IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 26, 2019, accepted July 5, 2019, date of publication July 22, 2019, date of current version August 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2930257

Improving Prediction Efficacy Through Abnormality Detection and Data Preprocessing

CHUN-CHEN TU[1], PIN-YU CHEN[2], AND NAISYIN WANG[1]
[1]Department of Statistics, University of Michigan, Ann Arbor, MI 48105, USA
[2]IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA

Corresponding author: Chun-Chen Tu (timtu@umich.edu)

This work was supported by the National Institutes of Health under Grant CA74552.

ABSTRACT Abnormal testing data can severely reduce model performance if not processed properly. In this paper, we propose a preprocessing system to handle different types of commonly seen abnormal testing data. The system consists of an aberrant data detector and an aberrant data corrector. The aberrant data detector is responsible for classifying the type of incoming data. Based on the data type, the aberrant data corrector will take different actions to amend testing data. Users can then apply their preferred prediction methods on the corrected testing data. Specifically, corrupted and adversarial images are used as examples of abnormal data. We show that corrupted data can be reconstructed through a Gaussian locally linear mappings method, and the prediction performance of adversarial samples can be improved by using the nearest neighbors as a surrogate. We compare the proposed aberrant data detector and corrector with existing and well-recognized alternatives. These approaches are published individually and do not put two components together as a pre-processing system. The numerical outcomes show that our proposed components, standing alone, are competitive. The proposed system is a generic method that can be applied to different downstream predictive models. We use three existing prediction methods to illustrate the general usage of the proposed system and its capability of improving prediction efficacy.

INDEX TERMS Data preprocessing, Gaussian mixture model, image reconstruction, outlier detection, principal component analysis.

I. INTRODUCTION

Prediction efficacy relies on testing data following the patterns learned by the prediction model built with the training samples. However, this assumption may not always be valid in practice. Newly observed data could be altered or contaminated, which reduces the prediction performance. To obtain reliable prediction results on irregular entries, it is necessary to detect the irregular samples and fix them accordingly. As the sources of causing irregular patterns may vary, different strategies should be applied to different types of abnormal data. In addition, it is not necessary to apply the repairing step to regular data. Thus, it may be beneficial to determine the data types before applying any repairing methods on the testing data [27], [33]. A reliable preprocessing system should contain a detector to determine the type of an input, and correctors to fix the irregular patterns, depending on the input type. An input could be normal, which follows the pattern captured by the model, or abnormal caused by different reasons such as being corrupted or being adversarially perturbed. A predictive model can be directly applied to a normal input. In the meantime, it is preferred to pre-process an abnormal input so that the irregularity can be fixed to ensure reliable prediction outcomes.

Recent studies focus on detecting two kinds of irregular samples: outliers and adversarial samples. Outlier detection, also known as novelty detection or anomaly detection, aims to construct a classifier to distinguish normal entities and outliers. Since it is impossible to eradicate all kinds of outliers, outlier detection often focuses on training a classifier on normal data, which is the so-called one-class classifier [8], [22], [32]. The classification performance

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaoqing Pan.

highly relies on the features extracted from the data. When the features are selected properly, one can obtain a powerful classifier and achieve high detection accuracy. The representative features that can separate normal and irregular samples vary from dataset to dataset and may require excessive time and efforts to identify. One can also detect outliers based on reconstruction errors. Developing a reliable reconstruction strategy is challenging. Several techniques such as low-rank approximation [25] or deep learning [4] are used to capture the representative features of normal entities. Since the features are extracted from normal samples, the reconstruction errors would be small for normal samples but large for outliers. Low-rank approximation methods often assume linearity and may not capture complex associations embedded in the dataset. Deep learning-based methods [15] are able to model sophisticated associations through different kinds of network architectures. However, these methods require a large volume of data for training, which may not be suitable for moderate size datasets.

The detection task becomes more challenging when dealing with adversarial samples [3], [18], [36]. The differences between adversarial samples and their normal counterparts are usually imperceptible, making the conventional strategies of detecting outliers fail. To overcome the difficulty, maximum mean discrepancy (MMD) [11] is used for testing whether two sets of data are drawn from the same underlying distribution [12]. However, the procedure requires bootstrapping to approximate p-values, which increases the computational burden. Feature squeezing (FS) [34], is motivated by the observation that the feature input spaces are often unnecessarily large, which leaves room for adversarial perturbations. FS intends to remove adversarial perturbations by squeezing out unnecessary input features. One can detect adversarial samples by comparing the model outcomes obtained from the original input and the squeezed version. If the outcomes differ, the input is considered as adversarial, otherwise, regular. FS if effective if classification is the task of interest. However, for prediction, the potentially corrupted samples from FS could enlarge the prediction error, with the side effect of lessening the ability to detect adversarial entities. MagNet [21] learns the manifold of normal samples and projects the inputs onto the learned data manifold. If the input is adversarial, the projection procedure removes the adversarial perturbations and changes the output results. Similar to FS, by comparing the output outcomes, one can detect the adversarial samples. However, MagNet may be less effective in detecting adversarial examples with sparse perturbations [17]. Latent information can be utilized to detect adversarial examples. In [13], PCA whitening is used to find the latent space of normal data. The projected coefficients of abnormal data are different from the ones obtained from the regular samples. Using this fact, one can effectively distinguish adversarial data from normal ones. Notably, all these methods focus on one type of irregular pattern. Detectors are built separately to distinguish either normal data v.s. outliers or normal data v.s. adversarial samples but cannot handle three kinds of inputs at the same time. Some detectors can be combined together to achieve the goal. However, this solution is not straightforward nor readily available and would require sophisticated skills from users to implement it.

The fixing process can be done by finding the closest representations in the latent space [2], [7], [19]. Since the latent space is constructed using regular observations, the abnormal patterns of the restored data will be mitigated. Nearest neighbors can also be used to reconstruct the problematic data. In [1], random sampling is adopted to search patch matches, and these matches are used to correct the damaged parts. With the patch-based methods copying patches to fill the missing parts, these methods can generate disconnected lines or broken edges. In addition, patch-based methods may not generate proper patches if the background around the missing region is too complicated. To improve the patch selection and textual synthesis process, a new priority definition is proposed in [6] to propagate geometry information. With the advancement in deep neural networks, the performance of image restoration has been greatly improved. Deep neural networks learn hidden representations and reconstruct images through convolutional filters [14], [16], [24], [35], which can produce meaningful repair results. When applying these reconstruction methods (patch-based or deep learning-based), the location information of the damaged parts needs to be accessible, which makes the repairing task more difficult in practice since this information is often unknown.

In this work, we propose a preprocessing system to handle different types of abnormal observations when conducting predictions. Our goal is to provide a general framework for regression tasks so that users can use their preferred model(s) for prediction while assuring the outcomes are robust to abnormal data. There are two building blocks in the proposed method. The **aberrant data detector** is used to distinguish different types of testing data. Different from the detectors in the literature, the proposed approach can classify inputs into three categories: normal, corrupted and adversarial. We discuss different alternatives to construct the detector and provide suggestions on the situations to use them. The goal of the **aberrant data corrector** is to provide amended testing data so that users can obtain reliable prediction outcomes from them. We devise a reconstruction method to effectively repair the data without knowing the location of the damaged parts. The proposed method contribute to design a detector for multiple types of irregular patterns. In addition, the reconstruct method utilizing locally-linear mappings is different from the approaches in the related works. The performance of the proposed system is demonstrated using three predictive models showing the general usage of the proposed approach and the capability of it on improving the prediction efficacy.

II. ROBUST PREPROCESSING SYSTEM

This work focuses on the task of predicting responses $Y \in \mathbb{R}^L$ using covariates $X \in \mathbb{R}^D$. It is well-known that outliers could severely affect model performance. There is a considerable amount of studies on building prediction models that are

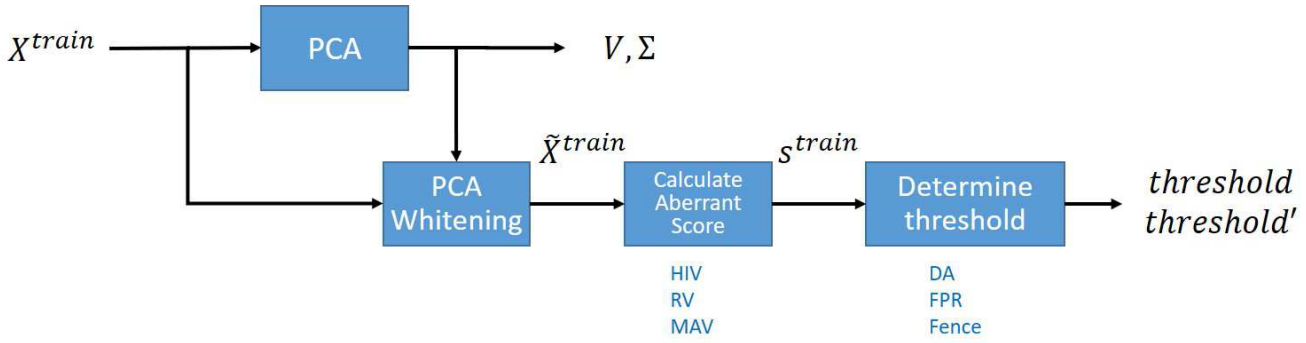


FIGURE 1. The flowchart of constructing the proposed robust preprocessing system at the training stage.

insensitive to outlying training data. However, even if a model is built using a robust process, the prediction errors could still be large if testing samples are abnormal.

Data could deviate from the normal pattern because of different reasons. One possibility is that the data collecting procedure is defected, which causes damages to the data. Another reason could be that data are perturbed under malicious intents. In this work, we refer to the first kind of abnormal data as corrupted data and the second type of data as adversarial data. These two kinds of abnormal data are commonly seen in practice. To properly handle these abnormal data, we propose a robust preprocessing system. An **aberrant data detector** is designed to distinguish normal, corrupted and adversarial data. In addition, an **aberrant data corrector** is devised to amend corrupted and adversarial data so that users can still apply their selections of prediction methods for prediction and obtain reliable outcomes.

A. ABERRANT DATA DETECTOR

The aberrant data detector is responsible for classifying the query data into three categories: normal, corrupted and adversarial. Principal component analysis (PCA) whitening is used as the core technique for detection. To perform PCA whitening, we first calculate the singular vectors and singular values of the training data. When a new testing sample comes in, we calculate its aberrant score and compare the score with a pre-defined *threshold*. If the aberrant score is greater than *threshold*, the testing data point is classified as abnormal (corrupted or adversarial). Otherwise, it is deemed a normal sample. The mechanism can essentially differentiate normal data from abnormal data. We further extend the approach so that two kinds of abnormal data, corrupted data and adversarial data, can be identified separately.

The flowcharts for constructing the robust preprocessing system and building the aberrant data detector/corrector are shown in Figs. 1 and 2, respectively. We first center the training data around zero and perform the singular value decomposition. Denote $X^{train} \in \mathbb{R}^{N\times D}$ as the centered training dataset where N is the number of the training samples and D is the dimension of the sample. We decompose the training dataset as $X^{train} = U\Sigma V^{\top}$ where U is an $N \times N$ unitary matrix, Σ is an $N \times D$ diagonal matrix and V is a $D \times D$ unitary matrix. Letting x_n be the n-th sample in X^{train} and $\tilde{x}_n \in \mathbb{R}^D$ be the normalized coefficients of x_n, we have

$$\tilde{x}_{n,i} = \frac{x_n \cdot v_i}{\sigma_i}, \tag{1}$$

where v_i is the i-th column vector of matrix V and σ_i is the i-th singular value following in descending order.

PCA whitening projects a query sample onto the principal components extracted from the training dataset. The projected coefficients are normalized by the singular values (σ_i) corresponding to the principal components (PC), which we refer to as the normalized coefficients. These normalized coefficients can be used to detect abnormal images. As shown in Fig. 3, the normalized coefficients for a normal sample lie within a small range whereas those for an abnormal sample might not. Here, the high indexes correspond to those for the small singular values. In particular, the scale of the high-indexed normalized coefficients for a problematic sample could be large. For a normal sample, even after the normalized coefficients are scaled by small singular values, the resulting coefficients are still within a certain range. On the other hand, we would obtain larger normalized coefficients from the last few principal components when they correspond to those being distorted or perturbed as shown in Fig. 3. Based on the normalized coefficients at high indexes, we can distinguish normal data from abnormal (corrupted and adversarial) data.

1) CALCULATE THE ABERRANT SCORE

In [13], the variance of the high-indexed normalized coefficients is used as the aberrant score. Hereinafter, we refer to this method as high-indexed variance (HIV). We denote P_{NC} as the portion of the high-indexed normalized coefficients used to calculate the variance. As an example, if the data dimension (D) is 1024 and $P_{NC} = 10$, we would use the last 103 normalized coefficients (the 992-th to the 1024-th) to calculate the variances. We further consider other approaches to calculating aberrant scores. The rolling variance (RV) method calculates the variances using the normalized coefficients in a sliding window. The length of the sliding window is defined by a parameter P_{window}, which is the portion of the consecutive normalized coefficients included in the

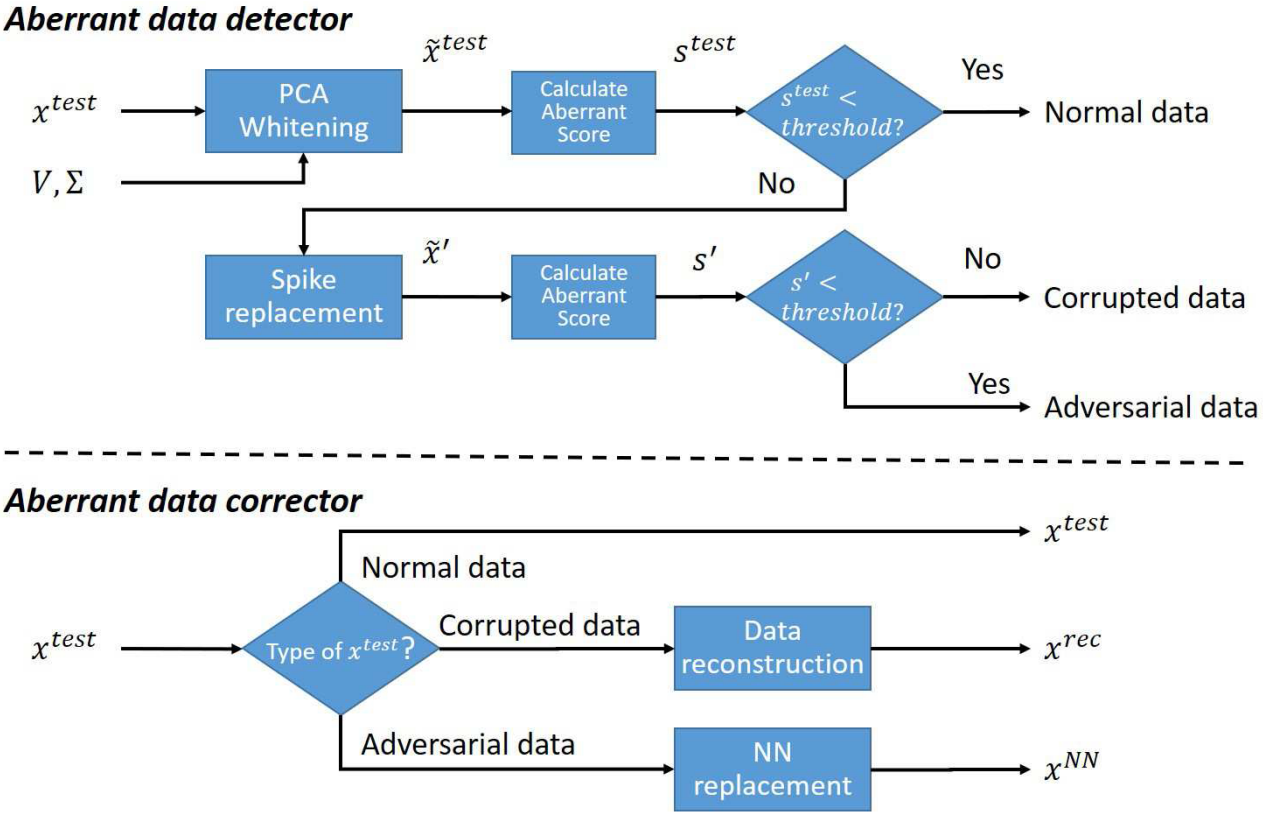


FIGURE 2. The flowchart of the proposed robust preprocessing system at the testing stage.

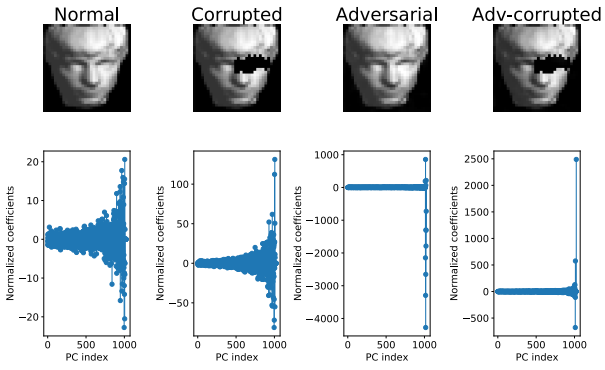


FIGURE 3. Examples of different kinds of images. The first row shows the normal, corrupted, adversarial and adversarially corrupted images. The second row shows the corresponding normalized coefficients against the principal component (PC) indexes.

sliding window. The aberrant score is the maximum value of all of the rolling variances. Finally, the maximum absolute value (MAV) method treats the normalized coefficients with the largest absolute values as the aberrant scores.

2) DETERMINE THE CLASSIFICATION THRESHOLD

In Fig. 1, we calculate the aberrant score s^{train} for each training sample. The next step is to determine the *threshold* for detection. Three methods are proposed. First, we can directly assign (DA) the *threshold*. The *threshold* is selected to reach the best detection results. Under this scenario, both normal and abnormal images are assumed to be accessible. In the second approach, we propose to control the false positive rate, *fpr*, of wrongly detecting abnormal data. That is, given a false positive rate *fpr*, the *threshold* is set to the $1-fpr$ quantile of the aberrant scores obtained from the normal training data. Our last proposed approach is to locate the so-called "fence" which is commonly used in the boxplots. We calculate the mean, μ_{score} and the standard deviation, σ_{score} of the aberrant scores, and the upper fence is calculated as $\mu_{score} + M \times \sigma_{score}$, so that the distance between the fence and the mean is M times standard deviation. We note that the last two approaches do not utilize abnormal entries and can accommodate the scenarios when only normal data are available. These three methods are referred to as "DA", "FPR" and "Fence," respectively.

3) DIFFERENTIATE CORRUPTED AND ADVERSARIAL DATA

By comparing the aberrant score and the pre-determined *threshold*, we can classify the testing data as normal or abnormal. To further differentiate corrupted from adversarial samples, we take advantage of the following observations. In Fig. 3, there are only a few extreme normalized coefficients for adversarial data. On the other hand, no obvious spike appears on the normalized coefficients of corrupted data. Thus, we can remove the spikes and re-calculate the aberrant

score again. If the classification result changes, we classify the query data as an adversarial entry, and otherwise as a corrupted entry.

As noted by the flowchart of the aberrant data detector at the testing stage shown in Fig. 2, for each testing data, $x^{test} \in \mathbb{R}^D$, we first calculate the aberrant score s^{test} using the results of SVD on the training dataset (V, Σ). The aberrant score is compared to $threshold'$. If $s^{test} < threshold'$, the testing data is classified as a normal sample. Otherwise, it is deemed as an abnormal sample. We next find its nearest neighbor in the training dataset, denoted as x^{NN}, and calculate the normalized coefficients of x^{NN}, denoted as $\tilde{x}^{NN}$. We replace the normalized coefficients of $\tilde{x}^{test}$. Denoting $\tilde{x}'$ as the new normalized coefficient vector after replacement, $\tilde{x}'$ is constructed as follows:

$$\tilde{x}'_i = \begin{cases} \tilde{x}_i^{NN} & \text{if } |\tilde{x}_i^{test}| > cutoff, \\ \tilde{x}_i^{test} & \text{otherwise,} \end{cases} \tag{2}$$

where the subscript i denotes the i-th normalized coefficient of $\tilde{x}^{test}$, $\tilde{x}^{NN}$ and $\tilde{x}'$; $cutoff$ is a pre-defined value to identify spike coefficients. The process described in (2) is referred to as spike replacement, where we replace the spike coefficients with normal ones. We then calculate the aberrant score of $\tilde{x}'$ as s' and compare s' to $threshold'$. The testing sample is classified as an adversarial sample if the aberrant score is less than $threshold'$, and as a corrupted sample otherwise.

B. ABERRANT DATA CORRECTOR

Based on the classification results of the aberrant data detector, the aberrant data corrector would adopt different mechanisms. For the corrupted data, we would conduct data reconstruction, which utilizes the associations learned by Gaussian Locally Linear Mappings (GLLiM) [5]. GLLiM assesses complicated relationships between X and Y by dividing data into different clusters, and data within each cluster are assumed to follow a linear association. For a K-component model, GLLiM introduces a latent variable Z such that

$$X = \sum_{k=1}^{K} \mathbb{I}(Z = k)(A_k Y + b_k + E_k), \tag{3}$$

where $\mathbb{I}$ is an indicator function, $A_k \in \mathbb{R}^{D\times L}$ and $b_k \in \mathbb{R}^D$ define the mapping from Y to X, and $E_k \in \mathbb{R}^{D\times D}$ is the error term capturing the remaining uncertainty. Under the Gaussianity assumption, the hierarchical structure can be written as follows:

$$p(X = x|Y = y, Z = k) = \mathcal{N}(x; A_k y + b_k, \Sigma_k), \tag{4}$$

$$p(Y = y, Z = k) = \mathcal{N}(y; c_k, \Gamma_k) \tag{5}$$

$$p(Z = k) = \pi_k, \tag{6}$$

where $\mathcal{N}(\cdot; \mu, \Sigma)$ denotes the Gaussian density function with mean μ and covariance Σ, $c_k \in \mathbb{R}^L$, $\Gamma_k \in \mathbb{R}^{L\times L}$ are the mean and the covariance matrix at low dimension, and π_k is the multinomial prior with $\sum_{k=1}^{K} \pi_k = 1$.

The parameters can be estimated using the Expectation Maximization algorithm described in [5]. GLLiM is a bi-directional modeling process. It is typical to let the dimension of X, D, to be larger than that of Y, L. The mixture setup learns the association not only from Y (low-dimensional) to X (high-dimensional) but also from X to Y. The former model defined in (4)-(6) is referred to as the inverse model, which can be used for data reconstruction. The latter can be obtained directly from the estimated inverse model, while avoiding dealing with a high-dimensional set of predictors in a regression setting, and is called the forward model, which can be used for prediction. Denote $GLLiM\text{-}Inv(\cdot)$ as the GLLiM inverse model:

$$GLLiM\text{-}Inv(y) = \sum_{k=1}^{K} \frac{\pi_k N(y; c_k, \Gamma_k)}{\sum_{j=1}^{K} \pi_j N(y; c_j, \Gamma_j)}(A_k y + b_k). \tag{7}$$

Given a response $y \in \mathbb{R}^L$, we can reconstruct the high-dimensional data x through $x = GLLiM\text{-}Inv(y)$. The reconstruction of the corrupted data can be formulated as follows. Let

$$y^* = \arg\min_y Sim(x^{test}, GLLiM\text{-}Inv(y)), \tag{8}$$

where $Sim(\cdot, \cdot)$ is a function measuring the similarity distance between the testing data x^{test} and the reconstructed data, $GLLiM\text{-}Inv(y)$. Note that with x^{test} being a corrupted image, we should leave out the corrupted pixels when measuring the similarity distance between the testing image and the reconstructed one. To achieve this goal, we design the similarity function using the truncated sum of squared differences. The similarity distance between two vectors a, b of dimension D for a given truncated quantile q is defined as follows:

$$Sim(a, b) = \sum_{i=1}^{D'} c'_i, \tag{9}$$

where we define c as a vector with its i-th entry $c_i = (a_i - b_i)^2$ and let c' be a permutation of c in ascending order, i.e. c'_i is the i-th smallest squared difference between elements in a and b. We define $D' = \lfloor D \times q \rfloor$ where $\lfloor \cdot \rfloor$ denotes the floor function. The truncated sum of squared differences is the summation over the first D' smallest squared differences. The reconstructed image is denoted as x^{rec} in Fig. 2, which can be obtained by $x^{rec} = GLLiM\text{-}Inv(y^*)$.

The output of the aberrant data corrector depends on the results of the detector. If the testing data is identified as normal, x^{test} would be provided directly. Otherwise, the detector will determine the type of abnormal data. The output would be x^{rec} for a corrupted sample. As for an adversarial example, we use the nearest neighbor in the training dataset, x^{NN}, as the prediction surrogate since adversarial examples are close to the manifold of the normal data. After obtaining the preprocessed data from the corrector, we can apply the selected predictive method for prediction.

III. PERFORMANCE EVALUATION

We use images to demonstrate the performance of the proposed method. Nevertheless, the preprocessing system is a general framework and is not restricted to image data. The face dataset [29] contains 698 images (of size 64 × 64 and being further condensed to 32 × 32) and is separated into a training dataset and a testing dataset. The training dataset contains 598 images and the rest of the 100 images are testing samples. Our goal is to predict the head pose (Y) using a given image (X). The pose of each image is defined by three variables in Y: *Light*, *Pan* and *Tilt*. We treat the original images in the dataset as normal data. To evaluate the proposed method, we will discuss the generation of the corrupted and adversarial images followed by demonstrating the performance of aberrant data detection and correction for different types of testing images. We compare these two components within our pre-processing system to the existing aberrant data detection and reconstruction methods, respectively. The configuration and experimental setup behind these comparisons are specified within subsections III-B, III-D, III-E and III-F. In particular, besides evaluating performances for each specific function (detection and correction), the prediction performances using three predictive models are presented to illustrate the efficacies of the proposed method for reducing the prediction errors.

A. GENERATING ABNORMAL DATA

Corrupted images are those with a small region of distortion. Different reasons can result in corrupted images. For example, a shadow will appear on the picture if the light source is blocked. In addition, images could be deteriorated because of physical damages such as stains or scratches. The irregular area could be small but could severely reduce the prediction performance. To generate corrupted images from the normal images, we randomly select an area within an image. The maximum size of the area is set to be 4 × 16 ($pixel^2$). Compared to the original image size, 32 × 32, at most 1/16 of the image will be corrupted. Next, we set the pixel within the selected area to be masked (replaced with pixel values of black color), mimicking the occurrence of damage. The second image in Fig. 3 is an example of the corrupted images.

Adversarial images are intentionally designed to result in model failure [10], [28]. By adding imperceptible perturbations to the input data, we can easily fool a well-trained model. That is, we would obtain large prediction errors. To test the robustness of the proposed method, we use AutoZOOM [31] to generate adversarial samples. AutoZOOM is an effective method that adopts dimension reduction techniques as well as random gradient vector to accelerate the generation process. The adversarial examples are generated toward a GLLiM forward model with the restriction of the normalized perturbation must be less than 0.001. We also consider mixed-type abnormal data by adding adversarial perturbations to the corrupted images with the same restriction of the normalized perturbation. Hereinafter, we call this type of abnormal images as adversarially corrupted (adv-corrupted). Specifically, adv-corrupted samples help us understand how would the aberrant data detector react when both abnormal patterns exist in a single sample. Examples of the adversarial image and the adv-corrupted image are shown in Fig. 3. Comparing the adversarial images to their counterparts, we can hardly identify the difference. However, adversarial perturbations can actually lead to large prediction errors as we will illustrate in Section III-E.

B. ABERRANT SAMPLES DETECTION

The proposed aberrant data detector reports two types of outcomes. The basic detector only differentiates normal and abnormal data. The full detector further divides the detected abnormal images into adversarial and corrupted ones. We report our experimental outcomes in two parts. Within this subsection, we use cross-validation to determine the tuning parameters employed in the classification processes for three proposed aberrant scores, detailed below, and report their numerical performances. The classification accuracies obtained by different settings were illustrated with the Receiver Operating Characteristic (ROC) curves and the corresponding area under the curve (AUC). The description and the comparative outcomes to two existing aberrant data detectors are given in Subsection III-C. The normal face images, coupling with the corrupted and adversarial counterparts, were used in the evaluation processes throughout the rest of the paper. For the basic detector, users need to specify the portion of the normalized coefficients used in the calculation of the aberrant score. We first perform a tuning parameter selection study on P_{NC}, P_{window} under different settings. Next, the detection performance of the data detector is presented.

When calculating the aberrant scores using a given method (HIV, RV or MAV), one needs to specify the portion of the normalized coefficients used in the calculation (P_{NC}) or the window size to calculate the rolling variance (P_{window}). We conduct a 10-fold cross-validation (CV) study on different combinations of aberrant score calculation and classification threshold determination to select the tuning parameters. The normal images in the CV training dataset are used to calculate the principal components and the singular values. Using these principal components and singular values, we calculate the normalized coefficients and aberrant scores of each CV testing sample and compare the aberrant scores to *threshold*, which is determined by the specified classification threshold. The quality of the detection performances is shown by considering different combinations of aberrant score calculation and classification threshold determination. The procedures are described below.

1) CALCULATE THE ABERRANT SCORE

1) High-indexed variance (HIV): We calculate the variance of the high-indexed normalized coefficients as the aberrant score. The parameter P_{NC} specifies the portion of the high-indexed normalized coefficients used for

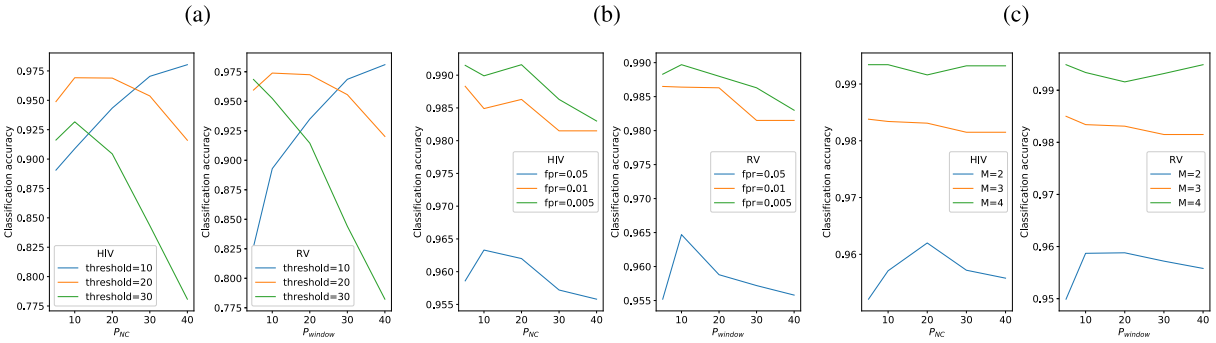


FIGURE 4. The classification accuracies under different settings. For each sub-figure, the plot on the left shows the results when the aberrant scores are calculated using HIV under different values of P_{NC} and the plot on the right shows the results when the aberrant scores are calculated using RV under different values of P_{window}. (a) The threshold is directly assigned (DA). (b) The threshold is calculated using the FPR method. (c) The threshold is determined using the Fence method.

calculation. We study the detection performance when $P_{NC} = 5, 10, 20, 30, 40$.

2) Rolling variance (RV): The rolling variance is calculated on the normalized coefficients with the window size defined by the parameter P_{window}, and the aberrant score is the maximum value of the rolling variance. We investigate the detection performance when $P_{window} = 5, 10, 20, 30, 40$.
3) Maximum absolute value (MAV): The aberrant score is the largest absolute value of the normalized coefficients. No extra tuning parameter is needed when using this method to calculate the aberrant score.

2) DETERMINE THE CLASSIFICATION THRESHOLD

1) Directly assign (DA): When this method is adopted, we specify the threshold directly by setting it to be 10, 20, 30.
2) False positive rate (FPR): By specifying the false positive rate, fpr, we will set the threshold as the $1 - fpr$ quantile of the training aberrant scores and evaluate the detection performance on the threshold. We set $fpr = 0.05, 0.01, 0.005$ for performance evaluation.
3) Fence: We calculate the mean, μ_{score}, and the standard deviation, σ_{score}, of the aberrant scores. The fence is defined by a tuning parameter M, which is a multiplier of the standard deviation, and is calculated as $fence = \mu_{score} + M \times \sigma_{score}$. We use the fence as the threshold and conduct studies when $M = 2, 3, 4$.

When using the DA approach, we use the normal training data and the abnormal training data together to evaluate the classification accuracy. In each fold of CV, the aberrant data detector is first built based on normal training images. Next, we calculate the classification accuracy using the CV normal testing images and their abnormal counterparts (corrupted and adversarial). To make the number of normal data and abnormal data the same, we double-weighted the normal data. As an example, if there are 60 CV testing data, we use 60 CV normal testing data with weight fraction 50%, 60 CV corrupted testing data with weight fraction 25% and 60 CV adversarial data with weight fraction 25% to calculate the classification accuracy. Fig. 4(a) shows the results for the DA approach. When more normalized coefficients are included to calculate the variance, i.e. large P_{NC} or large P_{window}, we obtain a smaller variance. Thus, we can obtain better classification accuracy when the threshold is small and P_{NC} (P_{window}) is large. We can obtain similar detection accuracy when $threshold = 10$ and $threshold = 20$ if P_{NC} and P_{window} are set to certain appropriate values. When $threshold = 30$, the detection accuracy is slightly lower if we use high-indexed normalized coefficients to calculate the variance. The selected parameters are shown in Table 1.

TABLE 1. The selected values of P_{NC} and P_{window} under different settings.

	$threshold$			fpr			M		
	10	20	30	0.05	0.01	0.005	2	3	4
P_{NC}	40	10	10	10	5	20	20	5	10
P_{window}	40	10	5	10	5	10	20	5	40

For the FPR and the Fence approaches, only the normal training images are used to build the aberrant data detector and to determine $threshold$. The results are shown in Fig. 4(b) and (c). Note that the lower the fpr is, the higher the threshold we would obtain, and thus the higher the accuracy we would obtain in detecting the normal images. Similarly, for a larger value of M in the Fence approach, we would set a larger value for the fence, which results in greater accuracy in classifying the normal images. The classification accuracy is not sensitive to the change in the values of P_{NC} or P_{window}. We selected the values of P_{NC} and P_{window} by CV, which are shown in Table 1.

3) RECEIVER OPERATING CHARACTERISTIC (ROC) CURVE FOR THE BASIC DETECTOR

To evaluate the overall detection performances, we construct the Receiver Operating Characteristic (ROC) curves using different approaches to calculate the aberrant score (HIV, RV and MAV). The area under the curve (AUC) are 0.9760, 0.9755 and 0.9697 for the three methods, respectively. The AUCs for HIV and RV are almost the same, and the AUC for

MAV is slightly smaller, but the results are still satisfactory. The AUC values suggest that the basic detectors built upon these three methods are all powerful tools to distinguish normal images from abnormal images.

C. DETECTION PERFORMANCE

The detection performance is evaluated using 100 testing images of three different kinds (normal, corrupted and adversarial). We implement the full version of the proposed aberrant detector and compare its detection performance to two existing approaches: the Coherence Pursuit (CoP) and Feature squeezing (FS). The details are described below.

1) Aberrant data detector (full version): Instead of classifying images into two categories (normal v.s. abnormal), we further divide the abnormal images into adversarial v.s. corrupted. The mechanism follows the same methodology except that if a query sample is identified as abnormal, we would perform spike replacement and classify the data point again. Our studies show that the performance is not sensitive to *cutoff*. Thus, we fix $cutoff = 30$. We use the same threshold as the criterion to differentiate corrupted and adversarial images. That is, we set $threshold' = threshold$. Through this extra step, we are able to classify the testing data into three categories. The detection is conducted using three proposed aberrant scores (HIV, RV and MAV) with the classification threshold being $threshold = 10$ (DA), $fpr = 0.05$ (FPR) and $M = 2$ (Fence). The tuning parameters are shown in Table 1.
2) CoP: Coherence pursuit (CoP) [25] is devised to discover robust principal components, which can facilitate detecting outliers. We follow the procedure in [25] and use 100 principal components for recovery with the recovery error as the classification criterion. For an input data x and its CoP estimated $\hat{x}$, the recovery error is defined as $||x - \hat{x}||_2/||x||_2$. The decision threshold is determined by the recovery errors obtained from the training dataset with the false positive rate equals 5%. CoP is designed to detect outliers. We evaluate its capability on differentiating normal and corrupted samples.
3) FS: Feature squeezing [34] is implemented for detecting adversarial entities. The model outputs could be largely different before and after applying feature squeezing. Thus, we use the normalized squared error of the model outcomes as the evaluation score. For an input, the normalized squared error is calculated as $||y - \hat{y}||_2/||y||_2$, where y and $\hat{y}$ are the model outcomes before and after feature squeezing. We use the joint score described in the paper, which summarizes the scores from different feature squeezing settings from 1-bit depth to 8-bit depth. The joint score is the maximum normalized squared errors obtained from different bit depths. We evaluate FS on its performance of detecting adversarial samples.

The detection accuracies is shown in Fig. 5. We observe that the detection accuracy of normal and adversarial samples

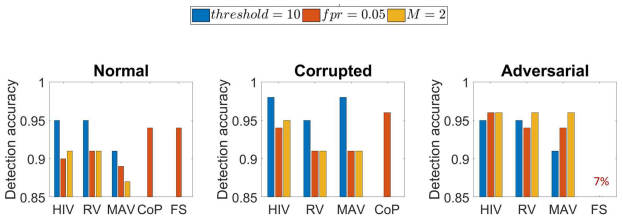


FIGURE 5. The detection accuracies of different types of the testing images when using different approaches to calculate the aberrant scores and threshold determination methods. Each type of the testing images contains 100 samples.

is lower when using MAV. This is because MAV uses only one value to represent the aberrant scores and cannot capture the variability of the normalized coefficients. The detection performances when using DA is generally the best. However, selecting the correct *threshold* for the DA method may be a time-consuming process. The FPR and the Fence method, though with slightly lower detection accuracies, can determine the threshold more generally and thus might be used as the methods to start with. The adv-corrupted data would be classified as corrupted. This type of data possesses both corrupted and adversarial patterns. After spike replacement is adopted, the large normalized coefficients are removed but the high variety coming from the corrupted parts still exist. The aberrant score would still be large and thus the adv-corrupted image would be classified as corrupted data.

On identifying corrupted data, the proposed aberrant detectors achieve similar detection accuracies to that of CoP. The FS's ability to detect adversarial data, even after excluding the corrupted samples, is surprisingly low. This seems to due to the fact that FS cannot determine if the high normalized squared errors are the consequences of adversarial perturbations or the errors brought-forward by the FS's modified inputs. Our proposed methods provide competitive and/or superior results in detecting both types of abnormal samples.

D. DATA RECONSTRUCTION

In this section we demonstrate the capability of reconstructing the corrupted images. Corrupted data are fixed using the following methods.

1) Corrupted data corrector: We use the proposed method to reconstruct a corrupted sample. Not knowing which pixels are damaged, we need a mechanism to only utilize the reliable information provided by regular pixels. In Fig 6, we compute the average squared differences of pixels between the corrupted images and their nearest neighbors in the training dataset. There is an abrupt change around the 90% quantile and thus we set $q = 0.9$. That is, we only calculate the similarity using 90% of the squared differences. The GLLiM inverse function is obtained with 20 clusters ($K = 20$) and 9 latent factors ($L_w = 9$) using the training dataset. Two scenarios are considered when applying corrupted data corrector. One is that no information

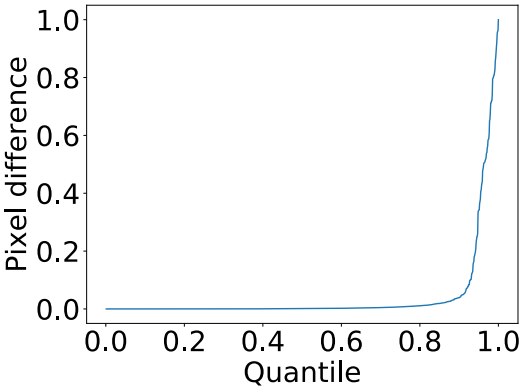


FIGURE 6. The sorted squared differences between corrupted images and their nearest neighbors in the training dataset.

about the damaged locations is available when fixing the corrupted images and the data corrector would reconstruct the whole images. The other is that the damage locations are known. Thus, we only replace pixels in the damage region after obtaining the reconstructed images. We report the outcomes for the former under ''Reconstructed'' and the latter under ''Supervised-reconstructed (S-Rec).''

2) NDI: New definition inpainting [6] utilizes a new priority definition to encourage geometry propagation. We use the default settings in this method to fix the corrupted region. NDI requires the information of the corrupted locations to be known.
3) PConv: In [16], partial convolution is used to avoid generating artifacts and leads to more realistic inpainting results. We use the pre-trained model to conduct inpainting. Like NDI, the information of the corrupted locations needs to be known in advance.

We conduct the experiment on 100 corrupted testing images and evaluate the reconstruction performance using the reconstruction mean squared error (Rec MSE), which is the average of the squared differences between the normal images and the reconstructed images over the damaged pixels. The Rec MSEs are 0.4410, 0.0064, 0.0064, 0.0653 and 0.0178 for Corrupted, Reconstructed, S-Rec, NDI and PConv, respectively. A figure in the Appendix shows a corrupted image and its reconstruction using different methods. We observe all the fixing approaches can effectively reduce the Rec MSE. The results obtained from the proposed corrupted data correctors (Reconstructed and S-Rec) outperform the other methods.

E. PREDICTION PERFORMANCE

Our proposed preprocessing system produces ''amended'' images that can be used by other prediction methods. A testing image is passed through the preprocessing system before applying the chosen prediction method. Depending on the identified testing data type, the preprocessing method would apply different mechanisms to the testing data. One can then apply the selected prediction method(s) to the processed data. We use the following prediction models to investigate the efficacies and generality of the proposed methods.

1) GLLiM: We use the GLLiM forward model for prediction. The forward model can be easily obtained from the inverse model we used for data reconstruction [5]. The GLLiM model is trained under $K = 20$, $L_w = 9$.
2) FGAM: Considering the predictor X and the scalar response Y, the functional generalized additive model (FGAM) [20] builds the relationship between X and Y as:

$$g(\mathbb{E}[Y|X]) = \beta_0 + \int F(X(t), t)dt, \tag{10}$$

where β_0 is the intercept, g is a known link function and F is an unspecified smooth function to be estimated. In our case, we set $g(x) = x$ and let t be the index of the image pixel. The function $F(\cdot, \cdot)$ is estimated through tensor-product B-splines with roughness penalties. FGAM is built using the R package *refund* [9]. We build the model on each dimension of Y using 100 knots, which leads to three FGAM models. We use the default values for the rest of the settings.
3) SAM: Similar to LASSO [30], the sparse additive model (SAM) [26] introduces the L_1 penalty to encourage sparse solutions on the functional coefficients. For the predictor X and the scalar response Y, SAM aims to find the solution that minimizes

$$\mathbb{E}\left\{Y - \sum_{d=1}^{D} \beta_d f_d(X_d)\right\}^2 \tag{11}$$

subject to

$$\sum_{d=1}^{D} |\beta_d| \leq P \tag{12}$$

$$\mathbb{E}[f_d^2] = 1, \tag{13}$$

where f_d is a function to be estimated, $\beta = (\beta_1, ..., \beta_D)^\top$ is a vector and P is a scalar constraint. The constraint of β imposes the sparsity of the estimated β. The model is trained using the R package *SAM* [37] under the default setting. We build a predictive model for each dimension of Y separately.

The prediction mean squared errors (PMSE) using different kinds of testing datasets are shown in Table 2. Each kind of testing dataset contains 100 testing images. Parts (a) and (b) of the table reconstruct the regular and ''adv-corrupted'' images, respectively. The PMSE of the reconstructed adv-corrupted images are referred to as ''Adv-corrupted Rec'' in Table 2. In addition, the 10%, 90% quantiles of prediction errors are presented to demonstrate the variation of each method's performance for almost all data samples.

The improvements on the reconstructed datasets demonstrate the benefits of adopting the preprocessing system. In part (a), the PMSE under ''Reconstructed'' is larger than the other reconstruction methods (S-Rec, NDI and NV) perhaps because the information of the damaged region

TABLE 2. The prediction performance of different types of images using different prediction models: GLLiM, FGAM and SAM. In each entry, the first number is the PMSE followed by the 10% and 90% quantiles of the prediction squared errors. (a) Results under Reconstrcuted, S-Rec, NDI and NV are obtained from reconstructing corrupted images using the specified approach, respectively. (b) The results of the same methods as in (a) for reconstructing Adv-corrupted images.

(a)

	Normal	Adversarial	Corrupted	Reconstructed	S-Rec	NDI	PConv
GLLiM	0.029 (0.002, 0.047)	0.263 (0.078, 0.593)	0.319 (0.027, 0.488)	0.044 (0.002, 0.096)	0.020 (0.002, 0.049)	0.040 (0.003, 0.101)	0.025 (0.002, 0.053)
FGAM	0.438 (0.041, 1.014)	0.545 (0.055, 1.266)	2.647 (0.347, 5.955)	0.476 (0.039, 0.780)	0.450 (0.044, 1.030)	0.675 (0.070, 1.423)	0.580 (0.049, 1.267)
SAM	0.120 (0.015, 0.241)	0.318 (0.042, 0.640)	0.390 (0.050, 0.794)	0.137 (0.015, 0.311)	0.127 (0.013, 0.222)	0.161 (0.014, 0.342)	0.159 (0.016, 0.340)

(b)

	Adv-corrupted	Reconstructed	S-Rec	NDI	PConv
GLLiM	0.727 (0.154, 1.265)	0.196 (0.027, 0.431)	0.203 (0.037, 0.429)	0.204 (0.033, 0.424)	0.193 (0.039, 0.439)
FGAM	2.698 (0.295, 6.224)	0.436 (0.027, 0.993)	0.567 (0.062, 1.325)	0.796 (0.077, 1.855)	0.719 (0.077, 1.538)
SAM	0.598 (0.076, 1.190)	0.277 (0.039, 0.604)	0.306 (0.045, 0.604)	0.342 (0.031, 0.672)	0.348 (0.060, 0.625)

TABLE 3. The prediction mean squared errors (PMSE) under different experimental settings. The Baseline column shows the original PMSE. The rest of the columns present the PMSE using different classification thresholds when the aberrant scores are calculated using HIV.

Pred. method	Baseline	$threshold = 10$	$fpr = 0.05$	$M = 2$
GLLiM	0.2280	0.0770	0.0812	0.0798
FGAM	0.8194	0.4371	0.4382	0.4375
SAM	0.1990	0.1497	0.1501	0.1498

is unknown. NDI and PConv require damaged locations information to conduct inpainting and perform well with the information available. With the damaged location information being known (S-Rec), we can obtain better PMSE compared to NDI and PConv. Improvements are also obtained in part (b) when the images are ''adv-corrupted''. Under this scenario, the prior knowledge about the damaged locations may or may not be helpful to our approach. Overall speaking, the proposed approach without using the prior knowledge about the damaged locations consistently well-perform. When the GLLiM prediction model is considered, the method PConv is also competitive. The 10% and 90% quantiles show that the proposed method is competitive for almost all of the data samples.

Note that the adversarial images are generated against the GLLiM forward model. However, the prediction loss is still large when the other two predictive methods are used, which implies the transferability of the adversarial examples [23].

F. OVERALL SYSTEM PERFORMANCE

To evaluate the overall performance of the proposed preprocessing system, we combine different types of images together. The testing dataset contains 100 normal images, 100 corrupted images, 100 adversarial images and 100 adv-corrupted images. Prediction models described in Section III-E would be used to conduct predictions. We weigh the normal and abnormal images equally. That is, when calculating the PMSE, the normal images would be calculated with weight fraction of 50%. As for the abnormal images, the PMSE will be calculated with weight fraction of 16.67% for each type of abnormal data. Table 3 shows the prediction results with ($threshold = 10$, $fpr = 0.05$, $M = 2$) and without (Baseline) the preprocessing system. The results are summarized when the aberrant scores are calculated using HIV. For each prediction method, the preprocessing system effectively reduces the prediction errors, which demonstrates that the proposed system is a general approach that can appropriately handle normal and abnormal testing entries. We can obtain the best prediction performance using the DA approach since the DA approach can provide better detection results. With a more accurate detection outcome, we would obtain a better ''amended'' testing data and thus the prediction performance is better.

IV. CONCLUSION AND FUTURE WORK

In this work, we proposed a preprocessing system that can detect abnormal data and provide ''amended'' samples. The proposed preprocessing system shows its ability to improve prediction performance in a model-agnostic manner. With the aberrant data detector and the aberrant data corrector, users can adopt their preferred prediction methods and obtain reliable outcomes even when the testing data are abnormal. For detecting different types of testing data, we propose three methods to determine the detection threshold. We suggest starting with the FPR or the Fence method and using the resulting detection threshold in the follow-up tuning process. The data reconstruction process is devised for corrupted data. Using the inverse regression learned by GLLiM, we can reconstruct the damaged data effectively. The overall performance of the proposed method is illustrated using three existing predictive models. We demonstrate the generality of the system and elucidate the necessity of the system for obtaining reliable prediction outcomes. For future work, we are interested in extending the framework to a more generalized scenario so that the proposed framework can identify and amend more types of irregular patterns.

APPENDIX A
SENSITIVITY TEST OF CUTOFF

The sensitivity test is conducted to investigate the selection of *cutoff*. Fig. 7 shows the detection accuracies against different settings of *cutoff*. The detection accuracies are calculated using 100 corrupted and 100 adversarial images. The detection threshold is directly set to 10 when using the DA method. We use $fpr = 0.05$ and $M = 2$ for the FPR and the Fence method, respectively. From the results, we see that the detection accuracy is insensitive to the choice of *cutoff*.

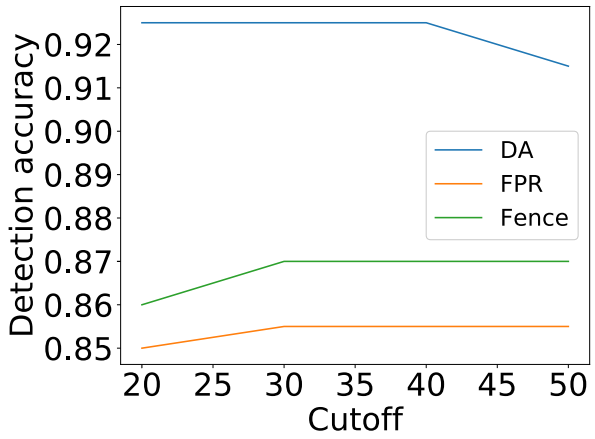


FIGURE 7. The detection accuracy of 100 corrupted images and 100 adversarial images using different *cutoff* values. For three threshold determination methods, we set *threshold* = 10 for the DA method, *fpr* = 0.05 for the FPR method and *M* = 2 for the Fence method.

FIGURE 8. Examples of the corrupted image and their reconstructed results using different methods. The Rec MSE for this example is 0.4002, 0.0034, 0.0034, 0.0528 and 0.0256 for Corrupted, Reconstructed, S-Rec, NDI and PConv, respectively.

APPENDIX B
RECONSTRUCTION RESULTS

Figure 8 shows an example of the corrupted image and their reconstructed images using different methods. We observe that the proposed method (Reconstructed and S-Rec) can generate more realistic results. Both NDI and PConv try to fill the missing parts but lose the important information (eye).

REFERENCES

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, Aug. 2009.

[2] A. N. Bhagoji, D. Cullina, and P. Mittal, "Dimensionality reduction as a defense against evasion attacks on machine learning classifiers," 2017, *arXiv:1704.02654v2*. [Online]. Available: https://pdfs.semanticscholar.org/b05e/86841ca65f4ba483b04e465fd54984ad6306.pdf

[3] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 3–14.

[4] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proc. Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, Jun. 2017, pp. 90–98.

[5] A. Deleforge, F. Forbes, and R. Horaud, "High-dimensional regression with Gaussian mixtures and partially-latent response variables," *Statist. Comput.*, vol. 25, no. 5, pp. 893–911, Sep. 2015.

[6] L.-J. Deng, T.-Z. Huang, and X.-L. Zhao, "Exemplar-based image inpainting using a modified priority definition," *PLoS ONE*, vol. 10, no. 10, Oct. 2015, Art. no. e0141199.

[7] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[8] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, Oct. 2016.

[9] J. Goldsmith, F. Scheipl, L. Huang, J. Wrobel, J. Gellar, J. Harezlak, M. W. McLean, B. Swihart, L. Xiao, C. Crainiceanu, P. T. Reiss, Y. Chen, S. Greven, L. Huo, M. G. Kundu, S. Y. Park, D. L. Miller, and A.-M. Staicu. (2018). *Refund: Regression With Functional Data*. [Online]. Available: https://CRAN.R-project.org/package=refund

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, Mar. 2015, pp. 1–11.

[11] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, Jan. 2012.

[12] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," Feb. 2017, *arXiv:1702.06280*. [Online]. Available: https://arxiv.org/abs/1702.06280

[13] D. Hendrycks and K. Gimpel, "Early methods for detecting adversarial images," in *Proc. Int. Conf. Learn. Represent. (Workshop Track)*, Mar. 2017, pp. 1–9.

[14] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, p. 107, Jul. 2017.

[15] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, pp. 1–13, 2017. [Online]. Available: https://link.springer.com/article/10.1007/s10586-017-1117-8

[16] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2018, pp. 89–105.

[17] P.-H. Lu, P.-Y. Chen, K.-C. Chen, and C.-M. Yu, "On the limitation of MagNet defense against L1-based adversarial examples," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2018, pp. 200–214.

[18] P.-H. Lu, P.-Y. Chen, and C.-M. Yu, "On the limitation of local intrinsic dimensionality for characterizing the subspaces of adversarial examples," in *Proc. Int. Conf. Learn. Represent. Workshop*, 2018, pp. 1–5.

[19] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Int. Conf. Mach. Learn.*, Jun. 2009, pp. 689–696.

[20] M. W. McLean, G. Hooker, A.-M. Staicu, F. Scheipl, and D. Ruppert, "Functional generalized additive models," *J. Comput. Graph. Statist.*, vol. 23, no. 1, pp. 249–269, 2014.

[21] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2017, pp. 135–147.

[22] M. M. Moya and D. R. Hush, "Network constraints and multi-objective optimization for one-class classification," *Neural Netw.*, vol. 9, no. 3, pp. 463–474, 1996.

[23] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: From phenomena to black-box attacks using adversarial samples," May 2016, *arXiv:1605.07277*. [Online]. Available: https://arxiv.org/abs/1605.07277

[24] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2536–2544.

[25] M. Rahmani and G. K. Atia, "Coherence pursuit: Fast, simple, and robust principal component analysis," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6260–6275, Dec. 2017.

[26] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman, "Sparse additive models," *J. Roy. Stat. Soc., B*, vol. 71, no. 5, pp. 1009–1030, 2009.

[27] A. Roy, J. Singha, S. S. Devi, and R. H. Laskar, "Impulse noise removal using SVM classification based fuzzy filter from gray scale images," *Signal Process.*, vol. 128, pp. 262–273, Nov. 2016.

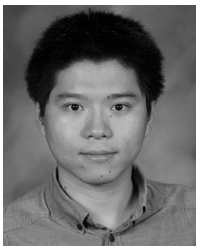
[28] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," Dec. 2013, *arXiv:1312.6199*. [Online]. Available: https://arxiv.org/abs/1312.6199

[29] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[30] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., B*, vol. 58, no. 1, pp. 267–288, 1996.

[31] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, "AutoZOOM: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, pp. 742–749.

[32] D. Wang, D. S. Yeung, and E. C. C. Tsang, "Structured one-class classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1283–1295, Dec. 2006.
[33] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 1, pp. 78–80, Jan. 1999.
[34] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," Apr. 2017, *arXiv:1704.01155*. [Online]. Available: https://arxiv.org/abs/1704.01155
[35] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6721–6729.
[36] Z. Yang, B. Li, P.-Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," in *Proc. Int. Conf. Learn. Represent.*, Mar. 2019, pp. 1–15.
[37] T. Zhao, X. Li, H. Liu, Y. Ma, H. Jiang, and K. Roeder. (2014). *SAM: Sparse Additive Modelling*. [Online]. Available: https://CRAN.R-project.org/package=SAM



CHUN-CHEN TU received the B.S. degree in electrical engineering and computer science and the M.S. degree in electronics engineering from National Chiao Tung University, Taiwan, in 2010 and 2012, respectively, and the Ph.D. degree in statistics from the University of Michigan, Ann Arbor, USA, in 2019.

His research interests include model robustness, signal processing, high-dimensional data analysis on mixture models, adversarial machine learning, and interpretability for neural networks and signal processing on the audio data.

PIN-YU CHEN received the B.S. degree in electrical engineering and computer science (undergraduate honors program) from National Chiao Tung University, Taiwan, in 2009, the M.S. degree in communication engineering from National Taiwan University, Taiwan, in 2011, and the Ph.D. degree in electrical engineering and computer science and the M.A. degree in statistics from the University of Michigan, Ann Arbor, USA, in 2016. He is currently a Research Staff Member with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. He is also affiliated with the MIT-IBM Watson AI Lab and is a Co-PI of MIT-IBM projects.

His current research interests include adversarial machine learning, robustness analysis of neural networks, graph and network data analytics and their applications to data mining, machine learning, signal processing, and cyber security. He was a recipient of the Chia-Lun Lo Fellowship from the University of Michigan Ann Arbor, the NIPS 2017 Best Reviewer Award, and of the IEEE GLOBECOM 2010 GOLD Best Paper Award and several travel grants, including IEEE ICASSP 2014 (NSF), IEEE ICASSP 2015 (SPS), IEEE Security and Privacy Symposium, NSF Graph Signal Processing Workshop 2016, and ACM KDD 2016. He is on the Editorial Board of *PLOS One*.

NAISYIN WANG received the B.Sc. degree in mathematics from Tsing-Hua University, Taiwan, the M.Sc. degree in applied statistics from The Ohio State University, USA, and the Ph.D. degree in statistics from Cornell University, USA. She is currently a Professor of statistics with the University of Michigan, USA. Her research interests include mixture models, longitudinal and functional data analysis, handling complications caused by outliers, missing data, and measured with errors observations and non-/semi-parametric models. She is an Elected Fellow of IMS, ASA, and AAAS.