Interactive Anomaly Identification with Erroneous Feedback

Takaaki Tagawa[1], Yukihiro Tadokoro[2], and Takehisa Yairi[3]

[1,2] *Toyota Central R&D Labs., Inc., Aichi, Japan.*
tagawa@mosk.tytlabs.co.jp
y.tadokoro@ieee.org

[3] *Research Center for Advanced Science and Technology, University of Tokyo, Tokyo, Japan.*
yairi@ailab.t.u-tokyo.ac.jp

ABSTRACT

The difficulties in analyzing large and extensive systems necessitate the use of efficient machine-learning tools to identify unknown system anomalies in order to avoid critical problems and ensure high reliability. Given that data logged by a system include unknown anomalies, anomaly identification models aim to simultaneously identify the time of occurrence and the features that contributed to these anomalies. To maximize accuracy, it is important to utilize the data as well as the domain knowledge of the system. However, it is difficult for a system analyst to possess not only machine-learning capabilities but also domain knowledge to incorporate into the model. In this paper, we propose a new anomaly identification framework capable of utilizing feedback based on domain knowledge without requiring any machine-learning capabilities. We also propose a novel method, the so-called rank ensemble method, to improve the accuracy of anomaly identification with erroneous feedback, that is, feedback that includes incorrect information. Our method enables erroneous information to be adaptively ignored by assuming consistency between the data and the user feedback. An intensive parameter study using benchmark datasets and a case study with real vehicle data demonstrate the applicability of our framework.

1. INTRODUCTION

Recent progress in system development technologies has enabled the construction of complex and extensive systems such as autonomous driving, factory automation, and computer network systems. Moreover, each of these systems frequently includes one or more subsystems; for example, an autonomous driving system consists of many modules, including those performing environmental sensing, decision-making, and control. These modules are interconnected in a complex manner, run in parallel, and are variously controlled to be versatile and have advanced features. Eventually, each of these systems becomes highly complex and requires expert knowledge to construct a prognostics and health management system to ensure reliability. In practice, when unknown anomalies occur within a system, it is important to incorporate countermeasures into the prognostics and health management system to prevent recurrence. Engineers accomplish this by analyzing the data logged at the approximate time of observation of the anomalies to identify the exact timing and causes of such anomalies. However, this data analysis is considered to be a difficult and costly process that requires efficient tools from various fields.

Consequently, users have been demanding cost-effective data-driven methods for investigating these systems. When unknown anomalies are observed within a system, data-driven methods automatically analyze the corresponding data logged by the system to identify the time when the anomalies occurred and the features that contributed to them. These methods are cost-effective because they require neither human resources nor specific knowledge of the system.

However, most existing data-driven anomaly detection methods have three main drawbacks (see Section 2 for further information):

1. Difficulty in identifying causes of anomalies.

In the machine-learning community, anomaly detection methods are commonly used to determine the occurrence of anomalies (see (Chandola, Banerjee, & Kumar, 2009) and references therein). However, this information is insufficient for users who require information about the features that actually caused the anomalies. Identifying the cause is important to enable users to implement countermeasures to prevent recurrence. A method that determines the occurrence of an anomaly and its contributory features is referred to as an

anomaly *identification* method. Methods capable of conducting anomaly identification, for example, (Candès, Li, Ma, & Wright, 2011) have been proposed; however, they exhibit limited performance owing to reasons that include the difficulty of incorporating domain-specific knowledge.

2. Limitation in utilizing domain knowledge.

The incorporation of domain knowledge into data-driven methods has limitations. It is often difficult for an analyst in one domain to have both machine-learning knowledge and domain knowledge. In addition, analysts have difficulty in introducing precise information into the models because they do not know enough about the anomaly itself. Models that rely on incorrect information will produce faulty estimations.

3. Lack of interactivity with users.

In principle, the process of anomaly identification is exploratory and interactive. The discovery of causality requires that analysts specify a suspicious part of a system and investigate it. If the part is determined to be normal, the analyst then suggests another part of the system to be analyzed based on the knowledge obtained during the initial stage of the investigation. In this case, specifying the suspicious part containing an unknown anomaly is often difficult and incorrectly specifying a part causes excessive trial-and-error problem solving. Data-driven anomaly detection methods are expected to help users with this exploratory process by automatically suggesting a suspicious part as well as by accepting feedback from users to refine the suggestions for the next trial. However, anomaly detection methods often analyze the data only once and are incompatible with the exploratory and interactive aspects of anomaly identification.

In this study, we propose a novel framework to overcome these three drawbacks simultaneously. Figure 1 illustrates the process of our framework. The process starts when an analyst notices something anomalous in a target system and logs the data at the approximate time when the anomalies were observed. Initially, the analyst does not know exactly when the anomaly occurred or what contributed to it. First, the framework conducts anomaly identification using only data based on a sparse and low-rank model (Candès et al., 2011). Our proposed approach shows that this method can be extended to incorporate user knowledge in a simple manner, that is, users only provide feedback on whether the predicted anomalies are indeed anomalous. Subsequently, the anomaly identification procedure is refined without requiring difficult and costly manual modifications. This process continues interactively until the users complete the analysis, and this corresponds well with the exploratory and interactive nature of anomaly identification. The experimental results, which are based on a variety of datasets, showed that our framework successfully

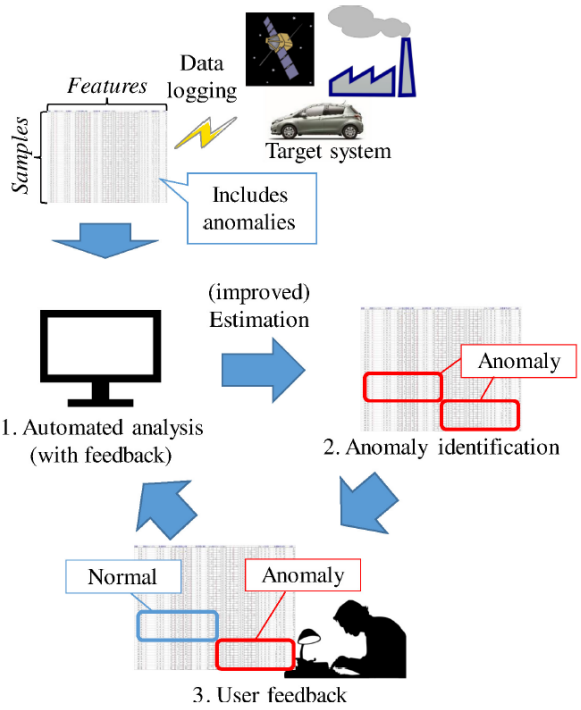

Figure 1. Interactive anomaly identification framework.

improves the accuracy of anomaly identification.

In addition, we construct a novel method based on our framework to improve its robustness to erroneous feedback, that is, feedback that includes incorrect information. The framework allows approximate labeling by users, thereby reducing the cost to users of providing precise feedback. In practice, our novel method, a so-called rank ensemble method inspired by (Parikh, Saluja, Dyer, & Xing, 2014), evaluates the low-rank representation of the normal part of the data. We observed that if the feedback is incorrect, the corresponding model representation based on such feedback tends to be inconsistent with the data distribution. Our method averages between low- and high-rank representations to obtain a new representation that expresses only the data that is consistent with both the data distribution and user feedback. Thus, the method is able to adaptively ignore only inconsistent parts. The experimental results demonstrate that our framework improves the accuracy of anomaly identification, even if the user feedback includes incorrect information. In addition, a case study using real vehicle data demonstrates the use of our framework by an analyst.

The remainder of this paper is structured as follows. Related studies are first described in Section 2. The problem setting presented in Sections 3 and 4 summarizes anomaly identification using a sparse low-rank method that forms the basis of our framework. In Section 5, the method is extended to propose a novel framework that accepts user feedback, and the

rank ensemble method to process erroneous feedback is introduced. Experiments are described in Section 6 to demonstrate the improvements achieved with the feedback and the robustness against user feedback that includes incorrect information. A case study experiment using real driving data is presented in Section 7 to demonstrate the use of our framework in a realistic setting. Finally, conclusions and future work are provided in Sections 8 and 9, respectively.

2. RELATED WORK

This section summarizes the published work related to anomaly identification methods.

In the machine-learning community, methods to identify the occurrence of an anomaly have been proposed for a number of applications, such as anomaly, novelty, fault, change, fraud, and intrusion detection (see (Chandola et al., 2009) and references therein). The methods include, for example, level-set estimation methods ((Scott & Nowak, 2006), (Hero, 2007), (Zhao & Saligrama, 2009), and (Sricharan & Hero, 2011)), local density-based methods ((Breunig, Kriegel, Ng, & Sander, 2000), (Papadimitriou, Kitagawa, B. Gibbons, & Faloutsos, 2003), and (Kriegel, Kröger, Schubert, & Zimek, 2009)), and discriminative methods ((Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001) and (Tax & Duin, 2004)). Usually, these methods find samples in a low-density region and detect them as anomalies. Although they are typically nonparametric models with the ability to detect the existence of anomalies in a complex data structure, they cannot identify the features that contributed to the anomalies. This makes it difficult for users to interpret why the detected samples are considered anomalies.

Reconstruction-based methods have been proposed to simultaneously identify when anomalies occur and which features contribute to them. They are usually methods based on dimensionality reduction such as principal component analysis (PCA), a mixture of probabilistic PCA (Yairi et al., 2017)(Tipping & Bishop, 1999), robust PCA(Candès et al., 2011)(Lin, Chen, & Ma, 2010), and neural networks(Subba, Biswas, & Karmakar, 2016)(Tagawa, Tadokoro, & Yairi, 2015). These methods first learn a low-dimensional representation of normal data, which is provided in addition to data with observed anomalies. The extent to which the sample deviates from normal behavior is represented by the distance between the learned representation and the sample, and those samples with a large distance are detected as anomalies. The distance is also obtained for each feature, and features with a large distance can be considered candidates for the causes of the anomalies. However, such methods require normal data, which are costly to obtain as human experts must verify the normality of the data.

Certain methods, such as matrix completion(Sindhwani, Bucak, Hu, & Mojsilovic, 2010)(Hsieh, Natarajan, & Dhillon, 2014), collaborative filtering(Su & Khoshgoftaar, 2009), and network traffic anomaly detection(Mardani, Mateos, & Giannakis, 2013b)(Mardani, Mateos, & Giannakis, 2013a), are able to utilize users' knowledge. However, they require precise information such as the correct labeling of all elements (an element value is either missed or not missed) or the network structure of the system. Such information is difficult to obtain with a complex black-box system, for which only ambiguous information is available.

Several methods proposed by the natural language processing community, such as (Raghavan, Madani, & Jones, 2006), (Elahi, Ricci, & Rubens, 2014), and (Settles, 2011), rely on interactive processes to utilize user knowledge. These methods iteratively select and require users to provide labels for the selected samples (items) and features (attributes) that are expected to improve the accuracy of the classification model the most. However, these methods are based on active learning methodologies; thus, they select and ask for labels for the samples and features that are the most uncertain in terms of the classification model. In contrast, the anomaly identification process aims to find anomalies but not uncertainties.

Among the above-mentioned studies, there are no methods capable of overcoming all three drawbacks introduced in Section 1. Therefore, to the best of our knowledge, this study is the first to propose a framework to overcome these drawbacks simultaneously.

3. PROBLEM SETTING

Let $\mathbf{d} = \{d_1, \ldots, d_K\}^T \in \mathbb{R}^K$ be K-dimensional features, and let $D = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$ be a $K \times N$ observation data matrix with N samples. For simplicity, we denote D_{ij} as the ith feature d_i of the jth sample $\mathbf{d}_j$, where $i \in 1, \ldots, K$, and $j \in 1, \ldots, N$. For example, in the case of a vehicular system, the K features data sampled over time by the vehicle's speed sensor, the engine rotation speed sensor, or the fuel economy metric (e.g., kilometers per liter). The observation dataset D is standardized such that each feature has a zero mean and a unit variance with respect to the N samples.

As described in Section 1, we focus on a situation where an analyst notices anomalous behavior in a target system and logs the data at the approximate time when the anomalies were supposed to be observed. Thus, given a dataset D logged from a system by observing unknown anomalies, it is assumed that D includes several anomalies. The problem to be solved is formulated as follows.

Problem: Given a dataset D containing unknown observed anomalies, identify the samples that include the anomalies, as well as the features that contributed to the anomalies.

Expressed differently, let $I_D = \{(i, j)|$, where the ith feature of the jth sample is the anomaly$\}$, and the problem requires

I_D to be identified accurately. We refer to this problem as an anomaly identification problem. The following two assumptions are given with the problem:

Assumption 1: Anomalies exist in only a few samples and features within the data, i.e., ($|I_D| \ll |D|$).

Assumption 2: Anomalies deviate from the normal behavior of the data.

Assumption 1 holds because if anomalies were dominant in D, they would be easy to identify by humans; hence, a data-driven analysis would be unnecessary. In addition, if Assumption 2 does not hold, it would no longer be possible to distinguish anomalous behavior from normal behavior based on deviations between anomalous and normal samples over feature space $\mathbf{d}$ of the data. The next section provides a detailed definition of deviation. Assumptions 1 and 2 are therefore fundamental in data-driven anomaly detection (Chandola et al., 2009).

4. OVERVIEW OF SPARSE AND LOW-RANK REPRESENTATION-BASED ANOMALY IDENTIFICATION

Considering the related work introduced in Section 2, we infer that among the existing methods, the robust PCA method (Candès et al., 2011) is the most compatible with our problem setting introduced in Section 3 for the following reasons. The method assumes that the anomalous part of the data is sparse (i.e., $|I_D| \ll |D|$); thus, the method is compatible with Assumption 1. The method also assumes a low-rank property for the normal part of the data. This property is based on the fundamental principle of dimensionality reduction methods that assumes normal data to be distributed on a low-dimensional subspace; thus, it is acceptable. The method is based on an inexact augmented Lagrange multiplier optimization with guaranteed convergence (Lin et al., 2010) and is computationally efficient for identifying anomalies. Although the method assumes linearity on the low-dimensional subspace, it is still acceptable as the given data are logged only at the approximate time that the anomalies occurred, the size of the dataset is not arbitrarily large, and it does not exhibit very strong nonlinearity. In addition, the method can accept label information to address the matrix completion problem, although the information must be correct and assigned to all the elements (Lin et al., 2010).

Based on this method, we constructed a novel framework capable of accepting user knowledge in an exploratory and interactive manner, as introduced in Section 5. In this section, we summarize the so-called sparse and low-rank representation method according to (Candès et al., 2011) and (Lin et al., 2010).

4.1. Sparse and Low-Rank Representation

Given D, including a few anomalies indicated by I_D, a sparse and low-rank method models $D = A + E$, where $A \in \mathbb{R}^{K \times N}$ is a low-rank matrix of $\text{rank}(A) \leq r \ll \min(K, N)$ with $r \in \mathbb{N}$, and $E \in \mathbb{R}^{K \times N}$ is a sparse matrix with the (i, j)th element $E_{ij} \neq 0$ if $(i, j) \in I_D$, or $E_{ij} = 0$ otherwise. As A represents the normal part of D, A is assumed to be a low-rank matrix. The matrix E represents an anomalous part of the matrix D and contains data that deviate from the normal part, A. According to Assumption 1, the (i, j)th element $E_{i,j}$ has a nonzero value only if $(i, j) \in I_D$; otherwise, $E_{ij} = 0$. Note that our goal in anomaly identification is to appropriately estimate E to enable identification of I_D.

The estimation of the two matrices A and E from data D requires the following optimization problem to be solved (Candès et al., 2011).

$$\begin{aligned} &\min_{A,E} \|E\|_0, \\ &\text{s.t.} \quad \text{rank}(A) \leq r, \quad D = A + E, \end{aligned} \tag{1}$$

where $\|\cdot\|_0$ is the l_0 norm. This problem requires the minimization of $\|E\|_0$ to obtain the sparse matrix E while preserving the low-rank property $\text{rank}(A) \leq r$. As problem (1) is difficult to solve, the following proximal problem is solved instead, according to (Candès et al., 2011)(Lin et al., 2010).

$$\begin{aligned} &\min_{A,E} \|A\|_* + \lambda \|E\|_1 \\ &\text{s.t.} \qquad D = A + E, \end{aligned} \tag{2}$$

where $\|\cdot\|_*$ is the nuclear norm, $\|\cdot\|_1$ is the l_1 norm, and λ is a tradeoff parameter.

4.2. Finding the Sparse Matrix E Based on an Augmented Lagrange Multiplier Method

Several methods have been proposed to solve the proximal problem (2) (refer to (Candès et al., 2011) and references therein). The inexact augmented Lagrange multiplier method (Lin et al., 2010), which can efficiently obtain A and E with a valid convergence guarantee, is adopted in this study. The method is summarized as follows.

Algorithm 1 shows the process of optimizing the two matrices A and E. $A^{(t)}$ and $E^{(t)}$ represent the updated values of A and E at each iterative step t, respectively. $\lambda, \{\mu^{(0)}\} \in \mathbb{R}, \alpha > 1$ are the parameters to be set beforehand. The algorithm iteratively updates the matrices $A^{(t)}$ and $E^{(t)}$ until convergence is reached.

At line 4, $A^{(t+1)}$ is calculated based on the following optimization with singular value decomposition (SVD) and the soft-thresholding operator $\mathcal{S}_\epsilon[x]$, which is defined in (5) of

Algorithm 1 Inexact Augmented Lagrange Multiplier Method (IALM) (Lin et al., 2010)

Require: $D, \lambda, \alpha > 1, \{\mu^{(0)}\}$.
1: Initialize $A^{(0)} = O, E^{(0)} = O, t = 0$.
2: **repeat**
3: $W_A = D - E^{(t)} + Y^{(t)}/\mu^{(t)}$.
4: $A^{(t+1)} = L_A(D, E^{(t)}, \mu^{(t)}, W_A)$.
5: $W_E = D - A^{(t+1)} + Y^{(t)}/\mu^{(t)}$.
6: $E^{(t+1)} = L_E(D, A^{(t+1)}, \mu^{(t)}, W_E)$.
7: $Y^{(t+1)} = Y^{(t)} + \mu^{(t)}(D - A^{(t+1)} - E^{(t+1)})$.
8: $\mu^{(t+1)} = \alpha\mu^{(t)}$.
9: $t = t + 1$.
10: **until** $A^{(t)}, E^{(t)}$ converge.
11: **return** $A = A^{(t)}, E = E^{(t)}$.

(Lin et al., 2010).

$$L_A(D, E^{(t)}, \mu^{(t)}, W_A) = \arg\min_{X \in \mathbb{R}^{K \times N}} \frac{1}{\mu^{(t)}}\|X\|_* + \frac{1}{2}\|X - W_A\|_F^2, \quad (3)$$

$$= U\mathcal{S}_{1/\mu_t}[S]V^T, \quad (4)$$

$$USV^T = W_A, \quad (5)$$

where U, S, and V are the left singular vector matrix, diagonal singular value matrix, and right singular vector matrix of W_A, respectively. The optimization provides $A^{(t+1)}$, which corresponds well with W_A while preserving the low-rank constraint $\|X\|_*$.

In the next step (line 6), the matrix $E^{(t)}$ is updated as follows:

$$L_E(D, A^{(t+1)}, \mu^{(t)}, W_E) = \arg\min_{X \in \mathbb{R}^{K \times N}} \frac{1}{\mu^{(t)}}\|X\|_1 + \frac{1}{2}\|X - W_E\|_F^2, \quad (6)$$

$$= \mathcal{S}_{\lambda/\mu_t}[W_E], \quad (7)$$

where $\|\cdot\|_F$ is the Frobenius norm. L_E yields $E_{(t+1)}$, which corresponds well with W_E while preserving the sparsity constraint $\|X\|_1$.

4.3. Limitations of Algorithm 1

Although Algorithm 1 obtains A and E efficiently with a valid convergence guarantee, it continues to present difficulties because the result still contains identification errors. This limited performance is mainly attributed to it being difficult to achieve a global optimum solution using Algorithm 1, that is, to ensure that the nonzero part of E completely indicates I_D. The reasons for this are that sparse and low-rank assumptions do not always completely fit an arbitrary data distribution, and the data often include noisy samples that deviate from a normal distribution even though they are not anomalous.

In this regard, users' domain-specific knowledge helps to overcome such problems. However, incorporating such knowledge usually requires the costly manual construction of a model. Instead, we propose a novel framework to automatically modify the model to include domain-specific knowledge based on iterative user feedback.

5. PROPOSED INTERACTIVE FRAMEWORK FOR ANOMALY IDENTIFICATION

This section proposes a framework that utilizes user feedback to interactively improve the accuracy of anomaly identification. Our framework is based on the sparse and low-rank model introduced in Section 4. The proposed model is automatically refined by incorporating user feedback to provide improved anomaly identification estimates to the user in the next step. We also propose a method to improve the model performance with incorporated feedback that may contain incorrect information but can be easily provided by users.

This process inherits interactiveness because it exchanges information with users as follows: 1) our model provides information about anomaly candidates to users given the user feedback thus far, and 2) users provide feedback to our model given insights from datasets via our model estimations. This interactive process continues to improve our model estimations and users' understanding of anomalies until the user completes the analysis.

5.1. Outline of Our Framework

Algorithm 2 outlines our framework. The details are described in the following sections and here we briefly describe the framework. Assume that an analyst observed unknown anomalies in a target system and logged data at the approximate time that the unknown anomalies were supposed to have occurred, or a large-scale anomaly detection method, for example, (Pham, Venkatesh, Lazarescu, & Budhaditya, 2014), determines the approximate time at which an anomaly occurred. Our framework operates offline and accepts the logged data

Algorithm 2 Interactive Framework With Expert Knowledge Feedback

Require: $D, \lambda, \alpha > 1, \{\mu^{(0)}\}, M$
1: Let $k = 0$ and initialize $\hat{A}^{(0)}, \hat{A}^{(0)}$ by the output A, E of Algorithm 1.
2: **repeat**
3: A set of (i, j) elements, the labels of which are unknown, with the top M values of $|\hat{E}_{ij}^{(t)}|$ is presented to users as anomaly candidates.
4: The users provide labels L_n, L_a indicating whether the M elements are normal, abnormal, or unknown.
5: Obtain $\hat{A}^{(t+1)}, \hat{E}^{(t+1)}$ by the output A, E of Algorithm 1 constrained by L_n, L_a (see 5.2.1 and 5.2.2 for details).
6: $t = t + 1$.
7: **until** user stops analysis
8: **return** $\hat{A} = \hat{A}^{(t)}, \hat{E} = \hat{E}^{(t)}$.

as input interactively to identify the actual timing and cause of the anomalies. Let D be the input data with observed anomalies, and let $\hat{A}^{(t)}, \hat{E}^{(t)}$ be the estimations of normal and anomalous parts of D at the tth iteration of Algorithm 2. The process first initializes $\hat{A}^{(0)}, \hat{E}^{(0)}$ by the output A, E of Algorithm 1 (line 1). Recall that $\hat{E}^{(t)}$ represents a deviation from normal behavior. We assume that the larger $|\hat{E}_{ij}^{(t)}|$ is, the higher the likelihood of the ith feature of the jth sample being an anomaly. Therefore, we use $|\hat{E}_{ij}^{(t)}|$ as the anomaly rate of the ith feature of the jth sample. Our framework adopts a set of (i, j) elements containing the top $M \in \mathbb{N}$ values of $|\hat{E}_{ij}|$ as candidates for anomalies (line 3). The value of M is set manually.

Given M candidates, users can provide label matrices $L_n, L_a \in \{0, 1\}^{K \times N}$ defined as follows:

$$L_{n,ij} = \begin{cases} 1, & \text{if } (i,j) \text{ is labeled normal} \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

$$L_{a,ij} = \begin{cases} 1, & \text{if } (i,j) \text{ is labeled an anomaly} \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

where $L_{n,ij} and L_{a,ij}$ denote the (i, j)th element of $L_n and L_a$, respectively. According to the feedback L_n, L_a, we can obtain $\hat{A}^{(t+1)}, \hat{E}^{(t+1)}$ by introducing a novel optimization method that extends Algorithm 1 and utilizes the user feedback L_n, L_a as constraints (line 5). The new $\hat{A}^{(t+1)}, \hat{E}^{(t+1)}$ is used to conduct the next iteration of anomaly identification until the users complete their analysis.

Our framework shows the users the top M anomalous elements to conduct anomaly identification and simultaneously to elicit their feedback for model refinement. We assume that users do not ignore the presented M anomaly candidates and conclude whether they are actually anomalies. Thus, it is cost-effective to also request feedback according to the conclusion. However, users are not restricted from providing feedback other than for the top M elements.

5.2. Method to Incorporate Complete Feedback

In this section, we introduce user feedback $L_{n,ij}$ and $L_{a,ij}$ to refine $\hat{A}^{(t)}, \hat{E}^{(t)}$ to $\hat{A}^{(t+1)}, \hat{E}^{(t+1)}$. For simplicity, this section only considers *complete* feedback that never includes incorrect information. Although this setting may be unrealistic, it yields several insights into how user knowledge can be utilized to improve the accuracy of anomaly identification in our proposed framework. Section 5.3 considers a more natural setting in which the user feedback includes incorrect information.

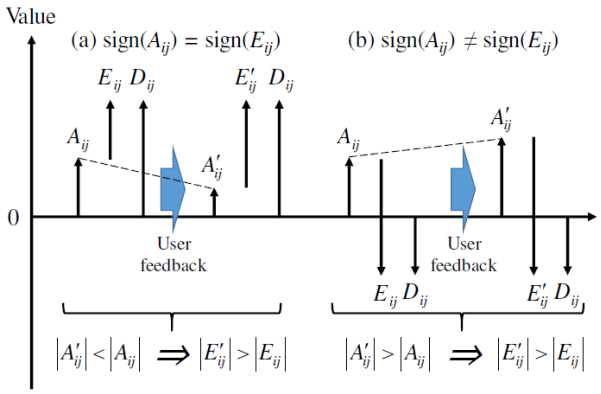

Figure 2. Conditions under which the two patterns of anomalies are emphasized based on their signs, where A_{ij}, E_{ij} and A'_{ij}, E'_{ij} are the outputs of Algorithm 2 without and with the constraints by L_a, respectively, $\text{sign}(x) = 1$ if $x \geq 0$, and $\text{sign}(x) = -1$ otherwise.

5.2.1. Updating A With the Feedback of Anomaly Label L_a

This section introduces a strategy for incorporating user feedback on anomalies according to L_a. The strategy is to constrain Algorithm 1 to ensure that (i, j) elements satisfying $L_{a,ij} = 1$ have a larger anomaly rate $|E_{ij}|$. Constraining these labeled elements can affect the overall output A, E because the computation is based on SVD, as shown in Section 4. The estimated E has a large $|E_{ij}|$ if the (i, j) element shows a pattern similar to the anomalies denoted by the user feedback L_a. Therefore, a strategy that incorporates user feedback by encouraging the detection of anomalous patterns given by L_a improves the accuracy of anomaly identification.

We categorize the anomalies that satisfy $L_{a,ij} = 1$ according to the signs of A_{ij} and E_{ij}, and explain how to emphasize them. Figure 2 shows the two patterns of anomalies for the (i, j)th element. Let A, E and A', E' be the outputs of Algorithm 1 without and with the constraints by L_a, and let $\text{sign}(x) = 1$ if $x \geq 0$, and $\text{sign}(x) = -1$ otherwise. Pattern (a) represents the case when $\text{sign}(A_{ij}) = \text{sign}(E_{ij})$. Enlarging $|E_{ij}|$ with this pattern, we must satisfy at least $|A'_{ij}| < |A_{ij}|$ to obtain the corresponding E'_{ij} that satisfies $|E'_{ij}| > |E_{ij}|$. The second pattern (b) represents the case when $\text{sign}(A_{ij}) \neq \text{sign}(E_{ij})$. Enlarging $|E_{ij}|$ with this pattern, we must satisfy at least $|A'_{ij}| > |A_{ij}|$ to obtain the corresponding E'_{ij} that satisfies $|E'_{ij}| > |E_{ij}|$.

For the first pattern (a), our strategy sets $|E_{ij}| = |D_{ij}|$, i.e., $A_{ij} = 0$. It is sufficient to satisfy the conditions to emphasize anomalies by $|E'_{ij}| > |E_{ij}|$. Although $|E'_{ij}|$ could be infinite, and thereby satisfy the conditions, this would be unrealistic and result in extreme outputs. As we assume each feature to have a zero mean and unit variance, $A'_{ij} = 0$ represents the mean value of the ith feature; thus, this is a realistic value

that also satisfies $|E'_{ij}| > |E_{ij}|$. Eventually, we modify (5) to obtain $A'_{ij} \approx 0$.

$$USV^T = W_A \odot (1 - L'_a), \tag{10}$$

$$L'_{a,ij} = \begin{cases} 1, \text{ if } L_{a,ij} = 1 \wedge \text{sign}(A^{(t)}_{ij}) = \text{sign}(E^{(t)}_{ij}) \\ 0, \text{ otherwise,} \end{cases} \tag{11}$$

where $\odot$ is the element-wise product. The (i, j) elements of W_A that satisfy $L'_{a,ij} = 1$ are set to zero before SVD is applied. Thus, (10) yields A', where $A'_{ij} \approx 0$ if $L'_{a,ij} = 1$. $E = D - A$ can emphasize unknown anomaly elements $|E_{ij}|$ that have similar anomalous patterns to those given by L_a.

The second pattern (b), where $\text{sign}(A_{ij}) \neq \text{sign}(E_{ij})$, cannot yield $|A'_{ij}| < |A_{ij}|$ by setting $A_{ij} = 0$. This pattern requires $|A'_{ij}| > |A_{ij}|$ to obtain $|E'_{ij}| > |E_{ij}|$. Although it is difficult to determine the most realistic value that satisfies $|A'_{ij}| > |A_{ij}|$, we ensure that A'_{ij} *at least* satisfies the constraint. Thus, after updating A using (10), if $|A'_{ij}| > |A_{ij}|$ is not satisfied, we set $A_{ij} = A'_{ij}$.

5.2.2. Updating E With the Feedback of Normal Label L_n

In this section, we follow the approach used in the matrix completion problem in Section 3.2 of (Lin et al., 2010) to update E with the normal label L_n. All that remains is to modify (7) as follows.

$$E^{(t+1)} = \mathcal{S}_{\lambda/\mu^{(t)}}[W_E \odot (1 - L_n)]. \tag{12}$$

This operation sets $E_{ij} = 0$ of (i, j) to satisfy $L_{n,ij} = 1$ such that $A_{ij} = D_{ij}$. By definition, the normal elements should be represented with the normal part A; thus, the constraint such that $A_{ij} = D_{ij}$ is consistent.

5.2.3. Risks of Overfitting

Our strategies adopt strong constraints because they directly modify the value of certain elements of A and E. Imposing these strong constraints is expected to significantly improve the accuracy of anomaly identification. However, this poses the risk of overfitting to the constraints and/or utilizing incorrect user feedback information.

In addition, labels are used to encourage the model to detect anomalies similar to those in the labeled elements. This carries the risk of confirmation bias since the updated model would not detect anomalies dissimilar to the labeled anomalies. Fortunately, this bias will be reduced as the number of feedback labels increases. The biased model tends to produce a greater number of false positives as the amount of feedback increases because the predictions would be inconsistent with the data, and are therefore likely to be wrong. Providing feedback in response to such false positives cancels the biases. However, the small amount of feedback still poses a risk to our framework.

The above risk is efficiently alleviated by the method introduced in Section 5.3.

5.3. Method to Incorporate Erroneous Feedback

In the previous section, we assumed that users can provide *complete* feedback that never includes incorrect information. However, in practice, users often provide *erroneous* feedback that is ambiguous and includes incorrect information. The risk of using erroneous feedback is its adverse effect on the representation model's learning process, i.e., the model would not be able to improve its performance by utilizing the feedback. In this section, we propose a novel approach known as the rank ensemble method, inspired by (Parikh et al., 2014), to enable our framework to utilize erroneous feedback. This proposed method, which is based on the consistency assumption, is introduced in the following section, and it adaptively ignores incorrect elements in the feedback.

5.3.1. Consistency Assumption

The rank ensemble method is based on the following assumption:

Consistency assumption: A good correspondence should exist between the data and the label feedback, but the information should be provided from different viewpoints.

This assumption is supported by the following two terms. First, consider a model learned from Algorithm 1 with the given data. If the obtained anomaly identification model is a global optimum for the data, we assume that the achieved nonzero part of $|E|$ identifies $|I_D|$ properly. However, a model trained by only considering the data would not easily attain the global optimum. Thus, we aim to leverage label information to improve optimality. Here, we say that the more closely the trained model approximates the global optimum, the greater the *consistency* of the model with the data.

Second, consider a model trained by Algorithm 2 with data and user feedback. We have feedback from a user indicating that a few (i, j)th elements are normal (or anomalous). As introduced in Section 5.2, we impose certain values of A_{ij} and E_{ij} according to the feedback. Therefore, the trained model should satisfy the label feedback to an acceptable extent. Here, we say that the greater the extent to which the model satisfies the label feedback, the greater the *consistency* of the model with the user feedback.

According to the consistency assumption, we assume that the above two notions of consistency simultaneously hold. Therefore, the greater the extent to which the model is consistent with the label feedback, the greater the consistency of the model with the data. This implies that, based on the consistency assumption, the data and the label feedback should

correspond, but the information should be acquired from different perspectives. The next section introduces our novel rank ensemble method based on the consistency assumption.

5.3.2. Rank Ensemble

As noted in Section 5.2.3, several risks violate the consistency assumption. Thus, we aim to adaptively ignore user feedback that does not follow the consistency assumption by considering it to be incorrect information. We achieve this goal by estimating how well the data and label feedback follow the consistency assumption by evaluating the structure of A and E. Label feedback that is inconsistent with the data is adaptively ignored using our novel rank ensemble approach inspired by (Parikh et al., 2014).

Let A, E be $\hat{A}^{(t)}, \hat{E}^{(t)}$ obtained from the tth step of Algorithm 2, where $\text{rank}(A) = r \ll \min(K, N)$. From SVD, $A = USV^T$, where U, S, V are $K \times r$, $r \times r$, and $K \times r$ matrices, respectively. S is a diagonal singular value matrix and is set to have mth singular value σ_m to an element (m, m) in descending order, i.e., $\sigma_1 \geq \ldots \geq \sigma_m \geq \ldots \geq \sigma_r$. Let $A_l = \sum_{m=1}^{l} u_m \sigma_m v_m^T$ be a low-rank representation using up to the lth singular vectors, where u_m and v_m are the corresponding left and right singular vectors to σ_m, $E_l = D - A_l$, $A = A_r$, $E = E_r$, $\text{rank}(A_l) = l$, and $l = 1, \ldots, r$. E_l is a residual and represents the part of D that cannot be expressed by A_l.

Suppose we have feedback from a user, indicating that several (i, j)th elements are normal. As introduced in Section 5, we impose $A_{r,ij} = A_{ij} = D_{ij}, E_{r,ij} = E_{ij} = 0$ for (i, j) satisfying $L_{n,ij} = 1$. In this situation, if $l \ll r$, A_l has less representation power than A_r and only the principal components shared across the data and feedback information can be expressed. This type of representation is known as a low-granularity representation (Parikh et al., 2014). Conversely, if $l \approx r$, A_l achieves a high-granularity representation and can even follow incorrect and inconsistent feedback given by L_n. In particular, if $l = r$, $A = A_l$, and $A_{l,ij} = A_{ij} = D_{ij}, E_{l,ij} = E_{ij} = 0$ for (i, j) satisfying $L_{n,ij} = 1$. Note that the same can be said for anomaly feedback L_a.

According to the consistency assumption, if the data and the label feedback are consistent, we can assume that the low-granularity representation also has an improved ability to satisfy the imposed constraints. However, if inconsistent labels are provided, these constraints are only satisfied by the high-granularity representation. Thus, we conjecture that only inconsistent labels would drastically change the representation between low- and high-granularity. To alleviate this effect, we use an ensemble of $|E_l|$ as follows to estimate the anomalous elements.

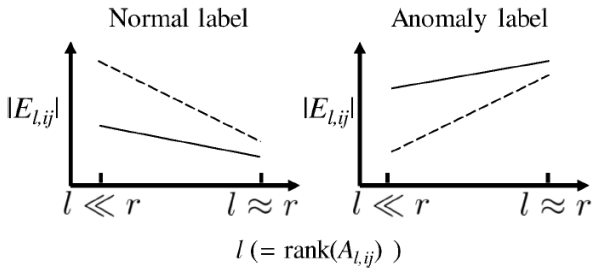

Figure 3. Relations between the size of $|E_{ij}|$ and the rank of A_l. The solid and broken lines denote that the label given to element (i, j) is correct and incorrect, respectively. Incorrect labels are not consistent with the corresponding data and the difference in $|E_{l,ij}|$ between $l \ll r$ and $l \approx r$ would be large.

$$\bar{E} = \frac{1}{r}\sum_{l=1}^{r} |E_l| = \frac{1}{l}\sum_{l=1}^{r} |D - A_l|. \tag{13}$$

We refer to this as the rank ensemble. If the user feedback for element (i, j) is incorrect, that is, the user either labels anomalies as normal or labels normal elements as anomalies, the values of $A_{l,ij}, E_{l,ij}$ change more drastically than if the user feedback regarding element (i, j) is correct (Figure 3). Because of this effect, $\bar{E}_{ij}$ becomes large if an incorrect normal label is given to element (i, j) (or $\bar{E}_{ij}$ becomes small if an incorrect anomaly label is given to element (i, j)). This indicates that $\bar{E}$ can adaptively ignore incorrect labels. However, note that $\bar{E}$ also alleviates the effect of correct labels; thus, its performance is poorer than when using $|E_r|$ if all the user feedback is correct. Finally, we use $\bar{E}$ for anomaly identification in line 3 of Algorithm 2.

6. EXPERIMENT WITH BENCHMARK DATASETS

The numerical experiments described in this section evaluated the ability of our framework to utilize erroneous user feedback to improve the accuracy of anomaly identification. The experiments were conducted with various parameter settings and with or without the rank ensemble method. Overall, the experiments indicated that at various settings, our framework with the rank ensemble method works well with erroneous feedback. In particular, increasing the amount of feedback is important to promote a significant improvement in anomaly identification, and this is enabled by accepting erroneous feedback. Overfitting due to an insufficient amount of feedback resulted in a biased model and failed to improve the performance, whereas increasing the amount of feedback or using the rank ensemble method successfully overcame this problem.

For the computational environment, we used Windows 7 (64-bit) with an Intel(R) Core(TM) i7-3970X CPU @ 3.50 GHz and 64 GB memory. All implementations were performed

Table 1. Benchmark datasets[1].

Dataset	K	N_n	N_a	KN_n	C_n	C_a
Pageblocks	10	4941	532	49410	2	3
Arrhythmia	274	257	176	70418	4	9
Letter	16	9940	10060	159040	13	13
Optdigits	62	2822	2798	174946	5	5
Mfeat	649	1000	1000	649000	5	5

[1] K: # of features, N_n: # of normal samples, N_a: # of anomaly samples, C_n: # of normal classes, C_a: # of anomaly classes.

using MATLAB R2012b.

6.1. Datasets

Table 1 summarizes the datasets used in the experiments. The datasets were obtained from the UCI machine-learning repository (Dua & Graff, 2017), a collection of datasets used by the machine learning community. We adopted multiclass datasets: Pageblocks(Malerba, 1995), Arrhythmia(Guvenir, Acar, & Muderrisoglu, 1998), Letter(Slate, 1991), Optdigits(Alpaydin & Kaynak, 1998), and Mfeat(Duin, n.d.). They enabled us to select some of the classes as normal data and the remaining classes as anomalous data. Note that the anomalous data selected from the multiclass datasets represent anomalies of a diverse nature. More information is available through the citations of the datasets. The datasets were standardized such that each feature had a zero mean and a unit variance. For computational reasons, some datasets containing more than 40,000 samples were randomly subsampled to 40,000 samples in total by using uniform distribution random sampling. Samples were subsampled per class such that the resulting subsampled 40,000 samples preserve the original class distribution over samples.

6.2. Experimental Parameters

We evaluated our novel framework shown in Algorithm 2 using each of the datasets introduced in the previous section. The parameters were varied with respect to the number and variety of anomalies, the amount of feedback, the rate of erroneous feedback, and whether the experiment was conducted with or without the rank ensemble method. The following sections explain how we set the number and variety of anomalies, amount of feedback, and rate of erroneous feedback.

6.2.1. Anomalies $|I_D|$

The given datasets require the corresponding anomaly indices I_D to simulate user feedback and to evaluate the accuracy of anomaly identification. We determined I_D by subsampling the anomaly datasets as follows, according to (Emmott, Das, Dietterich, Fern, & Wong, 2013) and (Siddiqui, Fern, Dietterich, & Wong, 2019). Recall that D is a $K \times N$ data matrix, and let $\nu = |I_D|/(KN)$ be the ratio of anomalies included in the data represented by D. As $|I_D| = (\sqrt{\nu}N)(\sqrt{\nu}K)$, for simplicity, we subsampled $\sqrt{\nu}N$ samples from the anomaly dataset and then designated the $\sqrt{\nu}K$ features of each selected sample as anomaly elements to create I_D. Since $|I_D|$ depends on ν and $|D|$ (especially N_n), the number of N_n should be fixed; thus, this enables us to determine only the effect of varying $|I_D|$ with fixed $|D|$ and vice versa. Although a change in ν also affects the size $|D|$, we approximately fix $|D|$ by replacing the number of normal elements KN_n with the average among the normal datasets shown in Table 1, which is $Z = 266748$. Thus, $|I_D| = \nu KN = \nu Z/(1-\sqrt{\nu})$ is independent of the size of each normal dataset and vice versa. In the case where $N = N_n + \sqrt{\nu}N$, $\sqrt{\nu}N = \sqrt{\nu}Z/(K - K\sqrt{\nu})$ samples are subsampled for an anomalous dataset. To subsample from the anomaly dataset, we followed (Emmott et al., 2013) and used a kernel logistic regression (KLR) model (Keerthi, Duan, Shevade, & Poo, 2005) to rank and subsample the top anomalous samples from each class, on the condition that the class distribution was preserved. Note that KLR cannot subsample features, but only samples in datasets. Therefore, it cannot be used as an anomaly identification method and is unrelated to our framework itself.

Table 2. Simulated analyst models to determine I_D.

Model	Parameters
Principal component analysis (PCA)	d_p
Random forest (RF)	c_f

Table 3. Values of M and $|I_D|$ according to ν and β, where the average number of normal elements is $Z = 266748$.

β or ν	0.005	0.010	0.050
$M = \beta Z$ or $\lvert I_D\rvert = \nu Z$	1334	2668	13337

As the datasets do not provide anomaly labels at the feature level, we must artificially determine the anomalous features. Given the subsampled anomaly dataset, according to a simulated analyst model introduced in (Siddiqui et al., 2019), a regularized random forest method was previously used to rank and determine the actual anomalous features for each anomaly sample. In contrast, we ensured the variability by using several methods listed in Table 2 as simulated analyst models to determine the $\sqrt{\nu}K$ features as anomalous elements. Each method can rank features with respect to the anomaly rate and determine the top anomalous features from each sample and $\sqrt{\nu}K$ in total.

The parameters of the models were determined as follows. Principal component analysis (PCA) is a linear dimensionality reduction method, and the number of principal components d_p is determined by $\max d$, subject to $\sum_{i=1}^{d} p_i / \sum_{i=1}^{K} p_i \leq 0.7$, where p_i is the ith principal component. PCA maps the dataset D to a learned low-dimensional representation to reconstruct A, and obtain $E = D - A$, the size of which ($|E|$) can be used as the anomaly rate. Random forest (RF) is a regression-tree-based ensemble method in which we set the number of trees in the ensemble as $c_f = 50$. The difference in

the regression errors between the experiment with and without a feature is used as the anomaly rate of the feature for RF, according to (Siddiqui et al., 2019).

6.2.2. Amount of Feedback

Let $\beta = M/|D|$. We varied β to determine the amount of feedback, M. Note that $|D|$ is maintained as a constant by Z and fixing ν similarly as introduced in the previous section; thus, the variation in β only shows the effect of M.

6.2.3. Erroneous Feedback

According to Algorithm 2, with every iteration, we have M new labels denoting whether certain elements are normal or anomalous, as indicated by L_n, L_a. Erroneous feedback leads to the assignment of incorrect labels with L_n, L_a; for anomalous elements, $L_{n,ij} = 1$, and for normal elements, $L_{a,ij} = 0$. We introduce an error rate $\rho \in [0, 1]$ to capture the extent of mislabeling. Let $M_n^{(t)}$ be the amount of feedback for normal labels, and let $M_a^{(t)}$ be the amount of feedback for anomalous labels for the tth iteration, where $M = M_n^{(t)} + M_a^{(t)}$. In every iteration, $\rho M_n^{(t)}$ of the normal label feedback and $\rho M_a^{(t)}$ of the anomalous label feedback are randomly assigned to incorrect labels. Note that if $\rho = 0$, we assume complete feedback that never includes incorrect information. The robustness of the rank ensemble method is then evaluated by varying ρ.

6.3. Experimental Setting and Evaluation

Recall that $\beta = M/|D|$, $\nu = |I_D|/|D|$, and $|D|$ is a constant maintained by the average number of normal elements KN_n among the datasets, that is, $Z = 266748$, to determine the actual values of M and I_D. For each dataset with I_D determined by each simulated analyst, we tested cases, in which $\beta = \{0.005, 0.05\}$ with fixed $\nu = 0.01$, and $\nu = \{0.005, 0.05\}$ with $\beta = 0.01$ fixed, to determine the effects of varying β and ν. For each combination of β and ν, we tested the error rate for $\rho = \{0, 0.2, 0.4\}$. Table 3 shows the corresponding size of M and $|I_D|$. Using these settings, our goal is to train a model to be capable of finding a good solution to Problem 2; thus, the area under the curve (AUC) of the receiver operating characteristic (ROC) based on $\hat{E}^{(t)}$ is used as an evaluation criterion: AUC shows the extent to which $\hat{E}^{(t)}$ of the tth iteration successfully indicates appropriate anomaly elements I_D.

To reduce the number of combinations, we only test the values.

We also compared our framework with Algorithm 1 without utilizing user feedback to estimate E. In every iteration of Algorithm 2, we fixed E and selected every other M candidate in descending order of the value of $|E_{ij}|$. Note that the accuracy of E is the baseline and we aim at least to obtain results that improve on the baseline.

The processing time of our framework is also important to allow it to be used interactively because computational inefficiency would require the user to wait longer and decrease its usability. We therefore evaluated the average computational time to output the prediction of each step, i.e., the time required to process line 5 of Algorithm 2 is evaluated for each dataset.

6.4. Results

Table 4 provides the obtained results, with a focus on selected datasets and the experimental conditions described in Section 6.3. Owing to the page limitation, each value in the tables shows the achieved AUC of only the fifth iterative cycle. If the amount of feedback reaches 50% of the data or AUC reaches 1, we stop the iteration and show the AUC of that point.

Table 5 lists the average computational time required to output the prediction of each step, i.e., the time to process line 5 of Algorithm 2.

6.4.1. Without the Rank Ensemble Method

In this section, we discuss the results obtained without using the rank ensemble method, as shown in the columns under "Without ensemble" in Table 4 as well as the processing times shown in Table 5.

Our framework without the ensemble largely succeeded in improving the AUCs with $\rho = 0$ compared to having no feedback. Few experiments, such as the Pageblocks dataset with the RF simulated analyst, yielded slightly worse results. This can be explained by overfitting to the user feedback, which causes a model to diverge from the data distribution and produce more false-positive estimates. The improvements here suggest that even for users who do not know about the model itself, user feedback alone is sufficient to improve the model's performance. Hence, the framework presents a cost-effective approach that achieves improvements on a wide variety of datasets and anomalies.

When ρ increases, the performance of our framework without the ensemble decreases drastically. This is understandable because our method adopts update strategies that strictly follow the constraints imposed by the user feedback. Therefore, without the rank ensemble, our framework is vulnerable to incorrect feedback. For large values of ρ, in most of the experiments, the AUCs were worse than those without feedback. This vulnerability to incorrect feedback is alleviated by using the rank ensemble method.

Even if ρ is small, the results such as $\beta = 0.005, \nu = 0.01$ of the Pageblocks dataset with the RF simulated analyst, failed to improve the AUC. This is considered to be caused by the confirmation bias influencing the feedback, as discussed in Section 5.2.3. Our framework aims to strongly follow the

Table 4. AUC at the 5th iteration.

Simulated Analyst	Dataset	β: ratio of M, ν: ratio of anomalies	Without ensemble			With ensemble			No feedback
			$\rho = 0$	$\rho = 0.2$	$\rho = 0.4$	$\rho = 0$	$\rho = 0.2$	$\rho = 0.4$	
PCA	Pageblocks	$\beta = 0.005, \nu = 0.01$	0.9994	0.9512	0.8950	0.9961	0.9964	0.9921	0.9921
		$\beta = 0.05, \nu = 0.01$	0.9999	0.9295	0.8678	0.9993	0.9977	0.9928	0.9921
		$\beta = 0.01, \nu = 0.005$	1.0000	0.9490	0.8906	0.9963	0.9958	0.9935	0.9921
		$\beta = 0.01, \nu = 0.05$	0.9956	0.9363	0.8749	0.9926	0.9878	0.9849	0.9661
	Arrhythmia	$\beta = 0.005, \nu = 0.01$	0.8399	0.798	0.7648	0.8837	0.8777	0.8730	0.8181
		$\beta = 0.05, \nu = 0.01$	0.9313	0.8701	0.8049	0.9116	0.9025	0.8879	0.8181
		$\beta = 0.01, \nu = 0.005$	0.8987	0.8443	0.8012	0.9308	0.9266	0.9118	0.8584
		$\beta = 0.01, \nu = 0.05$	0.7791	0.7362	0.6980	0.7950	0.7904	0.7896	0.7232
	Letter	$\beta = 0.005, \nu = 0.01$	0.9306	0.9048	0.8785	0.9374	0.9315	0.9252	0.9110
		$\beta = 0.05, \nu = 0.01$	0.9883	0.9213	0.8561	0.9824	0.9683	0.9509	0.9110
		$\beta = 0.01, \nu = 0.005$	0.9775	0.9256	0.8738	0.9798	0.9710	0.9601	0.9472
		$\beta = 0.01, \nu = 0.05$	0.8341	0.8141	0.7953	0.8437	0.8369	0.8299	0.8201
	Optdigits	$\beta = 0.005, \nu = 0.01$	0.9455	0.9061	0.8702	0.9602	0.9540	0.9503	0.9187
		$\beta = 0.05, \nu = 0.01$	0.9835	0.9209	0.8575	0.9841	0.9743	0.9628	0.9187
		$\beta = 0.01, \nu = 0.005$	0.9754	0.9168	0.8610	0.9847	0.9810	0.9726	0.9393
		$\beta = 0.01, \nu = 0.05$	0.8620	0.8386	0.8124	0.8781	0.8755	0.8692	0.8354
	Mfeat	$\beta = 0.005, \nu = 0.01$	0.9122	0.8802	0.8570	0.9374	0.9289	0.9223	0.9049
		$\beta = 0.05, \nu = 0.01$	0.9422	0.8821	0.8208	0.9694	0.9572	0.9432	0.9049
		$\beta = 0.01, \nu = 0.005$	0.9461	0.8995	0.8564	0.9651	0.9555	0.9449	0.9367
		$\beta = 0.01, \nu = 0.05$	0.8076	0.7809	0.7570	0.8451	0.8372	0.8289	0.7834
RF	Pageblocks	$\beta = 0.005, \nu = 0.01$	0.7563	0.7340	0.7032	0.8789	0.8833	0.8716	0.7696
		$\beta = 0.05, \nu = 0.01$	0.9126	0.8524	0.8157	0.9196	0.9114	0.8915	0.7494
		$\beta = 0.01, \nu = 0.005$	0.8436	0.7894	0.7656	0.8632	0.8427	0.8586	0.7793
		$\beta = 0.01, \nu = 0.05$	0.8270	0.7770	0.7193	0.8768	0.8478	0.8875	0.7595
	Arrhythmia	$\beta = 0.005, \nu = 0.01$	0.5286	0.5251	0.5113	0.5357	0.5374	0.5329	0.5038
		$\beta = 0.05, \nu = 0.01$	0.6974	0.6542	0.6294	0.6017	0.5843	0.5701	0.5123
		$\beta = 0.01, \nu = 0.005$	0.5723	0.5610	0.5408	0.5677	0.5579	0.5450	0.5138
		$\beta = 0.01, \nu = 0.05$	0.5800	0.5619	0.5443	0.5536	0.5433	0.5416	0.5055
	Letter	$\beta = 0.005, \nu = 0.01$	0.6103	0.6003	0.5956	0.6182	0.6138	0.6064	0.5640
		$\beta = 0.05, \nu = 0.01$	0.7330	0.6973	0.6517	0.6724	0.6622	0.6498	0.5704
		$\beta = 0.01, \nu = 0.005$	0.6394	0.6222	0.6085	0.6525	0.6449	0.6356	0.5885
		$\beta = 0.01, \nu = 0.05$	0.5554	0.5467	0.5378	0.5069	0.5044	0.5019	0.4706
	Optdigits	$\beta = 0.005, \nu = 0.01$	0.5896	0.5796	0.5754	0.6075	0.6036	0.6034	0.5709
		$\beta = 0.05, \nu = 0.01$	0.7008	0.6633	0.6278	0.6404	0.6288	0.6245	0.5633
		$\beta = 0.01, \nu = 0.005$	0.6166	0.6042	0.5898	0.6395	0.6364	0.6298	0.5864
		$\beta = 0.01, \nu = 0.05$	0.5667	0.5584	0.5494	0.5565	0.5543	0.5515	0.5347
	Mfeat	$\beta = 0.005, \nu = 0.01$	0.5311	0.5252	0.5178	0.5724	0.5726	0.5773	0.5454
		$\beta = 0.05, \nu = 0.01$	0.5773	0.5680	0.5464	0.5965	0.5981	0.5904	0.5468
		$\beta = 0.01, \nu = 0.005$	0.5368	0.5291	0.5344	0.5823	0.5881	0.5850	0.5531
		$\beta = 0.01, \nu = 0.05$	0.5310	0.5266	0.5248	0.5518	0.5514	0.5518	0.5230

feedback by forcing the corresponding value to be $E_{ij} = 0$ or $A_{ij} = 0$. Thus, the obtained model may overfit the imposed feedback and tend to diverge from the global optimum. As such, the model detects anomaly candidates similar to the labeled anomaly elements. The next section shows that the use of the rank ensemble method successfully avoids such biases.

A comparison of the values obtained for β, which represents the value of M, shows that a larger value of M results in a greater improvement in the AUC. However, to increase M, there is a tradeoff in terms of the feedback cost: achieving a high value for M requires the value of ρ to be large and vice versa. The next section shows that the rank ensemble method can alleviate the effect of ρ; thus, a large M value with a reasonable ρ value is probably the preferable choice in practice.

The differences between the values of ν, which represents the number of anomalies, show that the larger ν is, the worse the initial AUC (iteration #1) tends to be. This is because a large number of anomalies could include various anomalies that are difficult to detect. In addition, the results with a smaller ν value are more vulnerable to incorrect feedback. This is because, with a small number of anomalies, a large proportion of anomalies can be easily missed by even a slightly inappropriate modification of the anomaly identification model.

According to the processing times shown in Table 5, despite a longer time being taken with a large dataset such as Mfeat, the processing times were essentially short enough for users not having to wait long for the next prediction. Note that the computational time depends not only on the data size but also on the convergence speed, which varies with the experimental conditions.

Table 5. Average computational time [s].

Dataset	Average computational time [s]
Pageblocks	0.2370
Arrhythmia	5.5643
Letter	0.9698
Optdigits	1.4230
Mfeat	30.61

6.4.2. With the Rank Ensemble Method

Comparing the results obtained by our framework for $\rho > 0$ with and without incorporating the rank ensemble method, the rank ensemble method clearly improved the performance when using incorrect labels. Therefore, the use of our framework in combination with the rank ensemble method is highly successful because it is capable of achieving accurate anomaly identification with erroneous feedback. In practice, situations such as these commonly occur.

When $\rho = 0$, results such as $\beta = 0.05, \nu = 0.01$ of the Arrhythmia dataset with the RF simulated analyst perform worse than the framework without the rank ensemble method. This is because the rank ensemble method also smooths the constraints provided by even the correct labels. These results are acceptable because requiring feedback with a low error rate would be difficult, costly, and thus unrealistic.

Interestingly, for certain other datasets such as $\beta = 0.005, \nu = 0.01$ of the Mfeat dataset with the PCA simulated analyst our framework achieved improved AUCs with the rank ensemble method than without the rank ensemble method, even if $\rho = 0$. As explained in Sections 5.2.1 and 5.2.2, our update strategies strongly follow the given label constraints. This may cause extreme results including confirmation bias; hence, these strategies do not achieve the greatest improvement in the AUC, even if the given labels are correct. The rank ensemble method smooths the constraints such that the negative impact is countered and the performance is improved.

Comparing datasets of different sizes, β and ν, the results have tendencies similar to those of our framework without the rank ensemble method, as explained in Section 6.4.1. Thus, the rank ensemble method has little effect on these parameters, except for ρ.

7. EXPERIMENT WITH VEHICLE DRIVING DATA

We conducted a case study using real vehicle driving data with a specific feature set and a scenario to demonstrate how our framework supports operators in analyzing unknown anomalies. The problem considered here involves an analysis of the changes between two driving environments. The experiment demonstrates that our framework's predictions can be improved by user feedback and can assist users in identifying the true causes of anomalies.

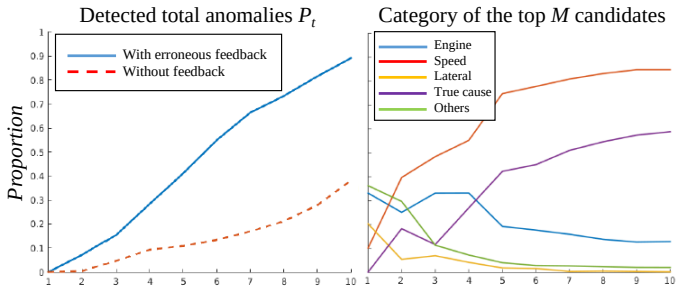

Figure 4. Left: the proportion of anomalies detected P_t. Right: categorical proportion of the top M anomalous candidates.

Table 6. Categories of attributes. The speed control category includes three true causes.

Categories	# of attributes
Engine control	17
Speed control	7
Lateral control	6
Others	7
Total	37

7.1. Datasets and Anomalies I_D

The fault diagnosis equipment recorded data from the vehicle engine and control systems containing 37 attributes with sampling rates of 0.5-–1.0 s. Table 6 presents a summary of these attributes by category.The data were recorded under two conditions. First, under flat-road conditions, the vehicle was driven on the same flat road on several occasions with constant acceleration and deceleration to obtain 1450 samples as the normal dataset. Second, we used the same flat-road conditions with slower acceleration and deceleration to obtain 737 samples as the changed dataset. For preprocessing, the sampling rate was adjusted to a constant of 0.5 s using linear interpolation. We combined the normal and changed datasets into a single dataset. The combined dataset was normalized for each attribute of the normal data to have a mean value of 0 with unit variance.

The driver inputs, specifically Accelerator Position (representing acceleration) and Stroke Sensor 1 & 2 (representing deceleration), were changed between the two datasets. Thus, the causes of the change (anomaly features) are Stroke Sensor 1 & 2 (representing deceleration) and Accelerator Position, which belong to the speed control category. In addition, the changed behavior was observed only when the driver accelerated or decelerated, and thus the Accelerator Position or Stroke Sensor 1 & 2 elements with nonzero values in the changed dataset were designated as anomaly I_D. In addition to the true causes, the throttle position and engine speed sensors, which belong to the engine control category, were also incorrectly labeled as anomalies in all the samples.

7.2. Erroneous Feedback

We follow the setup in Section 6.2.3 except for randomly inserting incorrect labels. Instead, we introduce an approximately realistic rough labeling scheme. This scheme assumes that the user has only approximate information about a situation and executes as follows: 1. The test driver feels a change affecting the speed and provides approximate feedback to the model predictions. 2. The driver selects some attributes related to speed and simply labels them all as anomalous. The rough labeling scheme requires neither precise information nor labeling and, therefore, has practical utility.

7.3. Experimental Setting and Evaluation

We used a different metric from Section 6 to focus on and improve the interpretability of this specific case study. The user's prime objective is to identify anomalies with minimum effort. Thus, instead of the AUC, which is an abstract measure based on both false and true positives, that was used in Section 6, we simply evaluated the proportion of anomaly elements correctly estimated for all true anomalies, denoted as P_t. If P_t is large with less user feedback, that is, a smaller number of repeated steps, this indicates success in identifying anomalous elements effectively with our interactive framework. We tested how well P_t is improved by user feedback. In this experiment, we set $\nu = 0.01$ to determine the value of M.

We also evaluate how the prediction of our framework changes according to user feedback. Based on the categories introduced in Table 6, we monitored the top M anomalous elements based on E at each step to determine how a category proportion changes.

7.4. Results

The plot on the left side of Figure 4 shows P_t over the feedback iterations. The figure indicates that our framework increases the proportion of anomalies detected to a greater extent with user feedback. This implies that our framework is efficient and useful for users.

The plot on the right side of Figure 4 shows the proportion of the top M anomalous elements by category at each step to determine the extent to which the model changes a prediction according to user feedback. In the initial step, the model detected many elements belonging to the engine control category, which is *indirectly* related to the true cause because changing the acceleration or deceleration patterns affects engine behavior. However, the initial model failed to detect the true causes. Conversely, in the latter step, when the model was used in conjunction with erroneous feedback, the model predicted elements belonging to the speed control category, which, largely, included the true causes. A large portion of the estimates of the top M candidates in the latter step denotes the true causes. This information provides useful insight for users to surmise that attributes in the speed category, rather than the engine category, are suspicious. Thus, our framework successfully incorporated erroneous feedback to detect the true causes correctly, whereas it avoided the detection of elements belonging to the engine category based on the feedback information.

8. CONCLUSION

This paper proposes a novel anomaly identification framework that can utilize user knowledge and feedback to improve performance interactively. Our framework is based on a sparse and low-rank model capable of identifying anomalies as well as the features responsible for causing the anomalies. We overcame the limitations associated with the accuracy of anomaly identification by utilizing user knowledge to improve accuracy. Instead of modifying the model structure manually, our framework obtains user knowledge as feedback in response to the estimation provided by the model, according to which the model is then modified automatically. The process continues interactively and is thus in agreement with the exploratory nature of procedures used to identify unknown anomalies. The experimental results demonstrated that our framework achieved consistent improvements in anomaly identification accuracy on several datasets. In addition, we propose a method to improve the accuracy of anomaly identification with *erroneous* feedback, that is, feedback that includes incorrect information. Based on the consistency assumption, we constructed a rank ensemble method that adaptively ignores incorrect information. The experiments performed using erroneous feedback confirmed that the use of our framework combined with the rank ensemble method continued to improve the accuracy, even when the user feedback included incorrect information.

9. FUTURE WORK

Our future work will include proposing the optimal conditions under which to update A, E using label feedback to achieve the largest improvement in anomaly identification accuracy. As only a sufficient condition was used to improve the accuracy in this work, that is, by increasing $|E_{ij}|$, there is scope to obtain greater improvement using the optimum condition. To derive the condition, the perturbation theory of SVD(Stewart, 1991) is expected to be useful for analyzing how the low-rank representation (SVD) is affected by the outliers. An extension to a nonlinear model, such as a low-rank subspace model (Liu et al., 2013), is also important for improving the accuracy.

REFERENCES

Alpaydin, E., & Kaynak, C. (1998). *Optical recognition of handwritten digits data set.* Retrieved

from https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000, May). Lof: Identifying density-based local outliers. *SIGMOD Rec.*, *29*(2), 93–104.

Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011, June). Robust principal component analysis? *J. ACM*, *58*(3), 11:1–11:37.

Chandola, V., Banerjee, A., & Kumar, V. (2009). /em anomaly detection: A survey. *ACM Comput. Surv.*, *41*(3), 15:1–15:58.

Dua, D., & Graff, C. (2017). *UCI machine learning repository.* Retrieved from http://archive.ics.uci.edu/ml

Duin, R. P. (n.d.). *Multiple features data set.* Retrieved from https://archive.ics.uci.edu/ml/datasets/Multiple+Features

Elahi, M., Ricci, F., & Rubens, N. (2014). Active learning in collaborative filtering recommender systems. In *E-commerce and web technologies* (pp. 113–124). Cham: Springer International Publishing.

Emmott, A. F., Das, S., Dietterich, T., Fern, A., & Wong, W.-K. (2013). Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the acm sigkdd workshop on outlier detection and description* (pp. 16–21). New York, NY, USA: ACM.

Guvenir, H. A., Acar, B., & Muderrisoglu, H. (1998). *Arrhythmia data set.* Retrieved from https://archive.ics.uci.edu/ml/datasets/arrhythmia

Hero, A. O. (2007). Geometric entropy minimization (gem) for anomaly detection and localization. In B. Schölkopf, J. C. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems 19* (pp. 585–592). MIT Press.

Hsieh, C., Natarajan, N., & Dhillon, I. S. (2014). PU learning for matrix completion. *CoRR*, *abs/1411.6081*. Retrieved from http://arxiv.org/abs/1411.6081

Keerthi, S. S., Duan, K. B., Shevade, S. K., & Poo, A. N. (2005, November). A fast dual algorithm for kernel logistic regression. *Mach. Learn.*, *61*(1-3), 151–165.

Kriegel, H.-P., Kröger, P., Schubert, E., & Zimek, A. (2009). Loop: Local outlier probabilities. In *Proceedings of the 18th acm conference on information and knowledge management* (pp. 1649–1652). New York, NY, USA: ACM.

Lin, Z., Chen, M., & Ma, Y. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *CoRR*, *abs/1009.5055*.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., & Ma, Y. (2013). Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, *35*(1), 171-184.

Malerba, D. (1995). *Page blocks classification dataset.* Retrieved from https://archive.ics.uci.edu/ml/datasets/Page+Blocks+Classification

Mardani, M., Mateos, G., & Giannakis, G. B. (2013a). Dynamic anomalography: Tracking network anomalies via sparsity and low rank. *J. Sel. Topics Signal Processing*, *7*(1), 50–66.

Mardani, M., Mateos, G., & Giannakis, G. B. (2013b, August). Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies. *IEEE Trans. Inf. Theor.*, *59*(8), 5186–5205.

Papadimitriou, S., Kitagawa, H., B. Gibbons, P., & Faloutsos, C. (2003, 01). Loci: Fast outlier detection using the local correlation integral. In (p. 315-326).

Parikh, A. P., Saluja, A., Dyer, C., & Xing, E. (2014, October). Language modeling with power low rank ensembles. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1487–1498). Doha, Qatar: Association for Computational Linguistics.

Pham, D.-S., Venkatesh, S., Lazarescu, M., & Budhaditya, S. (2014, January). Anomaly detection in large-scale data stream networks. *Data Min. Knowl. Discov.*, *28*(1), 145–189.

Raghavan, H., Madani, O., & Jones, R. (2006, December). Active learning with feedback on features and instances. *J. Mach. Learn. Res.*, *7*, 1655–1686.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., & Williamson, R. C. (2001, July). Estimating the support of a high-dimensional distribution. *Neural Comput.*, *13*(7), 1443–1471.

Scott, C., & Nowak, R. (2006). Learning minimum volume sets. In Y. Weiss, B. Schölkopf, & J. C. Platt (Eds.), *Advances in neural information processing systems 18* (pp. 1209–1216). MIT Press.

Settles, B. (2011). Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 1467–1478). Association for Computational Linguistics.

Siddiqui, M. A., Fern, A., Dietterich, T. G., & Wong, W.-K. (2019, January). Sequential feature explanations for anomaly detection. *ACM Trans. Knowl. Discov. Data*, *13*(1), 1:1–1:22.

Sindhwani, V., Bucak, S. S., Hu, J., & Mojsilovic, A. (2010). One-class matrix completion with low-density factorizations. In *ICDM* (pp. 1055–1060). IEEE Computer Society.

Slate, D. J. (1991). *Letter recognition data set.* Retrieved from https://archive.ics.uci.edu/ml/datasets/Letter+Recognition

Sricharan, K., & Hero, A. O. (2011). Efficient anomaly detection using bipartite k-nn graphs. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 24* (pp. 478–486). Curran Associates, Inc.

Stewart, G. W. (1991). Perturbation theory for the singular value decomposition. *SVD and Signal Processing, II: Algorithms, Analysis and Applications*, 99–109.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Adv. Artificial Intellegence*, *2009*, 421425:1–421425:19.

Subba, B., Biswas, S., & Karmakar, S. (2016, 03). A neural network based system for intrusion detection and attack classification. In (p. 1-6).

Tagawa, T., Tadokoro, Y., & Yairi, T. (2015, 26–28 Nov). Structured denoising autoencoder for fault detection and analysis. In D. Phung & H. Li (Eds.), *Proceedings of the sixth asian conference on machine learning* (Vol. 39, pp. 96–111). Nha Trang City, Vietnam: PMLR.

Tax, D. M. J., & Duin, R. P. W. (2004, January). Support vector data description. *Mach. Learn.*, *54*(1), 45–66.

Tipping, M. E., & Bishop, C. M. (1999). Mixtures of probabilistic principal component analyzers. *Neural Computation*, *11*(2), 443-482.

Yairi, T., Takeishi, N., Oda, T., Nakajima, Y., Naoki, N., & Takata, N. (2017, June). A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction. *IEEE Transactions on Aerospace and Electronic Systems*, *53*(3), 1384-1401.

Zhao, M., & Saligrama, V. (2009). Anomaly detection with score functions based on nearest neighbor graphs. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems 22* (pp. 2250–2258). Curran Associates, Inc.

BIOGRAPHIES

Takaaki Tagawa received the B.E. in mechanical engineering from Waseda University, Tokyo, Japan, in 2010, and the M.E. in aeronautics and astronautics engineering from the University of Tokyo, Tokyo, Japan, in 2012. Since 2012, he has been with Toyota Central R&D Labs., Inc. His current research interests include data mining and machine learning for anomaly detection and their applications.

Yukihiro Tadokoro received the B.E., M.E., and Ph.D. degrees in information electronics engineering from Nagoya University, Aichi, Japan, in 2000, 2002, and 2005, respectively. Since 2006, he has been with Toyota Central R&D Labs., Inc. In 2011 and 2012, he worked as a research scholar at the Department of Physics and Astronomy, Michigan State University, USA, to study nonlinear phenomena for future applications in the signal and information processing fields. His current research interests include data mining and machine learning, in addition to noise-related phenomena in nonlinear systems and their applications. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE), Japan, and the IEEE.

Takehisa Yairi received the M.Sc. and Ph.D. degrees in aerospace engineering from the University of Tokyo, Japan in 1996 and 1999, respectively. He is currently a Professor with the Research Center for Advanced Science and Technology, the University of Tokyo. His research interests include machine-learning theory and its application.