This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

Discovering Anomalies on Mixed-Type Data using a Generalized Student-t Based Approach

Yen-Cheng Lu, Feng Chen, Yating Wang and Chang-Tien Lu

Abstract—

Anomaly detection in mixed-type data is an important problem that has not been well addressed in the machine learning field. Existing approaches focus on computational efficiency and their correlation modeling between mixed-type attributes is heuristically driven, lacking a statistical foundation. In this paper, we propose MIxed-Type Robust dEtection (MITRE), a robust error buffering approach for anomaly detection in mixed-type datasets. Because of its non-Gaussian design, the problem is analytically intractable. Two novel Bayesian inference approaches are utilized to solve the intractable inferences: Integrated-nested Laplace Approximation (INLA), and Expectation Propagation (EP) with Variational Expectation-Maximization (EM). A set of algorithmic optimizations is implemented to improve the computational efficiency. A comprehensive suite of experiments was conducted on both synthetic and real world data to test the effectiveness and efficiency of MITRE.

Index Terms—Anomaly Detection, Mixed-type Data, Robust Estimation, Expectation Propagation, Variational Inference

1 Introduction

Anomaly detection is an important problem that has received a great deal of attention in recent years. The objective is to automatically detect abnormal patterns and identify unusual instances, so-called anomalies. For example, in signal processing, anomalies could be caused by random hardware failures or sensor faults, whilst anomalies in a credit card transaction dataset could represent fraudulent transactions. Anomaly detection techniques have been widely applied in a variety of domains, including cyber security [1], health monitoring [2], financial systems [3], and military surveillance [4].

Approaches to anomaly detection include distance based [5] [6], local density based [7] [8], one-class classifier based [9] [10], and statistical model based methods [11] [12] [13]. Most of these approaches are designed for single-type datasets, whereas most real world datasets are composed of a mixture of different data types, such as numerical, binary, ordinal, nominal, and count. In the KDD panel discussion [14] and the resulting position paper [15], dealing with mixed-type data was identified as one of the ten most important challenges in data mining for the next decade. However, the direct application of single-type approaches to mixed-type data leads to the loss of significant correlations between attributes, and their extension to mixed-type data is technically challenging. For example, distance based approaches rely on well-defined measures to calculate the proximity between data observations but there is no uniform measure that can be used for mixed-type attributes, while the statistical model based approaches rely on modeling the correlations between different attributes but there is no uniform correlation measure available for mixed-type attributes. The limited number of methods designed for dealing with mixed-type data, including LOADED [16] and RELOADED [17] all focus primarily on computational efficiency and their correlation modeling between mixed-type attributes is heuristically driven, lacking a solid statistical foundation. There are three main challenges for mixed-type anomaly detection: **1) Modeling mutual correlations between mixed-type attributes:** Mixed-type datasets involve more than one confounded dimension of dependency between the attributes so the relationships among these attributes in multivariate types need to be captured; **2) Capturing large variations due to anomalies:** Most existing methods require a pure training dataset in order to learn what constitutes normal behavior. However, in the presence of anomalies, recognizing normal instances is challenging for unsupervised frameworks because these anomalies introduce large variations that can easily bias the estimation of normal patterns; and **3) Analytically intractable posterior inference:** The likelihood of non-Gaussian observations yields an analytically intractable distribution. Therefore, an approximation method is necessary to estimate the inference for the particular observations.

In this paper, a statistical-based approach to address the above challenges is proposed. We begin by presenting a new variant of the generalized linear model (GLM) that can capture the mutual correlations between mixed-type attributes. Specifically, the mixed-type attributes are mapped to latent numerical random variables that are multivariate Gaussian in nature. Each attribute is mapped to a corresponding latent numerical variable via a specific link function, such as a logit function for binary attributes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

and a log function for count attributes. By adopting this strategy, the dependency between mixed-type attributes is captured by the relationship between their latent variables using a variance-covariance matrix. Meanwhile, an "error buffer" component based on the Student-t distribution is incorporated to capture the large variations caused by anomalies. While fitting the data into the model, the error buffer absorbs all errors. The detection process then revisits the error buffer and detects those abnormal instances with irregular magnitudes of error. Unfortunately, the application of GLM and Student-t prior make the inference analytically intractable. We therefore propose an approach that adapts an Integrated-Nested Laplace Approximation (INLA) by applying optimization strategies to approximate the Bayesian inference. An alternative framework that incorporates Expectation-Propagation (EP) [18] and Variational Expectation-Maximization (VEM) framework is also proposed. The main contributions of our study can be summarized as follows:

1) **Constructing a novel unsupervised framework**: A new unsupervised framework capable of performing general purpose anomaly detection on mixed-type data is proposed that does not require labeled training data, which is in practice often difficult to obtain.
2) **Capturing anomalies' large variances and dependencies among mixed-type observations**: The proposed model addresses the two main challenges of detecting anomalies in a mixed-type model, i.e., modeling mutual correlations between mixed-type attributes and capturing large variations due to anomalies.
3) **Designing more effective approaches for Bayesian inference approximation**: Two approaches are proposed to approximate Bayesian inference, namely Integrated Nested Laplace Approximation (INLA) and Expectation Propagation with a Variational-EM framework.
4) **Conducting extensive experiments to validate the effectiveness and efficiency**: Our experimental results demonstrate that our proposed approaches outperformed most of the existing approaches tested on both synthetic and real benchmark datasets, with comparable computational efficiency. The advantages and limitations of the proposed approaches are also explored via an experimental analysis.

The remainder of this paper is organized as follows. Section 2 reviews the existing work in this area and Section 3 presents the problem formulation and the model design. In Section 4, the framework for the anomaly detection process is discussed, while the experiments on both simulated and real datasets are presented in Section 5. The paper concludes with a summary of the research and our finidings in Section 6.

2 Related Work

This section provides an overview of the status of current research on anomaly detection, including both single-type and mixed-type anomaly detection methods.

Single-type Anomaly Detection Methods: Early research on anomaly detection can be categorized into five groups, namely distance-based [5] [6], density-based [7] [8], cluster-based [19], classification-based [9] [10], and statistical-based [11] [12] [13] [20] methods.

Knorr et al. [5] presented the first distance based approach, which detects anomalies by applying a distance threshold. Another early distance-based method was proposed by Ramaswamy et al. [6], who extended the distance criterion by combining it with the k-nearest neighbor (KNN) based method. This category of methods is usually efficient, but the accuracy is compromised when the data distribution is skewed. Besides these distance-based approaches, density based approaches are also popular. For example, the local outlier factor (LOF) [7] and local correlation integral (LOCI) [8] methods are based on estimating the local densities around points of interest and their neighbors.

Other anomaly detection approaches address the problem by framing it as traditional data mining problems. The clustering-based method proposed in [19] first groups similar data and then labels those instances that are not well clustered as anomalies. Various classification-based approaches have also been proposed that assume that the designation of anomalies can be learned by a classification algorithm. This is exemplified by Das et al. [9], who present a one-class SVM based approach, and Roth [10], whose method is based on kernel Fisher discriminants.

Statistical-based approaches assume that the data follow a specific distribution, and detect anomalies by identifying instances with low probability densities. One of the main challenges here is to reduce the well-known masking and swamping effects. Anomalies can bias the estimation of distribution parameters, yielding biased probability densities that cause normal objects to be misidentified as anomalies, or vice versa. To address this issue, a number of methods have been proposed that make different distribution assumptions, including techniques based on the robust Mahalanobis distance [11], direction density ratio estimation [12], and the minimum covariance determinant estimator [13]. Recent advances have generally focused on applying robust statistics for outlier detection [20].

Another approach that often used for outlier detection is to apply robust Principle Component Analysis (PCA) [21] [22] [23]. Particularly we suited to extracting the most significant features from noisy datasets, these methods are either driven by robust statistics, e.g., trimming off extreme observations [21] or using median rather than mean values [22], or operate by directly decomposing the dataset into a low rank matrix and a sparse matrix [23]. In the first case, the outliers will be those data instances with any attributes deviating from a specific threshold value, while the outliers in the latter case are those data instances with any greater value in the sparse matrix.

Mixed-type Anomaly Detection Methods: Real world data usually consist of a mixture of data types, with non-numerical data presenting different features from numerical

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

data. For instance, categorical data has no particular order so it is not possible to quantify differences between data points [24], which means that detection methods that are suitable for numerical data might not necessarily provide a good fit for mixed-type datasets. Tran et al. [25] model heterogeneous datasets using Restricted Boltzmann Machines, where the dependency among data fields is captured by latent binary variables. Although their approach can be utilized as a classifier for discrete outputs or as a regression tool for continuous outputs, it does not explicitly consider any anomalies present in the datasets.

In the research reported in the literature, a popular approach is to process individual data types separately and then integrate the results for each data type to detect anomalies [16] [17] [26] [27].Two mixed-type anomaly detection approaches have been proposed by Otey et al., namely LOADED [16] and RELOADED [17]. LOADED uses an augmented lattice to calculate the support count of the item sets for the categorical attributes and then computes a correlation matrix for the numerical attributes. It detects anomalies by assigning an anomaly score based on the support of the item sets and the level of numerical attributes confirming that correlation. In an effort to improve the performance of LOADED, RELOADED reduces the memory usage by replacing the covariance matrix with a set of classifiers. These two algorithms achieve a high efficiency as targeted, although their detection accuracy could be further improved. Both LOADED and RELOADED are supervised methods and thus require training datasets. Mixed-type data can also be processed by integrating different single-type approaches. Koufakou et al. [26] [28] propose ODMAD for high dimensional datasets, which detects outliers in categorical fields and numerical fields separately. In particular, outliers in categorical fields are detected by counting and outliers in numerical fields are detected based on the data's distance from the center of the numerical fields. Although this method is relatively straightforward, it does not consider the relationships between categorical and numerical fields. Moreover, it requires a good understanding of the instance space in order to feed in several user-defined thresholds to filter out outliers. Tran and Jin [27] apply a $C4.5$ decision tree to symbolic attributes and a Gaussian Mixture Model (GMM) to model numerical fields, with And anomalies being detected by comparing the weighted sum of the score from the decision tree and the score from the GMM to a predetermined threshold. Similar to ODMAD, this method requires extensive fine-tuning work when assigning optimal weights for both scores and selecting a reasonable threshold for filtering out outliers.

Ye et al. [29] adopted a different approach, applying a projected outlier detection method (PODM) that jointly considers discrete fields and continuous fields to detect top-k anomalies. The fundamental principle when detecting anomalies is that an anomalous instance's presence in a lower dimension projection will be abnormally lower than the average. The idea here is thus to convert all continuous fields into discrete values and then partition data space into several cells. Given a set of subspaces that all instances are projected onto, a Gini entropy and an outlying degree are computed to measure whether a particular subspace is an anomaly or not. Finally, the outliers are identified from low density cells in the anomalous subspaces. The main drawback of this method is the need to choose a unit interval, i.e. an equi-width, for discretizing the numerical values. Due to the often widely variable range of the numerical attributes concerned, all these fields have to be preprocessed carefully.

The closest work to the new method proposed here was suggested by Zhang and Jin [30]. They apply the notion of the patterns observed in the majority of the data in terms of their attributes, where the number of patterns corresponds to the number of categorical data fields. Here, a pattern is studied by applying a linear logistic regression where the explanatory variables are numerical attributes and the response variable is a single categorical attribute. The advantage of a regression based model is that it reveals the functional relationship among the attributes [31]. However, although this approach models such relationships among the attributes, the logistic regression is sensitive to outlier effects.

On the other hand, regression based models have been widely studied in robust statistics research. For example, several robust linear regression approaches for numerical data are introduced in literature [32] [33], and Liu [34] also proposes a robust version of logistic regression and proves its capability to tolerate anomalies. Building on the existing work, the model we propose in this paper adopts a robust statistical approach to capture attribute dependencies using an input-output relationship with a Gaussian latent variable. Combining the robust design with the generalized linear models, the proposed approach is capable to handle mixed-type data while maintaining its robustness to anomalies. The new framework aims to provide a high detection accuracy and deliver the results in an acceptable time. The process detects anomalies from the input dataset directly, with no training data set required.

3 Model Design

This section begins by formalizing the problem in 3.1. In 3.2, we discuss the modeling of mixed-attributes in the framework of generalized linear models and an error buffering component to handle anomalous effects. The integrated Bayesian hierarchical model is presented in 3.3.

3.1 Problem Formulation

Consider N instances in a dataset $S = \{s_1 \cdots s_N\}$, in which each instance s has P response (or dependent) variables $\{y_1(s) \cdots y_P(s)\}$ and D explanatory (or independent) variables $\{x_1(s) \cdots x_D(s)\}$. The separation of the response (e.g., a house price) and explanatory (e.g., the house's size and number of rooms) variables is decided based on users' domain knowledge, and all the variables could be regarded as response (dependent) variables as a special case. The dependent variables could consist of different data types, combining numerical, binary, and/or categorical variables,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

whilst the explanatory attributes are typically set to be numerical. The objective is to model the data distribution and identify those instances that contain abnormal response variables or explanatory attributes.

Types of Anomalies: Since the attributes have been separated into two types, the anomalies can also be introduced as either abnormal response variables or unusual explanatory attributes. Thus, we define three types of anomalies based on their originating attribute groups:

1) **Type I Anomalies are** caused by abnormal values in the response variables.
2) **Type II Anomalies are** caused by abnormal values in the explanatory attributes.
3) **Type III Anomalies are** caused by abnormal values for both response variables and explanatory attributes.

Any object that has attributes that cause it to behave as if it belongs to one of the above three categories is defined as an anomaly. An anomalous object usually deviates considerably from the normal trends in the data and can hence be detected using our statistical model.

Predictive Process: The first step utilizes numerical response variables, which are typically assumed to follow a Gaussian distribution model. Thus, the Gaussian predictive process can be applied here. The following regression formulation represents the behavior of the instances:

$$Y(s) = X(s)\beta + \omega(s) + \varepsilon(s) \tag{1}$$

This formulation implies that similar instances should have similar explanatory attributes. The regression effect β is a $P \times D$ matrix, which represents the weights of the explanatory attributes with regard to the response variables. The dependency effect $\omega(s)$ is a Gaussian process used to capture the correlation between the response variables and a local adjustment is provided for each response attribute. The error effect $\varepsilon(s)$ captures the difference between the actual instance behavior and normal behavior. The instances are assumed to be independent and identically distributed (*i.i.d.*), which introduces the Gaussian likelihood as

$$\pi(Y(s)|\eta(s)) \sim \mathcal{N}(Y(s)|\eta(s), \sigma^2_{num}), \tag{2}$$

where $\eta(s) = X(s)\beta + \omega(s) + \varepsilon(s)$, and σ^2_{num} is set to a small number in order to allow the random effects for ω and ε to be captured.

3.2 GLM and Robust Error Buffering

The underlying concept of GLM (Generalized Linear Model) is to assume that non-numerical type data are generated from a particular exponential family distribution. Taking the binary response type as an example, each response variable is assumed to follow a Bernoulli distribution, such that $\pi(Y(s)|\eta(s)) \sim Bernoulli(g(\eta(s)))$, where g is a logit link function that converts the numerical likelihood value to the success probability of the Bernoulli distribution. In this case, a sigmoid function is applied for the conversion, e.g., $g(x) = \frac{1}{1+exp(-x)}$. GLM can handle not only binary data, but also count, categorical, multinomial, and other data types. In this work, we consider 4 data different types, namely numerical, binary, count, and categorical. The specific usage of GLM in our model will be discussed in the next subsection.

One of the major components in the proposed new algorithm is the robust error buffer. A latent random variable is included to absorb the error effect caused by measurement error, noise, or abnormal behaviors. The purpose of this mechanism is to separate the expected normal behavior from the errors. Instead of a simple Gaussian distribution, a Student-t distribution is utilized to model the error variation ε. A Student-t distribution has a heavier tail than a Gaussian distribution, and is widely used in robust statistics [15]. The heaviness of the tail is controlled by setting the number of degrees of freedom: when the degree of freedom approaches infinity, the Student-t distribution becomes equivalent to a Gaussian distribution. The probability density function of a Student-t distribution $st(0, df, \sigma)$ is defined as

$$p(\varepsilon) = \frac{\Gamma(\frac{df+1}{2})}{\Gamma(df/2)}(\frac{1}{\pi df \sigma})^{\frac{1}{2}}(1 + \frac{\varepsilon^2}{df\sigma})^{-\frac{df}{2}-\frac{1}{2}}, \tag{3}$$

where df represents the degrees of freedom, σ is the scale parameter, and Γ is the gamma function. Our model treats the error effect $\varepsilon(s)$ as a zero mean Student-t process, with a diagonal covariance matrix and a preset degree of freedom. There are two benefits to be gained by incorporating this error buffer in the model. First, the parameter estimation becomes robust and the normal behavior is modeled more accurately. Second, the errors are absorbed by this latent variable, making it possible to detect anomalies by checking the values of the variables.

3.3 A Bayesian Hierarchical Model

Integrating the components introduced in the above subsections allows us to complete the design of the new algorithm. Figure 1 shows the graphical representation of our model.

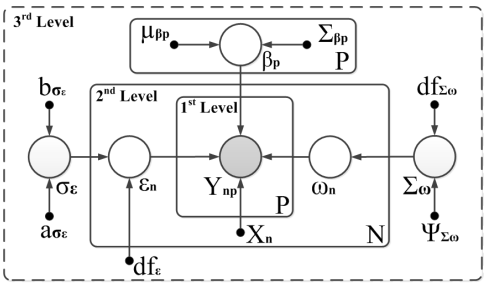


Fig. 1: Graphical Model Representation

The proposed model is based on a Bayesian hierarchical model, which enables the parameters to be automatically learned while also reserving the option for users to assign values to the hyper-parameters based on their prior knowledge.

The first (observation) level of the hierarchical model captures the relationships between the response variables. This level refers to the predictive process and the GLM, and models the relationships between the latent effects and the response variables. Here, 4 different types of data are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

TABLE 1: GLM Information

Type	Likelihood	Link Function
Numerical	Gaussian	Identity
Binary	Bernoulli	Logit Function
Count	Poisson	Log Function
Categorical	Nominal	Logit Function

considered: numerical, binary, count, and categorical. Each of the data types is associated with a specific type of likelihood. We model these data types in the traditional GLM manner, which assumes Gaussian, Bernoulli, and Poisson distributions for numerical, binary, and count, respectively. For categorical data, we follow the modeling strategy described in [35]. The categorical response variable is extended to K binary variables, where K is the number of categories of the variable. Table 1 lists the GLM likelihood and link function of each data type.

The second (latent variable) level is the latent variable level. This level contains the latent elements that refer to the effects of the error buffer and the correlation effect i.e., ω and ε. The main purpose of this level is to model the relationships between the latent variables and the parameters. More specifically, we can form the following equations:

$$\omega(s) \sim \mathcal{N}(\omega(s)|0, \Sigma_\omega), \quad (4)$$

$$\varepsilon(s) \sim St(\varepsilon(s)|0, \sigma_\varepsilon, df), \quad (5)$$

where Σ_ω is the covariance matrix used to model the covariance between the response variables, σ_ε is a diagonal covariance matrix that indicates the variances of the error effects, and df denotes the degree of freedom parameter. For convenience, we will use the symbol ν to denote the latent variable set.

The third (parameter) level defines the regression coefficients and conjugate priors for the model parameters, including the covariance matrix of ω and the covariance matrix of ε, designated Σ_ω and σ_ε, respectively.

The prior distribution of the regression coefficients β can be represented by

$$\beta_p \sim \mathcal{N}(\beta|\mu_{\beta p}, \Sigma_{\beta p}), \quad (6)$$

where β_p is the regression coefficient corresponding to the p-th response variable, and $\mu_{\beta p}$ and $\Sigma_{\beta p}$ are the hyper-parameters that define the Gaussian distribution of each β_p.

To reduce the dimensionality of θ, we retain only the variance of ω and ε in each response variable and the correlation between response variables:

$$\sigma^2_{\varepsilon p} \sim IG(a_{\varepsilon p}, b_{\varepsilon p}), \quad (7)$$

$$\Sigma_\omega \sim IW(\Phi, df_\omega), \quad (8)$$

The variance $\sigma^2_{\varepsilon p}$ for each response variable is assigned an inverse gamma distribution, and the covariance matrix of ω is assigned an inverse Wishart distribution. The symbols $a_{\varepsilon p}$, $b_{\varepsilon p}$, Φ ,and df_ω denote the hyper-parameters of these prior distributions. The model is now well defined and the next step is to fit the model based on the dataset. In the next section, we introduce the entire anomaly detection framework and describe the method used to approximate the Bayesian inference for the model.

4 Framework and Inference

This section presents the framework of the anomaly detection process, the statistical inference for the model, the computational cost, and the optimization schemes. Two new frameworks have been devised in this paper, one utilizing the INLA framework and the other an EP framework.

4.1 INLA Framework

First, we propose an approach adapted from the Integrated Nested Laplace Approximation (INLA) [36], which is a relatively new technique for approximating Bayesian inference. The Laplace approximation (LA) method approximates an arbitrary distribution to Gaussian by taking the mode as the mean and the second order derivative at the mode as the variance (or covariance matrix in multivariate distribution). The general idea of INLA is to use the Laplace Approximation iteratively to approximate the marginal posteriors for the latent variables. The advantage here is that the fitting process of INLA is particularly effective in a lower dimensional space for the model parameters.

4.1.1 Framework

Algorithm 1 MITRE-INLA

Require: The response variables Y and explanatory attributes X
Ensure: The anomalous instances
set $\theta = \theta_0$
while $\theta \neq argmax_\theta(p(\theta|Y))$ **do**
  set$\nu = \nu_0$
  while $\nu \neq argmax_\nu(p(\nu|Y, \theta))$ **do**
    $\nu = update\_\nu$
  end while
  $\hat{\nu} = \nu$
  $L = likelihoodofp(\theta|Y, \hat{\nu})$
  $\theta = update\_\theta(L)$
end while
θ_{sample} = sample from neighborhood of θ
set $L_{\theta_{samples}}, \hat{\nu}_{\theta_{sample}} = \phi$
for all θ_s in $\theta_{samples}$ **do**
  $\hat{\nu}_{\theta_s} = argmax_\nu(p(\nu|Y, \theta_s))$
  $L_{\theta_s} = likelihoodofp(\theta|Y, \hat{\nu}_{\theta,s})$
  put $\hat{\nu}_{\theta_s}$ into $\hat{\nu}_{\theta_{samples}}$
  put L_{θ_s} into $L_{\theta_{samples}}$
end for
$weight = normalize(L_{\theta_{samples}})$
$\nu^* = \hat{\nu}_{\theta_{samples}} * weight$
$\varepsilon^* = getErrorBuffer(\nu^*)$
set $AnomalySet = \phi$
for all ε^*_s in ε^* **do**
  if $\varepsilon^*_s > ErrorThreshold$ **then**
    put s in $AnomalySet$
  end if
end for
return $AnomalySet$

Algorithm 1 presents the framework for MITRE-INLA, which is composed of three major components: the Laplace approximation, variable estimation, and anomaly detection.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

Phase 1 - Laplace Approximation. Steps 1 to 10 show how the INLA framework is established by two Laplace approximations in a nested structure. The outer loop performs a maximum *a posteriori* (MAP) to θ. Since we can represent the posterior distribution of θ in the form:

$$p(\theta|Y) \propto \frac{p(\nu, Y, \theta)}{p(\nu|Y, \theta)}, \tag{9}$$

and our objective is to maximize $p(\theta|Y)$, we can treat the posterior density function $p(\theta|Y)$ as an objective function and this then becomes an optimization problem. The next step is to assign values for each input to this objective function $p(\theta|Y)$. Thus, the inner loop (steps 4-6) runs for the Laplace approximation to $p(\nu|Y, \theta)$. Applying a Taylor expansion to $p(\nu|Y, \theta)$, we can achieve an analytical formulation that restructures this density function into the quadratic form:

$$p(\nu|Y, \theta) = -\frac{1}{2}\nu^T Q\nu + \nu^T b. \tag{10}$$

Then, for each iteration at step 5, the latent variable set ν can be updated by $\nu = Q^{-1}b$. After a few iterations, ν will converge to a local optimum. This updating method, known as Iterative Reweighted Least-Squares (IRLS) [37], usually converges within 5 iterations. Steps 7-9 calculate the value of the objective function $p(\theta|Y)$ at the local optimum $\hat{\nu}$, and update θ according to this value. The iterations are continued until θ converges.

Phase 2 - Variable Estimation. After obtaining the mode of $p(\theta|Y)$, say $\hat{\theta}$, samples can be collected from the neighbors of $\hat{\theta}$ in the space of θ and used to estimate the optimum values of θ and ν. This is similar to the importance sampling [38] approach often used for numerical analysis, the difference being that samples are only collected from the mode region in the space. Steps 11-19 demonstrate this process.

Phase 3 - Anomaly Detection. Finally, steps 20-28 show the process used to detect anomalies. Having identified the optimum ν, say ν^*, we are able to use the optimized latent variable set to perform anomaly detection. We begin by extracting the fitted error buffer ε^* from ν^*, and examining its contents. Step 24 indicates how the anomalies are detected in terms of a pre-determined threshold. This threshold is typically set to 3 times the standard deviation, i.e., the absolute Z-score equals 3, just as in labeling anomalies for a Gaussian distribution.

4.1.2 Computational Cost and Optimization

The computational cost is usually a concern for statistical modeling techniques; if the method is to be applied as an online method, the efficiency becomes especially important. Here, the strategy is to approximate complex computations, accepting a slight drop in accuracy to gain a significant increase in efficiency. These optimizations have been successfully tested experimentally, as described in the Experimental Results section.

Latent Computational Optimization: In Algorithm 1, step 5 is a major bottleneck in the framework shown . The high dimensionality of the latent variable set makes the computation of the matrix inversion very slow. To optimize this step, the update is approximated by separating ν into $\varepsilon, \omega, \beta$ and then updating these three variables iteratively, as in the Gibbs sampling method. Algorithm 2 demonstrates the idea behind the approximation process. Steps 1-9 show how the original process breaks down into three smaller processes. Steps 2, 5, and 8 update the latent variables in the same sense as the original one. A Taylor expansion is performed on each of the three latent variables separately and inserted into the Gaussian quadratic form in equation (10) and updated by IRLS, i.e. iteratively performing $\beta = Q_\beta^{-1}b_\beta, \varepsilon = Q_\varepsilon^{-1}b_\varepsilon$, and $\omega = Q_\omega^{-1}b_\omega$. In each call on $update\_\nu$, two of the variables are fixed and the third updated, substantially reducing the computational cost as a result. The complexity of the original INLA update is $O((P(2N + D))^3)$, which refers to the size of the latent variable set in the matrix inversion, while the complexity of the optimized update is reduced to $O(N^3)$.

Algorithm 2 $update\_\nu$

Require: The original latent variable $\varepsilon, \omega, \beta$
Ensure: The updated latent variable $\varepsilon_{new}, \omega_{new}, \beta_{new}$
1: **while** $\beta \neq argmax_\beta(p(\beta|Y, \theta, \varepsilon, \omega))$ **do**
2: $\beta = update\_\beta(\varepsilon, \omega)$
3: **end while**
4: **while** $\varepsilon \neq argmax_\varepsilon(p(\varepsilon|Y, \theta, \beta, \omega))$ **do**
5: $\varepsilon = update\_\varepsilon(\beta, \omega)$
6: **end while**
7: **while** $\omega \neq argmax_\omega(p(\omega|Y, \theta, \varepsilon, \beta))$ **do**
8: $\omega = update\_\omega(\varepsilon, \beta)$
9: **end while**
10: **return** $\varepsilon_{new} = \varepsilon, \omega_{new} = \omega, \beta_{new} = \beta$

Approximate Parameter Estimation: Another bottleneck in Algorithm 1 is that when the dimension of the parameter space is huge, sampling and evaluating the weight from the $\hat{\theta}$ neighborhood is computationally intensive. We therefore approximate the optimum estimation by reducing the size of the samples in step 11. Although the estimated parameters will not exactly match the optimum, the latent variable set still follows approximately the same trend if the estimated parameters are close to the optimum, and our experience indicates that the approximated $\hat{\theta}$ is usually sufficiently close to the optimum solution of θ. Since the anomaly detection framework is only interested in the latent variables, having a minor bias in the parameter estimation will not actually affect the detection results.

4.2 EP Framework

Although the INLA method can provide a good approximation of the inference, the grid integration scheme in the neighborhood of θ (Line 11 of Algorithm 1) introduces a significant growth in the computational cost when the dimension of θ becomes large [39]. As an alternative solution, we therefore developed an approximate Bayesian inference approach for the MITRE model using Expectation Propagation (EP) with the Variational-EM framework. EP has been shown to give results that outperform Laplace's

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

method on accuracy in terms of predictive distributions and marginal likelihood estimations [18].

4.2.1 Framework

Under this framework, we can apply EP for the inference of the latent variables. Based on mean-field theory, the inference is embedded into an Expectation-Maximization loop to estimate the optimal model parameters θ. Algorithm 3 presents the framework of MITRE-EP.

Phase 1 - Approximate Inference. Steps 1 to 10 show the approximate inference using EP-EM, which will be introduced in the next subsection. The inner loop performs Expectation Propagation (EP) to estimate the latent variables, and the outer loop performs Expectation Maximization (EM) algorithm to estimate the parameter set θ. The details of this process are discussed in the next subsection.

Phase 2 - Anomaly Detection. This phase applies the same procedures as in the INLA framework. We extract the fitted error buffer ε^* from the estimated ν^*, and examine its contents to detect any anomalies. Steps 15-17 indicate how the anomalies are detected in terms of a pre-determined threshold. This threshold is typically set to 3 times the standard deviation, i.e., the Z-score equals 3, just as when labeling the anomalies for a Gaussian distribution.

Algorithm 3 MITRE-EP

Require: The response variables Y and explanatory attributes X
Ensure: The anomalous instances
1: set $\theta = \theta_0$
2: **while** $\theta \neq argmax_\theta(p(\theta|Y))$ **do**
3: set $\nu = \nu_0$
4: **while** $\nu \neq argmax_\nu(p(\nu|Y,\theta))$ **do**
5: $\nu = update\_\nu$
6: **end while**
7: $\hat{\nu} = \nu$
8: $L = likelihoodofp(\theta|Y,\hat{\nu})$
9: $\theta = update\_\theta(L)$
10: **end while**
11: set $\nu^* = mode(p(\nu|Y,\theta))$
12: $\varepsilon^* = getErrorBuffer(\nu^*)$
13: set $AnomalySet = \phi$
14: **for all** ε_s^* in ε^* **do**
15: **if** $\varepsilon_s^* > ErrorThreshold$ **then**
16: put s in $AnomalySet$
17: **end if**
18: **end for**
19: **return** $AnomalySet$

4.2.2 Approximate Inference

The E-step of the EM algorithm estimates the expectation of the posterior distribution $p(\theta|Y)$. Applying Bayes' theorem, the posterior is shown to be proportional to the joint distribution.

$$p(\theta|Y) \quad \propto \quad p(Y|\theta)p(\theta) \tag{11}$$

Thus, the expectation of the complete-data log posterior for a general θ value is given by

$$\mathcal{Q}(\theta, \theta^{old}) = \mathbb{E}_\nu[\ln p(\nu, Y|\theta)|\theta^{old}] + \ln p(\theta) + Const$$

where θ^{old} denotes the parameter values in the current iteration, and $Const$ presents the constant that does not depend on θ. In the M-step, the updated parameter estimation θ^{new} is determined by maximizing the expectation $\mathcal{Q}$, such that

$$\theta^{new} = \arg\max_\theta \mathcal{Q}(\theta, \theta^{old}) \tag{12}$$

The first objective is to estimate the expectation $\mathbb{E}_\nu[\ln p(\nu, Y|\theta)|\theta^{old}]$. As mentioned above, the mixed-type GLM and the student-t prior introduce a complicated joint distribution, rendering the inference intractable. Therefore, Expectation Propagation is applied here to approximate the inference.

When initiating the inner EP process, we begin by expanding $p(\nu, Y|\theta)$:

$$p(\nu, Y|\theta) = p(Y|\nu, \theta)p(\nu|\theta) \tag{13}$$

According to the model structure shown in Fig 1, the likelihood component can be written in a product form:

$$p(Y|\nu, \theta) = \prod_{n=1}^{N} \prod_{p=1}^{P} p(y_{np}|\beta_p, \omega_{np}, \varepsilon_{np}) \tag{14}$$

and

$$p(\nu|\theta) = \prod_{n=1}^{N} p(\omega_n|\theta) \prod_{p=1}^{P} p(\varepsilon_{np}|\theta) \tag{15}$$

Thus, the joint distribution becomes

$$p(\nu, Y|\theta) = \prod_{n=1}^{N} p(\omega_n|\theta) \prod_{p=1}^{P} p(\varepsilon_{np}|\theta)p(y|\beta_p, \omega_{np}, \varepsilon_{np})$$

By utilizing EP, the complicated distribution $p(\nu, Y|\theta)$ can be approximated to a Gaussian distribution. The approximated Gaussian is denoted by $q(\nu)$:

$$\begin{aligned} q(\nu) &= \frac{1}{Z} q_0(\nu) \prod_{n=1}^{N} \prod_{p=1}^{P} q_{np}(\nu) \\ &= \mathcal{N}(\nu|h, C) \end{aligned} \tag{16}$$

where $q_0(\nu) = \prod_{n=1}^{N} p(\omega|\theta)$ is the prior, and each $q_n(\nu)$ approximates the product of the likelihood and the student-t prior according to the n-th entity, i.e.,

$$q_{np}(\nu) = \mathcal{N}(\nu|m_{np}, R_{np}) \approx p(\varepsilon_{np}|\theta)p(y|\beta_p, \omega_{np}, \varepsilon_{np})$$

Since GLM is integrated to the model, this yields $p(y_n|\beta, \omega_n, \varepsilon_n)$, with different forms for different data types. Specifically, each instance n may consist of P response variables in variant types. To denote this, we use the subscript p to denote the index of the response variables of an instance. For example, for numerical data, the likelihood is assumed to follow a Gaussian distribution

$$p(y_{np}|\beta_p, \omega_{np}, \varepsilon_{np}) = \mathcal{N}(y_{np}|x_n\beta_p + \omega_{np} + \varepsilon_{np}, \lambda) \tag{17}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

To make the equations terse, we use $\eta_{np}\nu$ to denote $x_n\beta_p + \omega_{np} + \varepsilon_{np}$, where η_{np} is a $(2+D)*P$ vector. The EP algorithm iterates over all elements with regard to the subscripts n and p, and updates the approximated distribution using the deletion/inclusion scheme described in [18], i.e., deletion, moment matching, and updating. The following process is performed until m and R converge:

1) **Deletion:** remove q_{np} from the full proposal distribution q

$$q^{\backslash n,p}(\nu) = \mathcal{N}(\nu|h^{\backslash n,p}, C^{\backslash n,p}) \tag{18}$$

where

$$\begin{aligned} h^{\backslash n,p} &= h + C^{\backslash n,p}R_{np}^{-1}(h - m_{np}) && (19)\\ C^{\backslash n,p} &= (C^{-1} - R_{np}^{-1})^{-1} && (20)\end{aligned}$$

2) **Moment Matching:** find the new approximated proposal distribution $q^{(new)} \sim \mathcal{N}(h^{(new)}, C^{(new)})$ by matching the moment of

$$q_{prop}(\nu) = q^{\backslash n,p}(\nu)p(\varepsilon_{np}|\theta)p(y|\beta_p, \omega_{np}, \varepsilon_{np})$$

The mean and variance of $q^{\backslash n,p}$ can be found using Iterated Re-weighted Least Squares (IRLS). By expanding the Taylor series to the combined distribution at the point ν_0,

$$\begin{aligned} q_{prop}(\nu) = q_{prop}(\nu_0) + \nabla_\nu q_{prop}(\nu_0)(\nu - \nu_0) \\ + \frac{1}{2}\nabla\nabla_\nu q_{prop}(\nu_0)(\nu - \nu_0) \end{aligned} \tag{21}$$

Rearrange the series into squared form

$$q_{prop}(\nu) = -\frac{1}{2}\nu^T Q\nu + b\nu + const$$

such that a local optimum can be found at $Q^{-1}b$. By matching the coefficient of the above equations,

$$\begin{aligned} b(\nu_0) &= \nabla\nabla_\nu q_{prop}(\nu_0)\nu_0 - \nabla_\nu q_{prop}(\nu_0)\\ Q(\nu_0) &= \nabla\nabla_\nu q_{prop}(\nu_0)\end{aligned}$$

IRLS finds the mode of q_{prop} iteratively by setting

$$\nu^{(i+1)} = Q^{-1}(\nu^{(i)})b(\nu^{(i)}) \tag{22}$$

in each iteration, given a starting point $\nu^{(0)}$.
Since different likelihood functions are assumed for different data types, this step is handled in various ways according to the corresponding data type. The gradient and Hessian of the numerical likelihood are as follows:

$$\begin{aligned} \nabla_\nu q_{prop}(\nu) &= \frac{-\eta_{np}}{\lambda}(\eta_{np}^T\nu - Y_{np}) \\ &- C^{\backslash n,p^{-1}}(\nu - h^{\backslash n,p}) \\ &- \frac{(df+1)\varepsilon_{np}}{df\sigma_\varepsilon + \varepsilon_{np}^2}\eta_{\varepsilon_{np}} \\ \nabla\nabla_\nu q_{prop}(\nu) &= \frac{-1}{\lambda}\eta_{np}\eta_{np}^T - C^{\backslash n,p^{-1}} \\ &- \frac{(df+1)(\sigma_\varepsilon df - \varepsilon_{np}^2)}{(\sigma_\varepsilon df + \varepsilon_{np}^2)^2}\eta_{\varepsilon_{np}}\eta_{\varepsilon_{np}}^T\end{aligned}$$

The equations for the other data types can be derived in a similar way.

3) **Update**: update each approximated distribution by

$$q_{np}(\nu) = \frac{q^{(new)}}{q^{\backslash n,p}} \tag{23}$$

From equation (23) and the definition above we have

$$\begin{aligned} R_{np}^{-1} &= C^{(new)^{-1}} - C^{\backslash n,p^{-1}} && (24)\\ m_{np} &= R_{np}(C^{(new)^{-1}}h^{(new)} - C^{\backslash n,p^{-1}}h^{\backslash n,p})\end{aligned}$$

After the expectation for the latent variable ν has been approximated, the first part of the expectation can be formulated using the following expression:

$$\begin{aligned} &\mathbb{E}_\nu[\ln p(\nu, Y|\theta)|\theta^{old}] \\ =\ &\sum_{n=1}^{N}\ln\mathcal{N}(\hat{\omega}_n|0,\Sigma_\omega) + \sum_{n=1}^{N}\sum_{p=1}^{P}\ln\mathcal{ST}(\hat{\varepsilon}_{np}|0,\sigma_{\varepsilon_p})\end{aligned}$$

where $\hat{\nu}$ is the expected value of ν.
The expectation of the log distribution function θ is

$$\begin{aligned} &\mathcal{Q}(\theta,\theta^{old}) \\ =\ &\mathbb{E}_\nu[\ln p(\nu, Y|\theta)|\theta^{old}] + \ln p(\theta) \\ =\ &\sum_{n=1}^{N}\sum_{p=1}^{P}\ln p(y_{np}|\beta_p,\hat{\omega}_{np},\hat{\varepsilon}_{np}) + \sum_{p=1}^{P}\ln\mathcal{N}(\beta_p|\mu_{\beta p},\Sigma_{\beta p}) \\ +\ &\sum_{n=1}^{N}\ln\mathcal{N}(\hat{\omega}_n|0,\Sigma_\omega) + \sum_{n=1}^{N}\sum_{p=1}^{P}\ln\mathcal{ST}(\hat{\varepsilon}_{np}|0,\sigma_{\varepsilon_p}) \\ +\ &\ln IW(\Sigma_\omega|\Phi,df_\omega) + \sum_{p=1}^{P}\ln IG(\sigma_{\varepsilon_p}|a_{\varepsilon_p},b_{\varepsilon_p})\end{aligned} \tag{25}$$

To make the statement clearer, $\mathcal{Q}(\theta,\theta^{old})$ is separated into β, ω ,and ε components.

$$\begin{aligned} &\mathcal{Q}^{(\beta)}(\theta,\theta^{old}) \\ &= \sum_{p=1}^{P}\sum_{n=1}^{N}\ln p(Y_{np}|X_n\beta_p + \hat{\omega}_{np} + \hat{\varepsilon}_{np}) + \sum_{p=1}^{P}\ln p(\beta_p)\end{aligned}$$

$$\begin{aligned} &\mathcal{Q}^{(\omega)}(\theta,\theta^{old}) \\ &= \sum_{n=1}^{N}\left(-\ln 2\pi - \frac{1}{2}\ln|\Sigma_\omega| - \frac{1}{2}\hat{\omega}_n^T\Sigma_\omega\hat{\omega}_n\right) \\ &+ \frac{df_\omega}{2}|\Phi| - df_\omega\ln 2 - \ln\Gamma_2(\frac{df_\omega}{2}) \\ &\qquad - \frac{df_\omega+3}{2}\ln|\Sigma_\omega| - \frac{1}{2}tr(\Phi\Sigma_\omega^{-1})\end{aligned}$$

$$\begin{aligned} &\mathcal{Q}^{(\varepsilon)}(\theta,\theta^{old}) \\ &= \sum_{n=1}^{N}\sum_{p=1}^{P}\left(\ln\Gamma(\frac{df+1}{2}) - \ln\Gamma(\frac{df}{2}) - \frac{1}{2}\ln\pi df\right. \\ &\left. -\frac{1}{2}\ln\sigma_{\varepsilon_p}^2 + \frac{df+1}{2}\ln\left(1 + \frac{\hat{\varepsilon}_{np}^2}{2\sigma_{\varepsilon_p}^2}\right)\right) \\ &+ \sum_{p=1}^{P}\left(a\ln b - \ln\Gamma(a) - (a+1)\ln\sigma_{\varepsilon_p}^2 - \frac{b}{\sigma_{\varepsilon_p}^2}\right)\end{aligned}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

The M-Step maximizes the objective by finding the root of $\mathcal{Q}(\theta, \theta^{old})$. Because this is also an intractable problem, IRLS is applied once again here to seek an approximated solution. Iteratively updating the value by inputing the gradient and Hessian from equation 22, the root of θ can be approximated.

For $\sigma^2_{\varepsilon p}$ the gradient and Hessian are:

$$\frac{\partial}{\partial \sigma^2_{\varepsilon p}} \mathcal{Q}^{(\varepsilon)}(\theta, \theta^{old}) = -\frac{N}{2\sigma^2_{\varepsilon p}} + (\frac{df+1}{2}) \sum_{n=1}^{N} \left(\frac{-\hat{\varepsilon}^2_{np}}{4\sigma^2_{\varepsilon p} - 2\hat{\varepsilon}^2_{np}} \right) + \left(\frac{-(a+1)}{\sigma^2_{\varepsilon p}} + \frac{2b}{(\sigma^2_{\varepsilon p})^2} \right) \quad (26)$$

$$\frac{\partial^2}{\partial^2 \sigma^2_{\varepsilon p}} \mathcal{Q}^{(\varepsilon)}(\theta, \theta^{old}) = \frac{2N}{4(\sigma^2_{\varepsilon p})^2} + (\frac{df+1}{2}) \sum_{n=1}^{N} \left(\frac{4\sigma^2_{\varepsilon p}\hat{\varepsilon}^2_{np} - (\hat{\varepsilon}^2_{np})^2}{(4\sigma^2_{\varepsilon p} - 2\hat{\varepsilon}^2_{np})^2} \right) + \frac{a+1}{(\sigma^2_{\varepsilon p})^2} - \frac{4b}{(\sigma^2_{\varepsilon p})^3} \quad (27)$$

Since β corresponds to different likelihood functions for different data types, the maximization for each type can be calculated separately. Here we show only the equations for numerical data, for other data types, the β can be approximated using Laplace approximation (see supplemental material). For the β_p corresponding to the numerical data type, the root can be found by setting the first-order derivation to zero. Thus, β_p can be updated by

$$\beta_p^{(new)} = (\frac{1}{\lambda} \sum_{n=1}^{N} X_n^T X_n + \Sigma^{-1}_{\beta_p})^{-1} \times (\frac{1}{\lambda} \sum_{n=1}^{N} X_n^T (Y_{np} - \hat{\omega}_{np} - \hat{\varepsilon}_{np}) + \Sigma^{-1}_{\beta_p} \mu_{\beta_p}) \quad (28)$$

4.2.3 Computational Optimizations

In order to further boost the efficiency of the framework, several optimization schemes are proposed in this subsection. Here, the strategy is to approximate these complex computations, accepting a slight drop in accuracy in order to gain a significant increase in efficiency. These optimizations have also been successfully tested experimentally, as described in the Experimental Results section.

Correlation Parameter Reduction: Since the complexity of the process is proportional to the dimensionality of the parameters, one way to reduce the complexity is to reduce the number of parameters. For this optimization, we applied a Mutual Information [40] method to calculate the scores of the dependencies between each pair of the response attributes. By applying a user-defined parameter K, it is only necessary to consider the top K attribute correlations to be fitted. This approximation reduces the correlation parameter from $\binom{P}{2}$ to K. When P is a large number, this approximation significantly reduces the computational cost.

Sub-sampling Fitting: When the data size is large, we can further reduce the complexity by sampling only a small portion of the data and then detect the anomalies using the model built by the samples. When the size of these sampled instances and the number of sample batches are sufficient, the accuracy is maintained. This optimization also appies to the INLA based framework.

5 Experiments

Comprehensive experiments on MITRE were conducted to evaluate the following performance elements: detection accuracy, time efficiency, and impact of parameters. The results of the experimental analyses are presented in this section and organized as follows: Section 5.1 introduces the benchmark approaches. Section 5.2 discusses the detection accuracy, time efficiency, and the impact of parameters with synthetic data sets, and Section 5.3 provides an in-depth evaluation of MITRE's effectiveness when applied to real-life data sets. The results of these analyses are discussed in Section 5.4. All sets of the experiment were conducted on a Windows 7 machine with a 2.4 GHz Intel Dual Core CPU and 8GB of RAM.

5.1 Benchmark Approaches

Eight benchmark approaches were evaluated, namely LOADED [16], RELOADED [17], KNN-CT, LOF-CECT, OCS-PCT, OCS-RBF, FB-LOF [41], and ODMAD [26]. LOADED, RELOADED, and ODMAD are mixed-type anomaly detection methods; OCS-RBF (One-class SVM with RBF kernel) and FB-LOF (Feature bagging with an LOF base) are general numerical type anomaly detection methods. For OCS-RBF and FB-LOF, we preprocessed the dataset by converting categorical fields into their binary representation, and performing min-max normalization on all fields. The remaining three methods are all integrated single-type anomaly detection methods made up of combinations of six single-type anomaly detection methods, including three numerical anomaly detection methods (KNN, LOF [7] and OCS (One-class SVM) [9]) and three categorical anomaly detection methods (CT, CECT and PCT, all from [42]). Das and Schneider [42] have shown that these methods outperformed other categorical methods, leading to their selection as the benchmark methods for the categorical attributes in the current study. The integrated methods performed the detection procedures separately, and combined the scores into the same measure via a normalization process. For both LOADED and RELOADED, popular settings of the model parameters (correlation threshold = [0.1, 0.2, 0.3, 0.5, 0.8, 1]; frequency threshold = [0, 10, 20]; τ = [1,2,3,5]) were utilized, and the best results for each dataset reported here based on the true anomaly labels. For the other three approaches, the parameters were selected based on 10-fold cross validations. For ODMAD, we set the $minsup$ value to be the reciprocal of the number of categories of each categorical field.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

5.2 Synthetic Study

For the synthetic data study, we examined the detection accuracy of the proposed frameworks, compared the time costs against the benchmark methods, and analyzed the impact of the parameter settings.

5.2.1 Data sets

The synthetic data were generated based on the following model:

$$Z(s) = X(s)\beta + \omega(s) \tag{29}$$

We first generated $N \times D$ explanatory attributes X from a Gaussian distribution, with a set of $\beta : D \times P$ and the covariance between the attributes, to obtain a set of $Z = [Z_1, \ldots, Z_P]$. For each Z_i, we converted the Z_i to different types, such that

$$Y(s) = g_{type}(Z(s)), \tag{30}$$

where g is the link function for the specific types, such as binary or categorical. The anomalies were injected by randomly shifting the values of $Y(s)$ by a specific amount, for example, by swapping the classes of the categorical observations. In the following experiments, we generated a variety synthetic datasets according to the objective of each test. Each dataset was generated to contain 8-10% anomalous instances.

5.2.2 Detection Accuracy

In this set of experiments, we tested the model inference performance on 4 sets of synthetic data based on different combinations of data types, namely SynNB, SynNC, SynBC, and SynNBC. The symbols N, B, and C refer to numerical, binary, and categorical data types, respectively. The detection accuracy was examined among the synthetic datasets as shown in Table 2. MITRE-EP and MITRE-INLA significantly outperformed the other benchmark methods because there was a strong input-output relation in these simulated datasets. Although a synthetic study is not always convincing due to the presumptions involved in generating the data, these results clearly demonstrate that when the input-output relationships are strong and the pre-knowledge is available to the dataset, MITRE is capable of delivering markedly better results than any of the benchmark methods tested.

5.2.3 Time Cost

This set of experiments compared the time costs incurred by MITRE and the benchmark methods. We conducted these experiments on synthetic datasets in which the normal instances were generated based on a GLM that models mixed-type attributes, and the anomalous instances were generated by random shifting. Table 3 shows the time cost comparison among the various methods for datasets with different instance sizes. Experiments that ran over 2 hours are considered as failure. Overall, although our approach suffered from a higher time cost than the benchmark methods, it delivered much higher detection accuracy in a comparable time as discussed in Section 5.2.2 and the later experimental results for real-life data. Table 4 shows the time consumption with increasing size of P. Most of the benchmark methods failed to handle the higher dimension data. For example, ODMAD's computational cost grew exponentially with the number of categorical fields due to its exhaustive searching scheme. MITRE-INLA also suffered from a high dimension of P size, due to the θ estimation process (Section 4.1.1 Phase 2). MITRE-EP demonstrated its ability to accomplish a run with a P size of 100 within one hour.

TABLE 3: Time Cost Comparison in terms of Size of N (seconds)

Method \ Size	300	500	1K	10K	100K	1M	2M
MITRE-EP	2.74	4.39	8.72	97.58	113.43	1662.17	>7200
MITRE-INLA	1.99	8.38	32.65	133.54	>7200	>7200	>7200
KNN-CT	0.01	0.02	0.07	5.80	313.28	>7200	>7200
LOF-CECT	0.01	0.03	0.11	29.31	N/A	N/A	N/A
OCS-PCT	0.02	0.03	0.12	12.11	N/A	N/A	N/A
RELOADED	0.01	0.14	0.19	0.44	4.48	87.90	258.77
LOADED	0.07	0.10	0.22	2.54	23.59	249.13	484.02
OCS-RBF	0.02	0.03	0.07	8.38	606.84	>7200	>7200
FB-LOF	0.05	0.13	0.29	14.66	708.66	>7200	>7200
ODMAD	0.01	0.01	0.03	0.24	3.39	46.80	169.50

Experiments that exceeded the available memory resources are denoted by N/A

Experiments that ran over 2 hours are considered as failure

TABLE 4: Time Cost Comparison in terms of Size of P (seconds)

Method \ Size	10	25	50	100	200	300
MITRE-EP	8.77	19.06	153.30	1020.43	9344.83	>7200
MITRE-INLA	42.05	319.37	>7200	>7200	>7200	>7200
KNN-CT	0.14	158.24	>7200	>7200	>7200	>7200
LOF-CECT	6.69	596.32	>7200	>7200	>7200	>7200
OCS-PCT	0.24	>7200	>7200	>7200	>7200	>7200
RELOADED	0.24	0.46	1.02	2.26	5.44	7.86
LOADED	1.16	60.83	>7200	>7200	>7200	>7200
OCS-RBF	0.01	0.01	0.01	0.01	0.02	0.02
FB-LOF	0.30	0.37	0.51	0.87	1.63	2.11
ODMAD	0.018	>7200	>7200	>7200	>7200	>7200

Experiments that ran over 2 hours are considered as failure

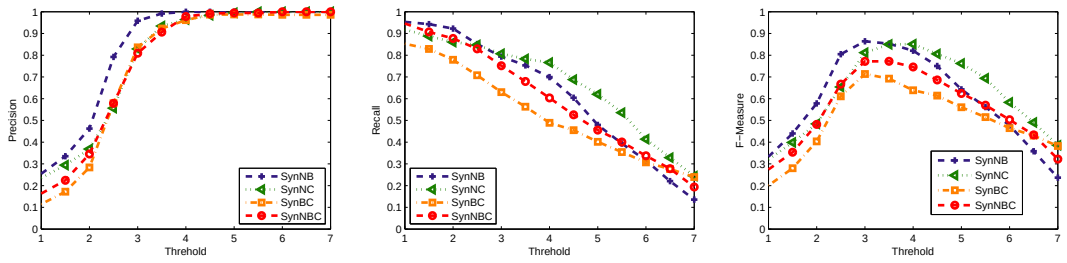
5.2.4 Impact of Parameters

Two major user input parameters in our new framework design are the threshold for determining anomalies, and the degree of freedom of the Student-t prior. The choice of these two parameters does affect the performance. We conducted this set of experiments using the synthetic datasets described previously, namely SynNB, SynNC, SynBC, SynNBC, for various sizes of instances. **Threshold of Anomalies:** This set of experiments analyzed the impact of the threshold used to determine anomalies. We used SynNB, SynNC, SynBC, SynNBC as described previously, with N equals to 100, 300, 500, and 1000. For each type-size combination, 10 variant realizations were generated. The threshold was tested over the range from 1 to 7 at 0.5 increments. Fig. 2 compares the effect of the different thresholds for the average precision, recall, and F-measure. Fig. 2 (a) shows that all the datasets follow the same

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

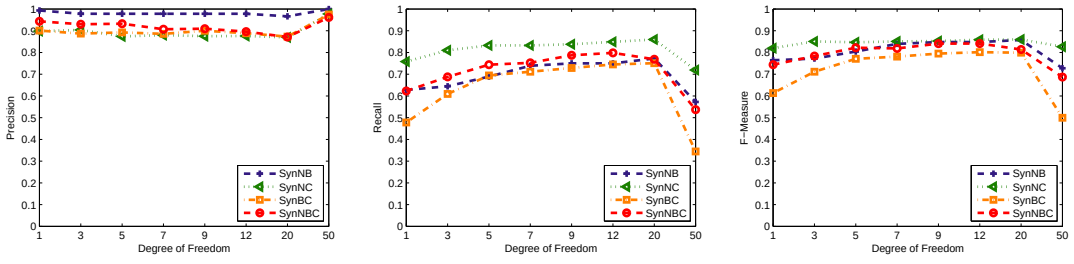
TABLE 2: Detection Rate Comparison among Synthetic Datasets (Precision, Recall)

Dataset	MITRE-EP	MITRE-INLA	KNN-CT	LOF-CECT	OCS-PCT	RELOADED	LOADED	OCS-RBF	FB-LOF	ODMAD
SynNB	**1.00**, 0.69	**1.00, 0.89**	0.11, 0.11	0.25, 0.50	0.29, 0.56	0.29, 0.56	0.28, 0.56	0.72, 0.72	0.06, 0.06	0.08, 0.61
SynNC	0.89, 0.82	**1.00, 0.89**	0.06, 0.06	0.40, 0.33	0.28, 0.56	0.29, 0.56	0.27, 0.56	0.72, 0.72	0.33, 0.33	0.06, 0.50
SynBC	**0.89, 0.67**	0.71, **0.67**	0.06, 0.06	0.33, 0.17	0.20, 0.39	0.20, 0.39	0.03, 0.06	0.33, 0.33	0.11, 0.11	0.08, 0.50
SynNBC	**0.92**, 0.73	0.80, **0.77**	0.04, 0.04	0.75, 0.33	0.33, 0.63	0.32, 0.59	0.21, 0.41	0.59, 0.59	0.04, 0.04	0.12, 0.58



(a) Precision (b) Recall (c) F-measure

Fig. 2: Impact of Error Threshold



(a) Precision (b) Recall (c) F-measure

Fig. 3: Impact of Degree of Freedom

pattern, with precision increasing significantly from 1 to 3.5, and then becoming moderate after 3.5. When the threshold equaled to 5, all of the datasets reached their maximum precision. Fig. 2 (b) shows that for all data types, the recall generally declined gradually. In Fig. 2 (c), the F-measure demonstrates a more obvious pattern. Here, for all data types, the peaks fall between the thresholds of 3 to 4, which confirms our hypothesis regarding the setting of the threshold. **Degree of Freedom:** This set of experiments analyzed the impact of degree of freedom df in the proposed model. We used the same synthetic data sets as in the previous set of experiments. When testing the impact of degree of freedom, the error threshold was fixed at 3. Fig. 3 shows that setting a lower df generally delivers higher precision, because the absorbed error highlights the difference between abnormal instances and normal instances. A slight increase in the F-measure from df=1 to df=3 has a visible effect; the inference was not able to converge to the global optimum in a limited number of iterations with the df set at less than 3.

5.3 Real-life Data Study

5.3.1 Data sets

We validated our approach using 14 real datasets, all of which can be found in the UCI machine learning repository [43]. Table 5 shows detailed information on these datasets. In the table, the types are denoted by N, B, and C for numerical, binary, and categorical, respectively. The response fields shown in Table 5 we used as Y and the remaining attributes as X in our experiment; the number refers to the n-th column of the raw dataset.

TABLE 5: Information in Real Datasets

Dataset	*Instances*	*Attrs*	*Type*	*Response*
Abalone	4177	9	C, N	1, 9
Yeast	1324	9	C, N	1, 6
WineQuality	4898	12	C, N	1, 12
Heart	163	11	C, B	3, 6
Autompg	398	8	C, N	1, 8
Wine	178	13	C, N	1, 2
ILPD	583	10	B, N	1, 2
Blood	748	5	B, N	4, 5
Concrete	103	10	B, N	8, 9, 10
Parkinsons	197	23	B, N	2, 18
Pima	768	8	B, N	3, 9
KEGG	53414	23	B, N	5, 7, 12, 13
MagicGamma	19020	11	B, N	1, 2, 11
Census	299285	42	C, N	6, 19, 25, 42

5.3.2 Anomaly Labels

Because the above datasets do not provide true anomaly labels, we preprocessed the data to obtain true anomaly labels in two different ways:

1. Rare Classes. For the first group of datasets (Abalone, Yeast, WineQuality, Heart and Autompg), we identified several rare categorical classes in the datasets. By following the same strategy as those used by existing anomaly detection studies [44], [45], these rare class instances were defined as true anomalies.

2. Random Shifting. For the remainig datasets, we regarded all the data objects as normal objects and followed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2583429, IEEE Transactions on Knowledge and Data Engineering

TABLE 6: Detection Rate Comparison among Real Datasets (Precision, Recall, F-measure, AUC)

Dataset	MITRE-EP	MITRE-INLA	KNN-CT	LOF-CECT	OCS-PCT
Abalone	**0.78**, 0.29, **0.42**, 0.94	0.25, **0.62**, 0.36, 0.98	0.16, 0.33, 0.22, 0.69	0.02, 0.04, 0.03, 0.49	0.20, 0.42, 0.27, 0.67
Yeast	**1.00**, 0.47, **0.64**, **1.00**	0.55, 0.67, 0.60, 0.59	0.29, 0.57, 0.38, 0.28	0.05, 0.10, 0.07, 0.15	0.21, 0.44, 0.28, 0.62
WineQuality	**0.50**, 0.29, 0.36, 0.93	0.33, 0.65, **0.44**, 0.95	0.03, 0.06, 0.04, 0.03	0.02, 0.04, 0.03, 0.03	0.04, 0.07, 0.05, 0.09
Heart	**1.00**, 0.57, 0.72, **0.99**	0.95, 0.75, 0.84, 0.98	0.46, **0.76**, 0.57, 0.50	0.45, 0.75, 0.56, 0.50	0.24, 0.43, 0.31, 0.50
Autompg	**0.47**, **1.00**, **0.64**, **1.00**	**0.47**, **1.00**, **0.64**, 0.99	0.00, 0.00, 0.00, 0.00	0.00, 0.00, 0.00, 0.00	**0.47**, **1.00**, **0.64**, 0.99
Wine	0.22, 0.66, **0.33**, 0.67	**0.33**, 0.30, 0.31, 0.63	0.09, 0.17, 0.12, 0.50	0.09, 0.17, 0.12, 0.50	0.09, 0.18, 0.12, 0.51
ILPD	0.22, **0.70**, 0.33, **0.78**	**0.84**, 0.18, 0.30, 0.77	0.26, 0.49, **0.34**, 0.60	0.12, 0.23, 0.16, 0.57	0.25, 0.49, 0.33, 0.59
Blood	**0.70**, 0.35, **0.47**, 0.74	0.56, 0.15, 0.24, **0.82**	0.23, 0.44, 0.30, 0.37	0.08, 0.15, 0.10, 0.35	0.24, 0.48, 0.32, 0.57
Concrete	0.57, **0.84**, **0.68**, **0.95**	**0.79**, 0.59, **0.68**, 0.92	0.07, 0.13, 0.09, 0.51	0.07, 0.14, 0.09, 0.50	0.09, 0.40, 0.15, 0.52
Parkinsons	0.60, **0.74**, **0.67**, **0.94**	**0.78**, 0.46, 0.58, 0.91	0.21, 0.42, 0.28, 0.37	0.23, 0.44, 0.30, 0.38	0.21, 0.41, 0.28, 0.50
Pima	0.79, **0.55**, **0.65**, 0.78	**0.83**, 0.27, 0.40, **0.82**	0.25, 0.48, 0.33, 0.44	0.06, 0.11, 0.08, 0.40	0.25, 0.49, 0.33, 0.66
KEGG	0.87, **0.65**, **0.77**, 0.75	0.59, 0.41, 0.48, 0.53	0.24, 0.46, 0.31, 0.37	N/A	N/A
MagicGamma	**0.67**, **0.66**, **0.66**, **0.83**	0.60, 0.55, 0.57, 0.82	0.14, 0.28, 0.19, 0.45	N/A	N/A
Census	**0.60**, **0.71**, **0.65**, **0.81**	0.51, 0.58, 0.54, 0.71	N/A	N/A	N/A

Dataset	RELOADED	LOADED	OCS-RBF	FB-LOF	ODMAD
Abalone	0.00, 0.00, 0.00, 0.29	0.00, 0.00, 0.00, 0.50	0.25, 0.25, 0.25, **0.99**	0.04, 0.04, 0.04, 0.74	0.01, 0.62, 0.02, 0.58
Yeast	0.00, 0.00, 0.00, 0.35	0.66, 0.66, 0.66, 0.58	0.63, 0.63, 0.63, 0.96	0.21, 0.21, 0.21, 0.50	0.05, **0.88**, 0.09, 0.91
WineQuality	0.00, 0.00, 0.00, 0.43	0.12, 0.12, 0.12, 0.51	0.11, 0.11, 0.11, 0.81	0.19, 0.19, 0.19, 0.75	0.12, **1.00**, 0.21, 0.91
Heart	0.51, 0.51, 0.51, 0.89	1.00, 0.16, 0.28, 0.72	0.65, 0.65, 0.65, 0.89	0.35, 0.35, 0.35, 0.57	0.99, **0.99**, **0.99**, **0.99**
Autompg	0.29, 0.29, 0.29, 0.70	0.33, 0.57, 0.42, 0.74	0.57, 0.57, 0.57, 0.98	0.10, 0.10, 0.10, 0.85	0.04, **1.00**, 0.08, 0.99
Wine	0.17, 0.36, 0.23, 0.59	0.12, 0.12, 0.12, 0.50	0.24, 0.24, 0.24, **0.77**	0.16, 0.16, 0.16, 0.60	0.10, **0.70**, 0.17, 0.56
ILPD	0.00, 0.00, 0.00, 0.50	0.09, 0.09, 0.09, 0.50	0.23, 0.23, 0.23, 0.68	0.09, 0.09, 0.09, 0.50	0.14, 0.71, 0.23, 0.45
Blood	0.03, 0.01, 0.02, 0.51	0.09, 0.09, 0.09, 0.50	0.39, 0.39, 0.39, 0.79	0.14, 0.14, 0.14, 0.58	0.19, **0.52**, 0.28, 0.64
Concrete	0.13, 0.26, 0.17, 0.58	0.08, 0.08, 0.08, 0.50	0.32, 0.32, 0.32, 0.72	0.17, 0.17, 0.17, 0.59	0.08, 0.43, 0.13, 0.49
Parkinsons	0.29, 0.21, 0.24, 0.59	0.07, 0.07, 0.07, 0.50	0.14, 0.14, 0.14, 0.72	0.21, 0.21, 0.21, 0.60	0.18, 0.53, 0.27, 0.65
Pima	0.10, 0.28, 0.15, 0.59	0.05, 0.05, 0.05, 0.50	0.52, 0.52, 0.52, 0.78	0.07, 0.07, 0.07, 0.52	0.31, 0.28, 0.29, 0.51
KEGG	0.78, 0.26, 0.39, 0.61	0.10, 0.10, 0.10, 0.50	0.51, 0.51, 0.51, **0.95**	N/A	**0.98**, 0.26, 0.41, 0.63
MagicGamma	0.28, 0.02, 0.04, 0.56	0.10, 0.10, 0.10, 0.50	0.29, 0.29, 0.29, 0.81	N/A	0.12, 0.46, 0.19, 0.55
Census	0.27, 0.30, 0.28, 0.64	0.10, 0.10, 0.10, 0.50	N/A	N/A	0.33, 0.29, 0.31, 0.61

Experiments that the methods failed to process are denoted by N/A

TABLE 7: Performance (AUC) Comparison for Various Random Shift Values

	MITRE-EP					MITRE-INLA				
Dataset \ Shift	1.0	1.5	2.0	2.5	3.0	1.0	1.5	2.0	2.5	3.0
Wine	0.59	0.58	0.63	0.67	0.67	0.62	0.61	0.59	0.63	0.64
ILPD	0.66	0.69	0.75	0.78	0.80	0.66	0.68	0.71	0.77	0.75
Blood	0.68	0.77	0.75	0.74	0.87	0.69	0.77	0.80	0.82	0.84
Concrete	0.74	0.84	0.93	0.95	0.97	0.81	0.82	0.93	0.92	0.97
Parkinsons	0.79	0.87	0.91	0.94	0.94	0.79	0.89	0.88	0.91	0.91
Pima	0.71	0.79	0.82	0.78	0.89	0.67	0.72	0.73	0.82	0.78
KEGG	0.60	0.59	0.67	0.75	0.74	0.54	0.55	0.53	0.53	0.68
MagicGamma	0.76	0.77	0.80	0.83	0.84	0.73	0.74	0.80	0.82	0.82
Census	0.70	0.71	0.74	0.81	0.79	0.69	0.68	0.72	0.71	0.76

the standard contamination procedure described in [11] and [13] to generate anomalies. We randomly selected 10% instances, and shifted the values on random fields. For the numerical attributes, we shifted the numerical values by 2.5 standard deviations and for the binary and categorical attributes, we switched the binary values to alternative values. The data for each dataset were preprocessed with 20 different artificial anomaly combinations and the average of the 20 results were calculated for each test.

5.3.3 Detection Accuracy

The main purpose of these experiments on real datasets was to validate our proposed anomaly detection method. Table 6 compares the metrics for precision, recall, F-measure, and Area Under Curve (AUC) for a number of different approaches. The results show that MITRE outperformed the benchmark approaches in terms of average precision and recall, which means that in most cases, the instances identified as anomalies by MITRE were true positives. MITRE also achieved the highest average AUC, signifying that our anomalous score measure always delivered the highest detection rate. Although several other benchmark approaches also achieved a high AUC, they also suffered from a high false positive rate or high false negative rate. For example, ODMAD achieved an AUC of over 0.9 on the *Yeast*, *WineQuality*, and *Auto-mpg* datasets, with nearly perfect recalls, but its performance in precisions did not exceed 0.12 because the estimated threshold set for anomalous scores was too low, so many normal instances were mistakenly labeled as anomalies. RELOADED, LOADED, and ODMAD required several parameters to be input as a set of hard thresholds, which significantly affected the performance of these methods. These approaches have the capacity to perform well after some parameter tuning process if the ground truth is known, but they will likely fail on many practical scenarios when the scale and the basis are unknown. In contrast, the proposed new method, MITRE, utilized the absolute value of the Z-score as the anomalous score with a statistical cutoff threshold under the Gaussian assumption, which is widely applied in many real world cases. Regardless of the scale of the different data attributes, this score represents the statistical significance and indicates to what extent it deviates from the normal behavior in the normalized basis.

The results also demonstrate the effectiveness of MITRE on large real-world datasets such as *Census*. The sub-

sampling fitting scheme (discussed in Section 4.2.3) effectively reduced the computational cost, while at the same time maintaining a good detection rate. In contrast, due to computation storage and time limitations, the benchmark methods LOF-CECT and OCS-PCT failed to process any of the datasets containing large number of instances (*KEGG*, *MagicGamma*, and *Census*) as they exceeded the available memory resources; KNN-CT, OCS-RBF, and FB-LOF also had problems with these large datasets as their running times were over 2 hours.

The impact of the outlier significance in the random shift data sets is shown in Table 7. We compared the AUC of random shifting significance levels ranging from one standard deviation to 3 times the standard deviation. Generally, shifted values of 1.5 the times standard deviation or less were difficult to be detect, although in some cases, our methods still performed well even when the anomalies were not significantly shifted. Based on our observations of datasets consisting of mixed-type data, and the shifting level only made a difference for numerical attributes, the binary and categorical anomalies were both detected accurately and the anomalous scores of these instances presented the correct ranking.

5.4 Result Analysis

The above experimental results demonstrate that MITRE-EP is an effective and efficient method for detecting anomalies in mixed-type data sets. It has a significantly better detection quality than the other benchmark approaches tested, achieving around 10-30% improvement over KNN-CT, LOF-CECT, OCS-PCT, OCS-RBF, and ODMAD and 20-40% over LOADED, RELOADED, and FB-LOF. The experimental results verified three main observations.

1) Efficient Approximation Process: The proposed approximate inference schemes provide faster and more accurate detection results. Compared with the INLA based method, MITRE-EP has better computational efficiency and higher detection accuracy on more of the real-life data sets.

2) Effectiveness on Large Mixed-type Datasets: When processing more sophisticated data sets, such as *Census*, *KEGG*, and *MagicGamma*, LOF-CECT and OCS-PCT failed to complete the process due to the significant growth of their memory usage. KNN-CT, OCS-RBF, and FB-LOF failed on the *Census* dataset due to their high time complexity. Our proposed methods were able to finish the process in a comparable time without any capacity problems.

3) Input-Output Relationship: When the datasets present strong input-output relationships for the explanatory attributes to the response variables, the MITRE methods deliver a much better performance on detection accuracy than the benchmark methods. Note that in making these comparisons, we followed the relationships suggested by the dataset providers for most of the real-life datasets.

6 Conclusions

This paper proposes a novel unsupervised framework for general purpose anomaly detection on mixed-type data. The new method integrates multivariate predictive process models with approximate Bayesian inference using Expectation Propagation and variational Expectation-Maximization. The predictive model consists of generalized linear models and robust error buffering latent variables. The approximation process and the optimization schemes provide more accurate and faster inference for the proposed predictive process model. Experimental results on synthetic and real datasets conclusively demonstrated that our proposed anomaly detection framework achieved much better performance on detection accuracy.

References

[1] V. Kumar, "Parallel and distributed computing for cybersecurity," *IEEE Distributed Systems Online*, vol. 6, no. 10, p. 1, 2005.

[2] C. Spence, L. Parra, and P. Sajda, "Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model," in *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA'01)*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 3–10.

[3] E. Aleskerov, B. Freisleben, and B. Rao, "Cardwatch: a neural network based database mining system for credit card fraud detection," in *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, Mar 1997, pp. 220–226.

[4] T. Brotherton and T. Johnson, "Anomaly detection for advanced military aircraft using neural networks," in *Aerospace Conference, 2001, IEEE Proceedings.*, vol. 6, 2001, pp. 3113–3123.

[5] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *The VLDB Journal*, vol. 8, no. 3–4, pp. 237–253, 2000.

[6] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *SIGMOD Record*, vol. 29, no. 2, pp. 427–438, May 2000.

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," *SIGMOD Record*, vol. 29, no. 2, pp. 93–104, May 2000.

[8] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos, "Loci: fast outlier detection using the local correlation integral," in *Data Engineering, 2003. Proceedings. 19th International Conference on*, March 2003, pp. 315–326.

[9] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study," in *KDD 10': 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 47–56.

[10] V. Roth, "Outlier detection with one-class kernel fisher discriminants," in *Advances in Neural Information Processing Systems 17*, 2005, pp. 1169–1176.

[11] M. Riani, A. C. Atkinson, and A. Cerioli, "Finding an unknown number of multivariate outliers," *Journal of the Royal Stats Society Series B*, vol. 71, no. 2, pp. 447–466, 2009.

[12] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," *Knowledge Information Systems*, 2011.

[13] A. Cerioli, "Multivariate outlier detection with high-breakdown estimators," *Journal of the American Statistical Association*, vol. 105, no. 489, pp. 147–156, 2009.

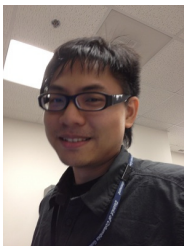
[14] G. Piatetsky-Shapiro, C. Djeraba, L. Getoor, R. Grossman, R. Feldman, and M. Zaki, "What are the grand challenges for data mining?: Kdd-2006 panel report," *SIGKDD Explor. Newsl.*, vol. 8, no. 2, pp. 70–77, 2006.

[15] Q. Yang and X. Wu, "10 challenging problems in data mining research." *International Journal of Information Technology and Decision Making*, vol. 5, no. 4, pp. 597–604, 2006.

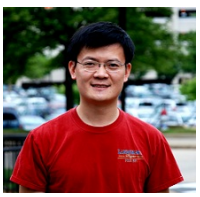
[16] A. Ghoting, M. E. Otey, S. Parthasarathy, and T. Ohio, "Loaded: Link-based outlier and anomaly detection in evolving data sets," in *Proceedings of the 4th IEEE International Conference of Data Mining*, 2004, pp. 387–390.

[17] M. E. Otey, S. Parthasarathy, and A. Ghoting, "Fast lightweight outlier detection in mixed-attribute data sets," *DMKD*, 2006.

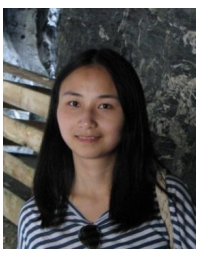
[18] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, ser. UAI '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 362–369.
[19] D. Yu, G. Sheikholeslami, and A. Zhang, "Findout: Finding outliers in very large datasets," Dept. of CSE SUNY Buffalo, Tech. Rep., 1999.
[20] D. E. Tyler, "Robust statistics: Theory and methods." *Journal of the American Statistical Association*, vol. 103, pp. 888–889, 2008.
[21] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," DTIC Document, Tech. Rep., 2003.
[22] C. Pascoal, M. R. de Oliveira, R. Valadas, P. Filzmoser, P. Salvador, and A. Pacheco, "Robust feature selection and robust pca for internet traffic anomaly detection," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1755–1763.
[23] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.
[24] D. Pyle, *Data preparation for data mining*. Morgan Kaufmann, 1999, vol. 1.
[25] T. Tran, D. Phung, and S. Venkatesh, "Mixed-variate restricted boltzmann machines," in *Proceedings of 3rd Asian Conference on Machine Learning (ACML)*, 2011.
[26] A. Koufakou, M. Georgiopoulos, and G. C. Anagnostopoulos, "Detecting outliers in high-dimensional datasets with mixed attributes." in *DMIN*, 2008, pp. 427–433.
[27] K.-N. Tran and H. Jin, "Detecting network anomalies in mixed-attribute data sets," in *IEEE Third International Conference on Knowledge Discovery and Data Mining (WKDD'10).*, 2010, pp. 383–386.
[28] A. Koufakou and M. Georgiopoulos, "A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes," *Data Mining and Knowledge Discovery*, vol. 20, no. 2, pp. 259–289, 2010.
[29] M. Ye, X. Li, and M. E. Orlowska, "Projected outlier detection in high-dimensional mixed-attributes data set," *Expert Systems with Applications*, vol. 36, no. 3, pp. 7104–7113, 2009.
[30] K. Zhang and H. Jin, "An effective pattern based outlier detection approach for mixed attribute data," in *AI 2010: Advances in Artificial Intelligence*. Springer, 2011, pp. 122–131.
[31] B. Warner and M. Misra, "Understanding neural networks as statistical tools," *The American Statistician*, vol. 50, no. 4, pp. 284–293.
[32] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York, NY, USA: John Wiley & Sons, Inc., 1987.
[33] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. Chapman and Hall/CRC, 2003.
[34] C. Liu, *Robit Regression: A Simple Robust Alternative to Logistic and Probit Regression*. John Wiley & Sons, Ltd, 2005, pp. 227–238.
[35] D. W. Hosmer and S. Lemeshow, *Applied logistic regression (Wiley Series in probability and statistics)*, 2nd ed. Wiley-Interscience Publication, 2000.
[36] H. Rue, S. Martino, and N. Chopin, "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations," *Journal of the Royal Statistical Society Series B*, vol. 71, no. 2, pp. 319–392, 2009.
[37] J. Gentle, *Solutions that Minimize Other Norms of the Residuals*. New York: Springer, 2007.
[38] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Section 7.9.1 Importance Sampling*. New York: Cambridge University Press, 2007.
[39] S. Martino and N. Chopin, "Implementing approximate Bayesian inference for latent Gaussian models using integrated nested Laplace approximations: A manual for the inla-program," Tech. Rep., 2008.
[40] R. E. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig, "The mutual information: Detecting and evaluating dependencies between variables." in *ECCB*, 2002, pp. 231–240.
[41] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ser. KDD '05. New York, NY, USA: ACM, 2005, pp. 157–166.
[42] K. Das and J. Schneider, "Detecting anomalous records in categorical datasets," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '07. New York, NY, USA: ACM, 2007, pp. 220–229. [Online]. Available: http://doi.acm.org/10.1145/1281192.1281219
[43] A. Frank and A. Asuncion, "UCI Machine Learning Repository," http://archive.ics.uci.edu/ml/, 2010.
[44] Z. He, X. Xu, J. Z. Huang, and S. Deng, "Fp-outlier: Frequent pattern based outlier detection." *Computational Science Information System*, vol. 2, no. 1, pp. 103–118, 2005.
[45] S. Wu and S. Wang, "Information-theoretic outlier detection for large-scale categorical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 589–602, 2013.



Yen-Cheng Lu Yen-Cheng Lu is a Ph.D. candidate in the Computer Science Department at Virginia Tech. He received his B.S. in Applied Mathematics from National Sun Yat-Sen University, Taiwan in 2008. His research interests are in the areas of statistical machine learning and data mining, especially outlier detection, spatio-temporal analysis, text mining, and transportation applications.



Feng Chen Feng Chen is an assistant professor in the Computer Science Department at SUNY Albany. He received his B.S. from Hunan University in 2001, M.S. degree from Beihang University, China in 2004, Ph.D degree from Virginia Tech, all in Computer Science. He has published 13 refereed articles in major data mining venues, including ACM-SIGKDD, ACM-CIKM, ACM-GIS, and IEEE-ICDM. He holds two U.S. patents on human activity analysis filed by IBM's T.J. Watson Research Center. His research interests are in the areas of statistical machine learning and data mining, with an emphasis on spatio-temporal analysis and energy disaggregation.



Yating Wang Yating Wang is a Ph.D. candidate in the Computer Science Department at Virginia Tech. He received her B.B.A. in Information Management and Systems in Hubei University of Technology, China in 2007. Her research interests are in the areas of statistical machine Learning in Mobile Ad Hoc Networks, particularly trust management in mobile networks.

Chang-Tien Lu Chang-Tien Lu received his MS degree in computer science from the Georgia Institute of Technology in 1996 and his PhD degree in computer science from the University of Minnesota in 2001. He is an associate professor in the Department of Computer Science, Virginia Tech. He served as General Co-Chair of the 20th IEEE International Conference on Tools with Artificial Intelligence in 2008 and 17th ACM International Conference on Advances in Geographic Information Systems in 2009. He is also serving as Vice Chair of the ACM Special Interest Group on Spatial Information (ACM SIGSPATIAL). His research interests include spatial databases, data mining, geographic information systems, and intelligent transportation systems.