_1 \sim p_{\text{data}}}(1 - L(\mathbf{x}_1, \mathcal{G}(\mathcal{E}(\mathbf{x}_1)))) + H(\mathbf{x}_1, \mathbf{x}_2) \\ - H(\mathbf{x}_1) - H(\mathbf{x}_2)\end{aligned}$$

The first term goes to zero, 1, when the reconstruction is perfect, then, the remaining term is maximized *iff*,

$$H(\mathbf{x}_1, \mathbf{x}_2) - H(\mathbf{x}_1) - H(\mathbf{x}_2) = 0$$

where $H(.)$ and $H(.,.)$ denote the marginal and joint entropies, respectively. Note that, the LHS of the above expression is the mutual information, which is denoted by $I(\mathbf{x}_1, \mathbf{x}_2)$. Hence, the claim holds. □

Theorem 2 signifies that $\mathbf{x}_1 \sim p_{\text{data}}$ and $\mathbf{x}_2 \sim p_{\text{noise}}$ have zero mutual information, i.e., the distributions p_{data} and p_{noise} are completely uncorrelated. This is exactly what we intend to get, i.e., we want to generate abnormal samples that are completely different from the normal samples (training data). Now, we will talk about how to choose the prior p_z after commenting on the contrast of our proposed formulation with [45]. In GAN [45], the generator is essentially mimicking the data distribution to fool the discriminator. On the contrary, because our problem requires that samples be generated from outside training data, our proposed generator generates samples outside the data distribution. Note that one can choose the Wasserstein loss function in Eq. 3 similar to [48]. Below we will mention some of the important characteristics of our proposed model.

- Though we have called it a GAN based autoencoder, clearly the decoder $\mathcal{G}$ is generating the samples and hence acts as a generator in GAN.
- In Equation 3, on samples drawn from p_{data}, autoencoder (i.e., both encoder and decoder) acts, i.e., $\mathcal{G}(\mathcal{E}(\mathbf{x}_1))$ should be very closed to $\mathbf{x}_1$, when $\mathbf{x}_1 \sim p_{\text{data}}$. On the contrary, on $\mathbf{z} \sim p_z$, only the decoder (generator $\mathcal{G}$) acts. Thus, the encoder is learned only from p_{data}, while the decoder (generator) is learned from both p_{data} and p_z.
- Unlike GAN, here we do not have a two player min max game, instead we have a maximization problem over all the unknown parameters. Intuitively, this can be justified, because we are not generating counterfeit samples.

How to choose prior p_z

If we do not know anything about the structure of the data, i.e., about p_{data}, an obvious choice of prior for p_z is a uniform prior. In this work, we have used PCA to extract the inherent lower-dimension subspace containing the data (or most of the data). This is essential not only for the selection of p_z but for the selection of the encoding dimension as well. By the construction of our proposed formulation, the support of p_z should be in the encoding dimension, i.e., in $\mathbf{R}^{d'}$. Given the data, we will choose d' to be the number of principal directions along which the data has $> 90\%$ variance. The span of these d' bases will give a point, $\mathcal{S}$, on the Grassmannian $\text{Gr}(d', d)$, i.e., the manifold of d' dimensional subspaces in $\mathbf{R}^d$. The PCA suggests that "most of the data" lies on $\mathcal{S}$. In order to make sure that the generator generates p_{noise} different from p_{data}, we will use the prior p_z as follows.

Let $\mathcal{N} \in \text{Gr}(d', d)$ be such that $\mathcal{N} \neq \mathcal{S}$. Let $\{\mathbf{n}_i\}_{i=1}^{d'}$ be the bases of $\mathcal{N}$. We will say a sample $\mathbf{z} \sim p_z$ if $\mathbf{z}_i = \mathbf{x}^t \mathbf{n}_i$, for all i, for some $\mathbf{x} \sim p_{\text{data}}$. Without any loss of generality, assume $2d' > d$; then, we can select the first $d - d'$ $\mathbf{n}_i$s to be orthogonal to $\mathcal{S}$ (this can be computed by using Gram-Schmidt orthogonalization). The remaining $\{\mathbf{n}_i\}$s we will select from the bases of $\mathcal{S}$.

4 Results

In this section, we will present experimental validation of our proposed GAN based model. Recall that in our setting, we have only the "normal" samples in the training set and both "normal" and "faulty" samples in the testing set. In the training phase, we will use our proposed GAN based framework to generate samples from the population that are uncorrelated to the normal population. We will teach a discriminator to do so. Then, in the testing phase, we will show that our trained discriminator can distinguish "normal" from "faulty" samples with high prediction accuracy. Furthermore, we will also show that using the prior, as suggested in Section 3.1, gives better prediction accuracy than the Gaussian prior.

4.1 Dataset

We use simulation data from a high-fidelity building energy system emulator. This emulator captures the building thermal dynamics, the performance of the building heating, ventilation, and air conditioning (HVAC), as well as the building control system. The control sequences that drive operation of the building HVAC are representative of typical existing large commercial office buildings in the U.S. We selected Chicago for the building location, and we used the typical meteorological year TMY3 data as simulation input. The data set comprises normal operation data and data representative of operation under five different fault types. We use these labeled data sets for training an SVM based classifier. The five fault types are the following: constant bias in outdoor air temperature measurement (Fault 1), constant bias in supply air temperature measurement (Fault 2), constant bias in return air temperature measurement (Fault 3), offset in supply air flow rate (Fault 4), and stuck cooling coil valve (Fault 5). Table 1 summarizes the characteristics of the data set including fault location, intensity, type, and data length.

Faulty Component	System	Time of Year	Fault Intensity	Data Length
-	Building HVAC	Jan-Dec	-	Yearly
Outdoor air temperature sensor	Mid-floor AHU	Feb, May, Aug, Nov	$\pm 2, \pm 4$ °C	Monthly
Supply air temperature sensor	Mid-floor AHU	Aug	-2 °C	Monthly
Return air temperature sensor	Mid-floor AHU	May-Jun	$+4$ °C	Monthly
Supply air flow rate set point	Mid-floor AHU	May-Jun	-0.1 kg/s	Monthly
Cooling coil valve actuator	Mid-floor AHU	Aug	25%, 50%	Monthly

Table 1: Data set includes normal and fault scenarios sampled at 1 minute resolution. AHU is air handling unit.

4.2 Application of SVM on simulated dataset

Support vector machines are statistical classifiers originally introduced by [63] and [64], later formally introduced by [65]. In this subsection, we will briefly demonstrate the use of SVMs for classifying properly labeled datasets with normal and various faulty operating conditions. SVM separates a given set of binary labeled training data with a hyperplane, which is at maximum distance from each binary label. Therefore, the objective of this classification method is to find the maximal margin hyperplane for a given training data set. For our work, a linear separation is not possible (i.e., to successfully draw a line to separate faulty and normal data sets); that motivates the necessity of using a radial basis function (RBF) kernel ([66]), along with finding a non-polynomial hyperplane to separate the labeled datasets.

Before describing SVM classification in detail, the RBF kernel (see [64]) on two samples x_i and x_j is defined as

$$K_{ij} \triangleq K(x_i, x_j) = \exp\Big(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\Big), \qquad (5)$$

where $\|x_i - x_j\|$ denotes the square of the Euclidean distance, and σ is a user-defined parameter, selected to be unity for this work.

A Scikit learning module available in Python 3.5+ is used for implementation of SVM on the building HVAC data set. Specifically, NuSVC is used with a cubic polynomial kernel function to train for normal and faulty data classification. As the nu value represents the upper bound on the fraction of training error, a range of nu values from 0.5 to 0.9 are tried during cross validation of the designed classifier. Table 2 shows the confusion matrix for the designed SVM classifier, for data sets labeled "normal" and "fault type 1," where the true positive accuracy rate is less than 50%. This finding, as mentioned before, justifies the need to develop an adversarial based classifier, which uses the given normal data to create representative faulty training dataset.

Table 2: Confusion matrix for simulated building environment data set using SVM classifier

		True diagnosis: Normal	True diagnosis: Fault 1	Total
Prediction	Normal	43.27%	56.73%	100%
	Fault	52.18%	47.82%	100%

4.3 Data set using Gaussian noise

In Table 3, the confusion matrix is shown for the GAN based autoencoder, where Gaussian noise is used as an input to GAN, for representing a training class of fault types for the GAN based autoencoder. From left to right, the values in Table 4, denote true positive rate (TPR), false positive rate (FPR), false negative rate (FNR), and true negative rate (TNR). Although in Table 3, the normal data set gives more than 90% TPR, the faulty data set gives around 40% TPR. We can conclude that Gaussian noise as an initial representative of a faulty data set does not represent a completely different faulty data set from the normal data set.

Table 3: Confusion matrix for building data set using Gaussian-noise generator

		True diagnosis: Normal	True diagnosis: Fault	Total
Prediction	Normal	93.10%	6.90%	100%
	Fault	58.40%	41.60%	100%

4.4 Data set using PCA and orthogonal projection

Table 4 shows the confusion matrix for the same high fidelity building data set, where PCA is used to generate an initial representative of a fault-incorporated data set as in Section 3.2. For the sake of completeness, a few other confusion matrix terms are also calculated as follows: TPR is 92.30%, TNR is 72.80%, FPR is 22.76%, FNR is 27.20%, accuracy (ACC) is 82.55%, positive predictive value (PPV) is 77.24%, negative predictive value (NPV) is 90.43%, false discovery rate (FDR) is 22.76%, and finally, false omission rate (FOR) is 9.57%. Table 4 shows better classification performance, compared to both the other methods described before.

Table 4: Confusion matrix for building data set using PCA transformation

		True diagnosis: Normal	True diagnosis: Fault	Total
Prediction	Normal	92.30%	7.70%	100%
	Fault	27.20%	72.80%	100%

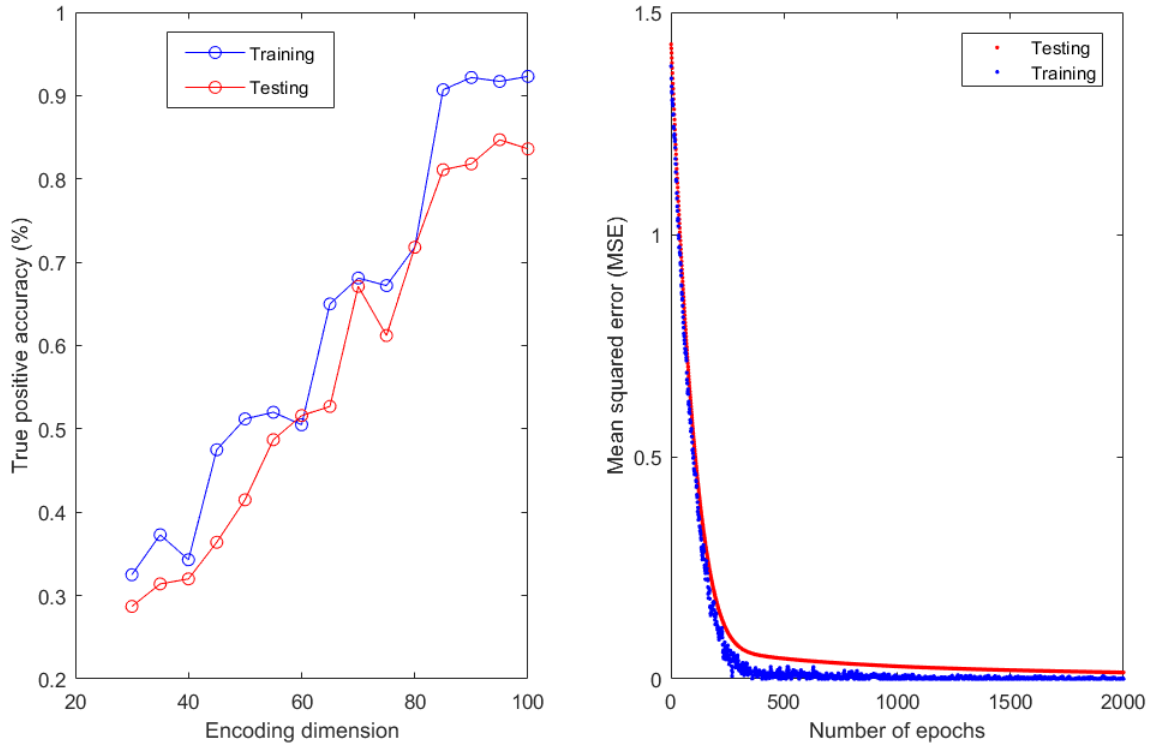

Figure 2: Change in true positive accuracy with change in encoding dimension (left); Training and testing performance of the proposed autoencoder with encoding dimension 100 (right)

4.5 Comparison of results

We demonstrated three different methods for the fault detection problem, applied to a high complexity building data set. The SVM classifier, despite using labeled data sets, gives poor TPRs for our data set. Our proposed GAN based deep autoencoder network is trained and tested using two different training approaches. First, we use a Gaussian-noise based data set as a representation of space S_1 (as in Section 2) to train the designed GAN and simultaneously find a representative class for a data set with faults, i.e., S_2^*. Although the Gaussian-noise based data set gives much better TPR for the normal data set than the SVM, it performs poorly when identifying a data set with faulty conditions. Second, we use orthogonal transformation on the normal data set to generate S_2, and subsequently our proposed GAN based autoencoder is trained on this new S_2 to generate S_2^*. Although the orthogonal transformation based training approach gives similar TPRs for the normal data set, it gives significantly better performance for the data set with faulty conditions than the Gaussian-noise based training approach.

4.6 Group testing

In this section, we will do some statistical analysis of the output produced by our proposed framework. More specifically, we will do group testing in the encoding space, i.e., we will pass the generated noise and the data through the trained encoder and perform a group test. However, because we do not know the distribution of the data and noise in the encoding space, we cannot do a two-sample t-test. We will develop a group testing scheme for our purpose. Let $\left\{\mathbf{y}_i^1\right\}$ and $\left\{\mathbf{y}_i^2\right\}$ be two sets of samples in the encoding space generated using our proposed network. Let $\left\{C_i^1 := \mathbf{y}_i^1 \left(\mathbf{y}_i^1\right)^t\right\}$ and $\left\{C_i^2 := \mathbf{y}_i^2 \left(\mathbf{y}_i^2\right)^t\right\}$ be the corresponding covariance matrices capturing the interactions among dimensions. We will identify each of the covariance matrices with the product space of Stiefel and symmetric positive definite (SPD) matrices, as proposed in [67].

Now, we perform the kernel based two-sample test to find the group difference [68] between $\left\{C_i^1\right\}$ and $\left\{C_i^2\right\}$. In order to use their formulation, we first define the intrinsic metric we will use in this work. We will use the general linear (GL)-invariant metric for SPD matrices, which is defined as follows: Given X, Y as two SPD matrices, the distance, $d(X,Y) = \sqrt{\text{trace}\left(\left(Log\left(X^{-1}Y\right)\right)^2\right)}$. For the Stiefel manifold, we will use the canonical metric [69]. On the product space, we will use the ℓ_1 norm as the product metric. As the kernel, we will use the Gaussian RBF, which is defined as follows: Given $C_1 = (A, X)$ and $C_2 = (B, Y)$ as two points on the product space, the kernel, $k\left(C_1, C_2\right) := \exp\left(-d^2\left(C_1, C_2\right)\right)$. Here, d is the product metric. Given $\left\{C_i^1\right\}_{i=1}^{N_1}$ and $\left\{C_i^2\right\}_{i=1}^{N_2}$, the maximum mean

discrepancy (MMD) is defined as follows:

$$\text{MMD}\left(\{C_i^1\},\{C_i^2\}\right)^2 = \frac{1}{N_1^2}\sum_{i,j} k\left(C_i^1, C_j^1\right) - \frac{2}{N_1 N_2}\sum_{i,j} k\left(C_i^1, C_j^2\right) + \frac{1}{N_2^2}\sum_{i,j} k\left(C_i^2, C_j^2\right) \quad (6)$$

For a level α test, we reject the null hypothesis $H_0 = \{\text{samples from the two groups are from same distribution}\}$ if $MMD < 2\sqrt{1/\max N_1, N_2}\left(1+\sqrt{-\log\alpha}\right)$. Finally, we conclude from the experiments that for our proposed framework, we reject the null hypothesis with 95% confidence.

5 Conclusion

A novel GAN based autoencoder is introduced in this paper. This proposed network performs very well when compared to an SVM based classifier. Although the SVM classifier uses labeled training data for classification, it still gives less than 50% TPR for our high complexity simulated data set. On the other hand, the proposed GAN based deep autoencoder gives significantly better performance for two different types of training scenarios. The proposed GAN based autoencoder is initially trained on a random Gaussian data set. Next, orthogonal projection is used to generate a data set that is perpendicular to the given normal data set. This orthogonally projected data set is used as an initial fault-incorporated data set for our proposed GAN based autoencoder for training. Confusion matrices for both training scenarios are presented, and both of them perform very well compared to the SVM based classification approach. Finally, a statistical group test demonstrates that our encoded normal and GAN based fault-incorporated data spaces (i.e., data sets in S_0^* and S_2^* spaces, respectively) are statistically different, and subsequently validates the favorable performance of our proposed network.

References

[1] RJ Patton. Fault detection and diagnosis in aerospace systems using analytical redundancy. *Computing & Control Engineering Journal*, 2(3):127–136, 1991. 1

[2] Ron J Patton and Jie Chen. Robust fault detection of jet engine sensor systems using eigenstructure assignment. *Journal of Guidance, Control, and Dynamics*, 15(6):1491–1497, 1992. 1

[3] Ron J Patton and Jie Chen. Review of parity space approaches to fault diagnosis for aerospace systems. *Journal of Guidance, Control, and Dynamics*, 17:278–278, 1994. 1

[4] Shuo Zhang and Miroslav Barić. A Bayesian approach to hybrid fault detection and isolation. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 4468–4473. IEEE, 2015. 1, 2

[5] Warren E Dixon, Ian D Walker, Darren M Dawson, and John P Hartranft. Fault detection for robot manipulators with parametric uncertainty: A prediction-error-based approach. *IEEE Transactions on Robotics and Automation*, 16(6):689–699, 2000. 1

[6] Aengus Murray, Bruce Hare, and Akihiro Hirao. Resolver position sensing system with integrated fault detection for automotive applications. In *Sensors, 2002. Proceedings of IEEE*, volume 2, pages 864–869. IEEE, 2002. 1

[7] Domenico Capriglione, Consolatina Liguori, Cesare Pianese, and Antonio Pietrosanto. On-line sensor fault detection, isolation, and accommodation in automotive engines. *IEEE Transactions on Instrumentation and Measurement*, 52(4):1182–1189, 2003. 1

[8] Rolf Isermann. Model-based fault detection and diagnosis–status and applications. *IFAC Proceedings Volumes*, 37(6):49–60, 2004. 1

[9] Michael L McIntyre, Warren E Dixon, Darren M Dawson, and Ian D Walker. Fault detection and identification for robot manipulators. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4981–4986. IEEE, 2004. 1

[10] Inseok Hwang, Sungwan Kim, Youdan Kim, and Chze Eng Seah. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, 18(3):636–653, 2010. 1

[11] Justin Flett and Gary M Bone. Fault detection and diagnosis of diesel engine valve trains. *Mechanical Systems and Signal Processing*, 72:316–327, 2016. 1

[12] Steven K Firth, Kevin J Lomas, and Simon J Rees. A simple model of PV system performance and its use in fault detection. *Solar Energy*, 84(4):624–635, 2010. 1

[13] A Chouder and S Silvestre. Automatic supervision and fault detection of PV systems based on power losses analysis. *Energy conversion and Management*, 51(10):1929–1937, 2010. 1

[14] Henry Braun, Santoshi T Buddha, Venkatachalam Krishnan, Andreas Spanias, Cihan Tepedelenlioglu, Ted Yeider, and Toru Takehara. Signal processing for fault detection in photovoltaic arrays. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1681–1684. IEEE, 2012. 1

[15] Ye Zhao, Ling Yang, Brad Lehman, Jean-François de Palma, Jerry Mosesian, and Robert Lyons. Decision tree-based fault detection and classification in solar photovoltaic arrays. In *Applied Power Electronics Conference and Exposition (APEC), 2012 Twenty-Seventh Annual IEEE*, pages 93–99. IEEE, 2012. 1

[16] Yongheng Yang, Frede Blaabjerg, and Zhixiang Zou. Benchmarking of grid fault modes in single-phase grid-connected photovoltaic systems. *IEEE Transactions on Industry Applications*, 49(5):2167–2176, 2013. 1

[17] Santiago Silvestre, Aissa Chouder, and Engin Karatepe. Automatic fault detection in grid connected PV systems. *Solar Energy*, 94:119–127, 2013. 1

[18] Elyes Garoudja, Fouzi Harrou, Ying Sun, Kamel Kara, Aissa Chouder, and Santiago Silvestre. A statistical-based approach for fault detection and diagnosis in a photovoltaic system. In *Systems and Control (ICSC), 2017 6th International Conference on*, pages 75–80. IEEE, 2017. 1

[19] ME Ropp, M Begovic, and A Rohatgi. Prevention of islanding in grid-connected photovoltaic systems.

Progress in Photovoltaics: Research and Applications, 7(1):39–59, 1999.

[20] Zhimin Du, Bo Fan, Xinqiao Jin, and Jinlei Chi. Fault detection and diagnosis for buildings and HVAC systems using combined neural networks and subtractive clustering analysis. *Building and Environment*, 73:1–11, 2014. 1

[21] Dian-ce Gao, Shengwei Wang, Kui Shan, and Chengchu Yan. A system-level fault detection and diagnosis method for low delta-t syndrome in the complex hvac systems. *Applied Energy*, 164:1028–1038, 2016. 1

[22] Richard Vernon Beard. *Failure accomodation in linear systems through self-reorganization.* PhD thesis, Massachusetts Institute of Technology, 1971. 1

[23] Harold Lee Jones. *Failure detection in linear systems.* PhD thesis, Massachusetts Institute of Technology, 1973. 1

[24] M-A Massoumnia, George C Verghese, and Alan S Willsky. Failure detection and identification. *IEEE transactions on automatic control*, 34(3):316–321, 1989. 1

[25] Paul M Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica*, 26(3):459–474, 1990. 1

[26] RJ Patton and J Chen. Observer-based fault detection and isolation: Robustness and applications. *Control Engineering Practice*, 5(5):671–682, 1997. 1

[27] Christopher Edwards, Sarah K Spurgeon, and Ron J Patton. Sliding mode observers for fault detection and isolation. *Automatica*, 36(4):541–553, 2000. 1

[28] Chee Pin Tan and Christopher Edwards. Sliding mode observers for robust detection and reconstruction of actuator and sensor faults. *International Journal of Robust and Nonlinear Control*, 13(5):443–463, 2003. 1

[29] Rolf Isermann and Peter Balle. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997. 1

[30] Robert Hermann and Arthur Krener. Nonlinear controllability and observability. *IEEE Transactions on automatic control*, 22(5):728–740, 1977. 1

[31] Claudio De Persis and Alberto Isidori. A geometric approach to nonlinear fault detection and isolation. *IEEE transactions on automatic control*, 46(6):853–865, 2001. 1

[32] Alberto Isidori. *Nonlinear control systems*. Springer Science & Business Media, 2013. 1

[33] Agostino Martinelli. Nonlinear unknown input observability: The general analytic solution. *arXiv preprint arXiv:1704.03252*, 2017. 1

[34] Peter S Maybeck. Multiple model adaptive algorithms for detecting and compensating sensor and actuator/surface failures in aircraft flight control systems. *International Journal of Robust and Nonlinear Control*, 9(14):1051–1070, 1999. 1

[35] Michael Elgersma and Sonja Glavaski. Reconfigurable control for active management of aircraft system failures. In *American Control Conference, 2001. Proceedings of the 2001*, volume 4, pages 2627–2639. IEEE, 2001. 1

[36] JD Boskovic, S-M Li, and Raman K Mehra. Intelligent spacecraft control using multiple models, switching, and tuning. In *Intelligent Control/Intelligent Systems and Semiotics, 1999. Proceedings of the 1999 IEEE International Symposium on*, pages 84–89. IEEE, 1999. 1

[37] Jovan D Boskovic, Sai-Ming Li, and Raman K Mehra. On-line failure detection and identification (fdi) and adaptive reconfigurable control (ARC) in aerospace applications. In *American Control Conference, 2001. Proceedings of the 2001*, volume 4, pages 2625–2626. IEEE, 2001. 1

[38] József Bokor and Zoltán Szabó. Fault detection and isolation in nonlinear systems. *Annual Reviews in Control*, 33(2):113–123, 2009. 1

[39] SX Ding, P Zhang, A Naik, EL Ding, and B Huang. Subspace method aided data-driven design of fault detection and isolation systems. *Journal of Process Control*, 19(9):1496–1510, 2009. 2

[40] Jianfei Dong and Michel Verhaegen. Data driven fault detection and isolation of a wind turbine benchmark. *IFAC Proceedings Volumes*, 44(1):7086–7091, 2011. 2

[41] Xiao He, Zidong Wang, Yang Liu, and DH Zhou. Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach. *IEEE Transactions on Industrial Informatics*, 9(3):1670–1679, 2013. 2

[42] Andre Lemos, Walmir Caminhas, and Fernando Gomide. Adaptive fault detection and diagnosis using an evolving fuzzy classifier. *Information Sciences*, 220:64–85, 2013. 2

[43] Jinane Harmouche, Claude Delpha, and Demba Diallo. Incipient fault detection and diagnosis based on kullback–leibler divergence using principal component analysis: Part i. *Signal Processing*, 94:278–287, 2014. 2

[44] Shiyi Bao, Lijia Luo, Jianfeng Mao, and Di Tang. Improved fault detection and diagnosis using sparse global-local preserving projections. *Journal of Process Control*, 47:121–135, 2016. 2

[45] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2, 3, 4

[46] Shibani Santurkar, Ludwig Schmidt, and Aleksander Madry. A classification-based perspective on GAN distributions. *arXiv preprint arXiv:1711.00970*, 2017. 2

[47] Ilya O Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models. In *Advances in Neural Information Processing Systems*, pages 5430–5439, 2017. 2

[48] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2, 4

[49] LI Chongxuan, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 4091–4101, 2017. 2

[50] Min Lin. Softmax GAN. *arXiv preprint arXiv:1704.06191*, 2017. 2

[51] Parametrizing filters of a CNN with a GAN, author=Kilcher, Yannic and Becigneul, Gary and Hofmann, Thomas, journal=arXiv preprint arXiv:1710.11386, year=2017. 2

[52] How to train your DRAGAN, author=Kodali, Naveen and Abernethy, Jacob and Hays, James and Kira, Zsolt, journal=arXiv preprint arXiv:1705.07215, year=2017. 2

[53] Panpan Zheng, Shuhan Yuan, Xintao Wu, Jun Li, and Aidong Lu. One-class adversarial nets for fraud detection. *arXiv preprint arXiv:1803.01798*, 2018. 2

[54] Tarik Arici and Asli Celikyilmaz. Associative adversarial networks. *arXiv preprint arXiv:1611.06953*, 2016. 2

[55] Yunus Saatci and Andrew G Wilson. Bayesian GAN. In *Advances in Neural Information Processing Systems*, pages 3622–3631, 2017. 2

[56] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016. 2

[57] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. It takes (only) two: Adversarial generator-encoder networks. 2018. 2

[58] Alex Kurakin, Dan Boneh, Florian Tramr, Ian Goodfellow, Nicolas Papernot, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. 2018. 2

[59] Aurko Roy, Colin Raffel, Ian Goodfellow, and Jacob Buckman. Thermometer encoding: One hot way to resist adversarial examples. 2018. 2

[60] Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. AE-GAN: adversarial eliminating with GAN. *arXiv preprint arXiv:1707.05474*, 2017. 2

[61] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 2

[62] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 153–160, 2007. 3

[63] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992. 5

[64] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical Learning Theory*, volume 1. Wiley New York, 1998. 5

[65] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000. 5

[66] Ke-Lin Du and MNS Swamy. Radial basis function networks. In *Neural Networks and Statistical Learning*, pages 299–335. Springer, 2014. 5

[67] Silvere Bonnabel and Rodolphe Sepulchre. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070, 2009. 6

[68] Arthur Gretton, Kenji Fukumizu, Zaid Harchaoui, and Bharath K Sriperumbudur. A fast, consistent kernel two-sample test. In *Advances in Neural Information Processing systems*, pages 673–681, 2009. 6

[69] Tetsuya Kaneko, Simone Fiori, and Toshihisa Tanaka. Empirical arithmetic averaging over the compact Stiefel manifold. *IEEE Transactions on Signal Processing*, 61(4):883–894, 2013. 6