Anomaly Detection in Categorical Datasets with Artificial Contrasts

by

Seyyedehnasim Mousavi

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2016 by the
Graduate Supervisory Committee:

George Runger, Chair

ARIZONA STATE UNIVERSITY

December 2016

ABSTRACT

Anomaly is a deviation from the normal behavior of the system and anomaly detection techniques try to identify unusual instances based on deviation from the normal data. In this work, I propose a machine-learning algorithm, referred to as Artificial Contrasts, for anomaly detection in categorical data in which neither the dimension, the specific attributes involved, nor the form of the pattern is known a priori. I use RandomForest (RF) technique as an effective learner for artificial contrast. RF is a powerful algorithm that can handle relations of attributes in high dimensional data and detect anomalies while providing probability estimates for risk decisions.

I apply the model to two simulated data sets and one real data set. The model was able to detect anomalies with a very high accuracy. Finally, by comparing the proposed model with other models in the literature, I demonstrate superior performance of the proposed model.

I would like to dedicate my thesis to my beloved husband, Sina, who is a real source of support and inspiration in my life. This work is also dedicated to my parents and brother who always support me and love me unconditionally.

TABLE OF CONTENTS

Page

LIST OF TABLES v

LIST OF FIGURES vi

1 INTRODUCTION 1

1.1 What Is Anomaly/Outlier? 3

1.2 Fraud Detection 4

1.3 Credit Card Fraud Detection 4

1.4 Mobile Phone Fraud Detection 5

1.5 Insurance Claim Fraud Detection 6

1.6 Insider Trading Detection 6

1.7 Industrial Failure Detection 7

2 LITERATURE REVIEW OF ANOMALY DETECTION METHODS 7

2.1 Statistical Model Based 8

2.2 Distance Based Model 9

2.3 Classification 10

2.4 Clustering 10

3 APPROPRIATE MODEL FOR ANOMALY DETECTION 13

3.1 Data 13

3.2 Types of Anomalies 14

3.2.1 Point Anomalies 15

3.2.2 Contextual Anomalies 15

Page

3.2.3 Collective Anomalies 17

3.3 Labels of Data 17

3.3.1 Supervised Techniques 18

3.3.2 Semi Supervised Techniques 20

3.3.3 Unsupervised Techniques 21

3.4 Output of Anomaly Detection 22

3.4.1 Scores... 22

3.4.2 Labels.. 22

4 ALGORITHM.......... 23

4.1 Artificial Contrast.......... 23

4.2 Random Forest 24

4.3 Methodology 26

4.3.1 Experiments 30

4.4 Method Comparison.......... 43

4.4.1 Data… 43

5 CONCLUSION.......... 50

6 REFERENCES 51

LIST OF TABLES

Table Page

1 Confusion Matrix of Low Dimensional Dataset 31

2 Confusion Matrix of Model on Dataset with 20 Variables 34

3 Confusion Matrix of Model on Dataset with 90 Variables 36

4 Average Statistics of Area under Roc Curve for 5 Replications of Model on Test Set for High Dimensional Dataset 38

5 Confusion Matrix of Model on Dataset with 20 Variables after Applying Feature Selection 40

6 Comparison of Methods 44

7 Average Statistics of Area under Roc Curve for 5 Replications of Model on Test Set with Different Malignant Records in Training Set 47

8 Average Statistics of Area under Roc Curve for 5 Replications of Model on Test Set with Different Benign Records in Training Set 48

9 Packages and Codes 49

LIST OF FIGURES

Figure Page

1. Distance-Based Method (It Is Hard for Distance-Based Method to Find Local Outlier O2) (Adapted from Chandola et al. [2]) 9

2. Point Anomalies (Adaptted from Chandola et al. [2]) 15

3. Collective Anomalies (Adaptted from Chandola et al. [2]) 17

4. Example of Roc Curve 28

5. Steps of the Algorithm 30

6. Roc Curve of Applying the Model on the Low Dimensional Dataset (The Area under the Curve = 0.86) 32

7. Histogram for Normal Class Probability Plot in $D0$ (The Model Recognizes the Patterned Records with High Accuracy) 32

8. Histogram for Normal Class Probability Plot in $D1$ (The Model Recognizes the Random Generated Records with High Accuracy) 33

9. Roc Curve of Applying the Model on the Dataset with 20 Variables (The Area under the Curve = 0.9) 34

10. Histogram for Normal Class Probability Plot in $D0$ (The Model Detects the Patterned Dataset) 35

11. Histogram for Normal Class Probability Plot in $D1$ (The Model Detects the Random Generated Data Records) 35

12. Roc Curve of Applying the Model on High Dimensional Dataset (The Area under the Curve = 1) 36

Figure Page

13. Histogram for Normal Class Probability Plot in $D0$ (The Model Detects the Random Generated Data Records) 37

14. Histogram for Normal Class Probability Plot in $D1$ (The Model Detects the Random Generated Data Records) 37

15. Roc Curve for 5 Replications of Model for High Dimensional Simulated Dataset.... 38

16. Histogram for Normal Class Probability Plot in $D0$ (The Accuracy of Model Improved after Applying Feature Selection on the Data Set with 20 Variables) 41

17. Histogram for Normal Class Probability Plot in $D1$ (The Accuracy of Model Improved after Applying Feature Selection on the Data Set with 20 Variables) 41

18. Histogram for Normal Class Probability Plot in $D0$ (All Data Are Randomly Generated) 42

19. Histogram for Normal Class Probability Plot in $D1$ (All Data Are Randomly Generated) Method Comparison... 42

20. Roc Curve of Test Set with 0 Malignant Records in the Training Set (Area under the Curve = 0.99) 45

21. Roc Curve of Test Set for 5 Replications of Model with 0 Malignant Records in the Training Set (All of the Curves Are Almost Identical).. 45

22. Roc Curve of Test Set with 10 Malignant Records in the Training Set (Area under the Curve = 0.98) 46

23. Roc Curve of Test Set with 20 Malignant Records in the Training Set (Area under the Curve = 0.97) 46

Figure Page

24. Roc Curve of Test Set with 40 Malignant Records in the Training Set (Area under the Curve = 0.95) 46

25. Roc Curve of Test Set with 100 Benign Records in the Training Set (Area under the Curve = 0.90) 48

26. Roc Curve of Test Set with 50 Benign Records in the Training Set (Area under the Curve = 0.87) 48

1 INTRODUCTION

Anomalies are patterns in data that do not conform to the expected normal data behavior. There are three categories of anomalies: point anomalies, contextual anomalies and collective anomalies. In this thesis, I consider only point anomalies. Point anomalies are individual instances or a small group of instances whose behavior are significantly different from the normal behavior of the rest of the data. Anomalous points negatively affect the result of analysis. Therefore, it is important and critical to identify those points. In addition, identifying anomalous points helps the analyst further investigate the source of anomalies.

Anomaly detection has broad applications in different industries, such as finance for credit card and loan applications, fraud detection, intrusion detection, production line, marketing, supply chain management, etc. Many statistical and machine learning algorithms have been developed to detect anomalies in different types of data sets (e.g. numerical data set, mixed of numerical and categorical data sets and categorical data sets). Although there exist different supervised and semi-supervised methods that can detect anomalies, unsupervised learning techniques such as clustering have been used widely in anomaly detection because in real applications we rarely have data labels that classify normal and anomaly instances; and unlabeled data can only be handled by unsupervised techniques.

On the other hand, supervised methods have capabilities that could help us to build accurate models. For example, supervised techniques could capture attributes' relations, and by using feature selection methods, those methods could identify important variables and transform a high-dimensional data set to a low-dimensional one. Therefore, having a supervised method that could be applicable on unlabeled data would be a great opportunity in using supervised learning models’ advantages while unsupervised models are perfect for our data.

Many researches have used statistical methods for anomaly detection of numerical data. However, it is non-trivial to extend these techniques to the case of categorical feature values. It is common to find areas such as computer networks, security, financial transactions, marketing, supply chains, and so forth where data is primarily categorical or mixed categorical and numerical in nature. Thus, there is a need to develop an anomaly detection method that can handle categorical data.

In this study, I use Artificial Contrast (AC) method that essentially converts an unsupervised problem to a supervised one by contrasting the normal data with the anomalous data. The artificial contrast method with a suitable supervised learner can help detect patterns in the data, and distinguish normal data from anomalous data.

Feature selection techniques will then be employed to identify the features involved in the pattern. I can also produce a probability estimate of the “normalness” of the instance. This method was successfully employed by Hwang et al. [1] in converting the classical multivariate control chart monitoring problem into a supervised learning problem. The focus of the reference was numerical data. I extend the idea here to our case of categorical data.

The remainder of the thesis is organized as follows: in section 2 I review previous works on anomaly detection in categorical data set. In section 3 I investigate different types of anomalies and determine how we can choose an appropriate model based on the data set that we have. Section 4 is about my proposed algorithm and the experiment that I have done on two simulated datasets. I also apply the model on a real dataset and compare the model with three other models. Section 5 has conclusion and future works.

1.1 What Is Anomaly/Outlier?

Anomalies are patterns in data that do not follow the normal behavior of the data. A data set could have anomalies for different reasons such as malicious activity, credit card fraud, cyber-intrusion, terrorist activity or breakdown of a system [2]. Anomalies represent abnormal behavior of system, so it is important to figure out anomalies and the causes of the those points.

An anomaly can represent a problem in an image such as a land mine. It could be a signal that shows an intruder inside a system with malicious intensions. Anomalies in production systems could be faults in production lines that could be identified by monitoring specific features of products and comparing the results with the normal products data. In credit card application, anomalies could show an abnormal behavior of cardholder. Anomaly detection techniques could monitor customers' purchasing behavior and detect abnormal transactions. Anomalies may show stolen credit card or a real change in a customer purchasing habits. In addition, in phone monitoring, customers' phone usage is tracked and if a usage does not follow the customer's usage pattern, it will be considered as an anomaly. In some cases such as loan application and social security payments, anomaly detection techniques help to detect anomalies before application approval or doing any payments.

An unusual computer network traffic pattern may be showing that the computer has been hacked and is sending data to unauthorized users [3]. In MRI, the presence of tumor creates anomalous parts in the image that help specialists to recognize those tumors [4]. In credit card transactions, anomalous instances show a different usage of credit card that could be because the card or the credit card's information has been stolen [5].

In this section, I briefly describe some applications of anomaly detection:

1.2 Fraud Detection

Fraud refers to abnormal activities that occur in organizations such as banks, insurances, stock markets etc. Fraud detection is a technique that can detect these activities. Fraud could be done by an actual customer or by another person that has stolen customer's information and are working with the organization's products/services without consent in a shape of customer. It is very important for organizations to detect these activities in an early stage, because it prevents them from losing much money. I discuss more about fraud in different organizations.

1.3 Credit Card Fraud Detection

Credit card fraud is one of the most important fraud that has taken many considerations in financial institutions. Credit card fraud causes huge financial loses; in addition to costs associated with the thieves' spending, the costs of issuing new credit or debit card and sending those cards to customers are high too. Fraud losses in US during 2013 was about $7 billion dollars [6]. According to fraud statistics in creditcard.com, in 2014 in US, 90% of people who were victim of fraud request a new card that cost for the financial institution about $ 12.75 per card. Credit card fraud can occur in different ways: credit card could be stolen itself, or its information could be stolen during online transactions or by online hackers, and it could occur with counterfeit credit card. Statistics show that about 45% of credit card frauds occur in online transactions [7]. To find the frauds, banks and financial institutions should have an algorithm that could detect frauds at the time it occurs and learn the fraud patterns for future predictions. To understand more about how the methods should work, I prepare a brief explanation of credit card data and how a fraud transaction is different from the other ones.

Credit card usage data includes customer's purchases records that can be used for understanding purchase patterns. Data sets typically have features like user ID, spending amount, duration between two credit card usage, location of transaction, etc. If a strange transaction happens and that is beyond the scope of normal behavior of a specific customer then it is a signal of anomaly. Sometimes, the strange occurrence is not fraud and it is a real change in customers' behavior or a real unique transaction in customer's life.

For example, when a customer goes on vacation for a week, the location of purchasing would change. On vacation, people usually spend more money and on special or new activities such as dining at different restaurants, visiting museums, etc.

Also, sometimes customer's life changes and so does his/her spending style. These two examples are anomalies, but not a fraud and they are not costly for an organization.

A type of credit card anomaly that is costly for organizations and must be detected is a fraud in which somebody has stolen customers' information or credit card and is trying to make purchases using it.

1.4 Mobile Phone Fraud Detection

In phone fraud detection, calling behavior of a user is a reference and if a call occurs from a phone user that is not compatible with his/her behavior, it will be detected as an anomaly and it appears that the account has been misused.

Phone call data sets usually have features related to calls like calling location, calling duration, calling time, calling destination, etc. For example, anomaly could be a call to a destination that is not in a user's historical call records, or may be the volume of call in a specific duration of time is much higher than user's call behavior pattern.

1.5 Insurance Claim Fraud Detection

Another type of fraud that causes huge costs to companies is insurance frauds. To prevent the associated costs it is worth to have an algorithm that can learn from previous claims and predict future fraud, before any claim, with high accuracy. Insurance frauds usually occur when a claimant tries to get some advantages from insurance, the advantages that otherwise will not be given to them. For example, a person who had a car accident can claim something from insurance that is not real. He/she could file more than one claim for a specific injury, file a claim for injuries that are not related to the accident, report higher costs than real costs for car repairs and report wage losses for the accident while that wage losses are not real.

1.6 Insider Trading Detection

Another application of anomaly detection techniques is detecting insider trading in an early stage. Insider trading is a phenomenon found in stock markets, where people make illegal profits by acting on (or leaking) inside information before the information is made public. The inside information can be of different forms [8]. It could refer to the knowledge of a pending merger/acquisition, a terrorist attack affecting a particular industry, pending legislation affecting a particular industry, or any information that would affect the stock prices in a particular industry. Insider trading can be detected by identifying anomalous trading activities in the market.

The available data is from several heterogeneous sources such as option trading data, stock trading data, news. The data has temporal associations since the data is collected continuously. The temporal and streaming nature has also been exploited in certain techniques [9].

Anomaly detection techniques in this domain are required to detect fraud in an online manner and as early as possible, to prevent people/organizations from making illegal profits.

1.7 Industrial Failure Detection

Continuous usage and normal wear and tear could cause industrial units damage. To save money and prevent those damages, it is important for companies to have a system that records machines working behaviors and conditions. Anomaly detection techniques have been applied in this domain to detect the damages. Companies usually use sensors that can record machines operations. If a machine does not work properly or there is a problem in machines' units, the machine will deviate from its normal operation, so anomaly detection techniques could find those problems.

2 LITERATURE REVIEW OF ANOMALY DETECTION METHODS

Anomaly detection is an important issue that has been studied in a wide variety of areas and applications. There are different methods of anomaly detection. In this section I review different anomaly detection techniques that have been developed until now. Also, I specifically reviewed the techniques that can handle categorical data set anomalies.

2.1 Statistical Model Based

These methods assume the data instances come from a parametric distribution with parameters θ and probability density function f(x,θ) where x is an observation [2]. In these methods if a point is out of distribution's boundaries then it will be considered an anomaly or outlier. Example of parametric methods are Gaussian models and Regression models [10], [11].

Although these methods are efficient for most of single-dimensional and univariate data but it has some disadvantages too. One problem with these kind of models is finding an appropriate model for different data set and applications is hard. Another difficulty with these methods is these models are not efficient for high dimensional datasets [12], [13]. Furthermore, in high dimensional data set because the data set is become less dense, it is hard for parametric models to determine the convex hull [1].

For overcoming these problems, some methods have been developed. Principal Component Analysis is one of those methods that could be useful for high dimensional data. Also, Ida and Rousseeuw [14] proposed an algorithm that organizes data instances in layers. Those instances that are on the shallow layers tend more to be outliers that those points on the deep layers [15], [16]. However, these kind of methods still are not efficient for more than two dimensions data sets.

On the other hand, non-parametric methods make fewer assumptions for data distributions. Non-parametric methods do not make any assumption about the structure of the data and they get the structure from the given data set [2]. Examples of non-parametric models are: Histogram based models and Kernel function based models.

2.2 Distance Based Model

One of non-parametric methods is distance-based approaches. These approaches do not have any assumptions about data distributions. These methods only use distance between points. Although these kind of techniques do not make any assumption about the data distribution, but they are not appropriate for high dimensional data, because calculating the distance between every points in a data set leads to high computational complexity (e.g. nearest neighbor approach that has quadratic complexity with respect to the dataset size).

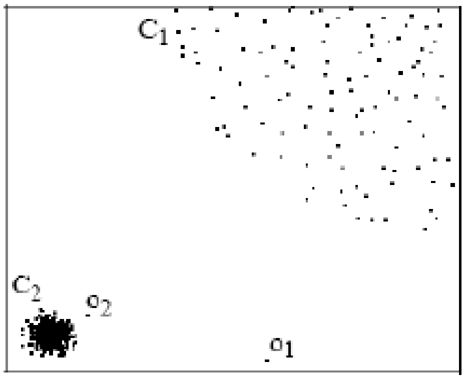

Fig. 1. Distance-Based Method (It Is Hard for Distance-Based Method to Find Local Outlier O2) (Adapted from Chandola et al. [2])

To overcome this problem, there have been improvements of the original distance-based algorithms. Kumar [17] proposed a distance-based algorithm that can detect outliers in high dimensional dataset. In their work, object O is an outlier if at least a fraction of p of all objects are further than a defined distance D from point O.

Finally, Bay and Schwabacher [18], use randomize data to make an efficient search space. Their model helps to have a near linear complexity even though the problem is complex quadratic

2.3 Classification

Classification is a technique that learn a model from a labeled training data and then classify test data based on the learned model [11]. The model that has been learned from training data could distinguish between normal and anomalous instances. Based on the number of classes belong to normal data, classification anomaly detection techniques are divided into two groups: multi-class and one class. In multi-class classification, it is assumed that training data has more than one class [12], [19]. An effective learner could distinguish between each normal classes. Moreover, if a point in a test set does not belong to none of these classes, then it will be considered as an anomalous point. Bayesian-Network is an example of multi-class classification technique. Support Vector Machine (SVM) have been used for anomaly detection as a one-class method. Neural-Network and Rule-Based algorithms work for both one-class and multi-class anomaly detection.

2.4 Clustering

Clustering is an unsupervised technique that is used to group similar data instances into clusters [11], [20]. Clustering-based techniques could have different assumption for similarity of data points but in general, these methods differentiate between groups of data that are similar to each other and those points that have different behavior which are typically called outliers.

Some clustering techniques are built based on this assumption that normal instances belong to a cluster while anomalies do not belong to any cluster [2]. Clustering techniques based on this assumption apply a clustering method on a data set and every points that are not in a cluster is considered as an outlier. Clustering techniques like BSCAN [21], ROCK [22], and SNN clustering [23] can be used.

Another assumption implies that normal data instances lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid [2]. Techniques that are based on this assumption finding anomalies in two steps. First, they cluster data points using a clustering algorithm and then they calculate the distance between each instance with the centroid of the closest cluster. This distance score is a measure of anomaly. There are some clustering algorithms that can detect anomalies based on these two steps. Self-Organizing Maps (SOM), K-means Clustering, and Expectation Maximization (EM) are the techniques that are working based on these two steps. Smith et al. [15] studied these methods for clustering training data and then using those defined clusters, they classified test data.

A disadvantage of this assumption is if anomalies are enough big that can build a cluster by themselves, the techniques which are based on this assumption cannot detect them as anomalies [2].

A third assumption is normal data points build large and dense clusters while anomalies build small and sparse clusters [2]. Actually, this assumption mitigates the problem that assumption 2 has. Techniques that follow this assumption usually have a threshold for density of normal clusters. If an instance falls below that threshold, that instance would be anomaly. He et al. [22] proposed an algorithm called FindCBLOF, which assigns an anomaly score to each data point based on the size of the cluster that point belongs to.

Most of the current algorithms are developed for numerical data set. Therefore, when using these methods for analyzing a data set that has mixture of numerical and categorical variables, we miss the information that categorical variables have and our analysis will not be accurate. In this section, I reviewed multiple algorithms that have been developed to find anomalies in categorical data sets. Ghoting et al. [24] proposed an algorithm called LOADED that can detect anomalies in data set that have both numerical and categorical variables. They consider the link between two categorical variables as a sign of their relationships and the number of common attributes and values between two variables as a strength of the links. A point is considered an outlier if they have less links. They assign a score to each point based on their links to other points. Based on that score, anomalous instances could be recognized.

He et al. [25] proposed an algorithm called Frequent Pattern (FP) outlier detection. The method detects the frequent patterns of a data set with association mining. For each transaction it computes a score called Frequent Pattern Outlier Factor (FOPF), if a transaction follow the pattern its FPOF is high, otherwise it will be low. The transactions that have low FOPF score are considered as outliers.

He et al. [26] proposed an algorithm that optimized the entropy of the data set. The algorithm recognized outliers iteratively and removing those points from data set while minimizing the entropy of the target data set. Another approach that can detect outliers in categorical data set is to represent the data as a hypergraph. Wei et al. [11] proposed hypergraph method in which they found frequent item sets and for each item set they used multidimensional array to detect outliers.

Another approach that is useful for anomaly detection in categorical data sets has been proposed by Das and Schneider [27]. In this approach, they used conditional probability ratio to detect unusual combination of attributes. This technique detect unusual combinations but not rare combinations that sometimes could be anomalies. To detect rare combinations they used marginal probability. They also showed that their method outperforms to Bayesian Network, another useful technique in anomaly detection.

3 APPROPRIATE MODEL FOR ANOMALY DETECTION

There are some criteria that we should consider when we are choosing an appropriate model for anomaly detection. Input data is a very basic criterion that we should consider in a first step of each anomaly detection. Considering whether the data set has label or not is very impactful on the technique we choose for analyzing our data. Also, every applications of anomaly detection has its own limitations and requirements that should be considered in choosing an appropriate technique. In this section, I discuss these criterions in details and see different aspects of anomaly detection.

3.1 Data

One of an important and basic factor in analyzing every data that affects the chosen model is data. Data is generally a collection of instances (also referred as object, record, point, vector, pattern, event, case, sample, observation, or entity) [12]. Each data instance is described with a set of attributes (also referred to as variable, characteristic, feature, field, or dimension). The attributes can have different types such as binary, categorical, or continuous [2]. Each data instance could be described with one attribute, so we will have a univariate model, or a data instance may need more than one attributes to be described, this case is called multivariate. In multivariate case, all attributes could be in a same type or they could have different attribute types.

The types of attributes is an important aspect for choosing an appropriate anomaly detection technique. For example, in statistical techniques, there are different models that could handle continuous and categorical variables. Based on the nature of data, we could select the best model. In addition, for anomaly detection with nearest neighbor techniques, the distance measure is determined based on the type of attributes that are involved in the model.

Data instances’ relations categorize input data [12]. Different techniques should be applied on different input data. Although most of the current anomaly detection techniques are considering point data that there is not any relationships among instances, but data instances could have relations with each other, for example sequence data, spatial data, and graph data.

Sequence data has linearly ordered data instances. Time-series data, genome sequences, and protein sequences are examples of sequence data. On the other hand, spatial data has related data instances. Each data instance in spatial data is related to its neighbors instances, for example, vehicular traffic data, and ecological data. Graph data consists of vertices which are data instances and those vertices are connected to each other with edges [2].

3.2 Types of Anomalies

After understanding the input data and the complexity of the problem, another important step is configuring the application and type of anomalies that application could have. Anomalies can be classified into following three categories: point anomalies, contextual anomalies and collective anomalies.

3.2.1 Point Anomalies

Point anomalies are an individual instance or some few instances that have different behavior from rest of the data. For example, in figure 2 below, o1, o2, and o3 are very far from the rest of the data, so they are considered as point anomalies.

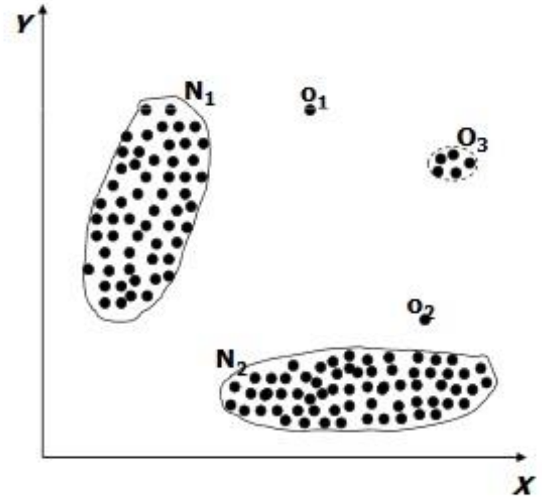

Fig. 2. Point Anomalies (Adaptted from Chandola et al. [2])

As a real example, consider a credit card fraud detection problem. We assume that we have a data set that contains the amount an individual person spend with his/her credit card. A transaction that has very high amount compared to his/her regular transactions is considered as point anomaly.

3.2.2 Contextual Anomalies

Contextual anomalies are those points that are considered as anomalous points within a specific context, and other than that context, the points may be not anomalous instances.

To detect contextual anomalies, the data set should have a specific construction. We should have two sets of attributes: contextual attributes and behavioral attributes.

Contextual attributes help to determine the neighborhood for a specific instance. We are supposing that an instance follows its neighbor's behavior, but if the behavior of that instance is different from neighbors, so it will be considered as anomaly.

Example of contextual attributes are longitude and latitude of locations in spatial data. In time-series data, the contextual attribute is time that shows the position of an instance in entire dataset [2].

Non-contextual characteristics of an instance is described with the behavioral attributes. For example, in a spatial data set describing the average rainfall of the entire world, the amount of rainfall at any location is a behavioral attribute [2].

Anomalies are those values from behavioral attributes that are strange in a specific context.

A data instance could have different behavior in different context; it could be anomaly in a specific context and could be normal in another context.

An example of contextual anomalies could be in credit card fraud detection. A contextual attributes could be the amount of purchase in a specific location. Suppose a customer usually spend $200 weekly for grocery in Sprouts market. If that person spend $1000 in a week in that specific store, that is an anomaly, since it does not follow the normal behavior of that customer in the context of location, even though that customer could spend $1000 on other purchasing in another store and that purchase is considered normal.

3.2.3 Collective Anomalies

Collective anomaly is a group of related instances that does not follow the behavior of the rest of the data set. Each of the individual instances may not be anomalies by themselves, but in a collection with other anomalies, all of them are considered as anomalous points. Figure 3 shows a collective anomalies in an electrocardiogram output [28]. The highlighted part shows anomalies because the low value exists for a long time. We should notice that the low value itself is not considered as anomaly, but when it occurs long time then it is abnormal

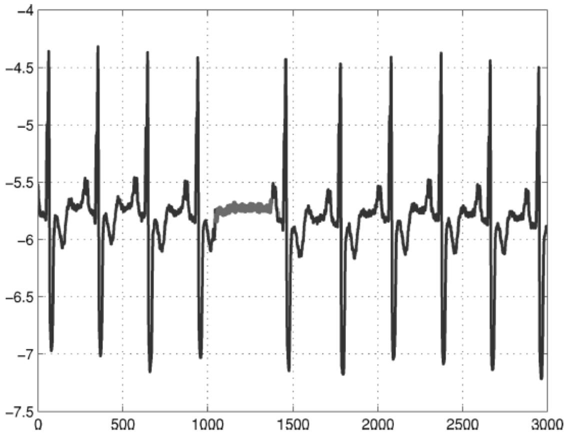

Fig. 3. Collective Anomalies (Adaptted from Chandola et al. [2])

3.3 Labels of Data

Data labels in a data set determine whether that instance is normal or anomalous. Since labeling is usually done by human experts, it is an expensive task. In addition, getting labels for normal data instances are easier than having labels for all types of anomalies that could occur in a data set. In addition, it is possible that during time different types of anomalies could occur that we do not have labels for them in the dataset. Based on the availability of labels for normal and anomalous points we have different analysis methods: supervised, semi-supervised and unsupervised.

3.3.1 Supervised Techniques

Supervised techniques need labels for both normal and anomalous instances of training data set. Typically, a predictive model is built for the data set to classify normal and anomalous instances. For any unseen data set, the predictive model is applied to that data and the result would be classified instances as normal or anomalies. Example of supervised techniques are decision tree, support vector machines, artificial neural networks, rule-based techniques, etc.

There are four issues with supervised techniques that should be considered:

Bias-variance tradeoff

A first issue is the tradeoff between bias and variance. Imagine that we have available several different, but equally good, training data sets. A learning algorithm is biased for a particular input x if, when trained on each of these data sets, it is systematically incorrect when predicting the correct output for x. A learning algorithm has high variance for a particular input x if it predicts different output values when trained on different training sets. The prediction error of a learned classifier is related to the sum of the bias and the variance of the learning algorithm. Generally, there is a tradeoff between bias and variance. A learning algorithm with low bias must be "flexible" so that it can fit the data well. But if the learning algorithm is too flexible, it will fit each training data set differently, and hence have high variance. A key aspect of many supervised learning methods is that they are able to adjust this tradeoff between bias and variance (either automatically or by providing a bias/variance parameter that the user can adjust).

Function complexity and amount of training data

The second issue is the amount of training data available relative to the complexity of the "true" function (classifier or regression function). If the true function is simple, then an "inflexible" learning algorithm with high bias and low variance will be able to learn it from a small amount of data. But if the true function is highly complex (e.g., because it involves complex interactions among many different input features and behaves differently in different parts of the input space), then the function will only be learnable from a very large amount of training data and using a "flexible" learning algorithm with low bias and high variance. Good learning algorithms therefore automatically adjust the bias/variance tradeoff based on the amount of data available and the apparent complexity of the function to be learned.

Dimensionality of the input space

A third issue is the dimensionality of the input space. If the input feature vectors have very high dimension, the learning problem can be difficult even if the true function only depends on a small number of those features. This is because the many "extra" dimensions can confuse the learning algorithm and cause it to have high variance. Hence, high input dimensionality typically requires tuning the classifier to have low variance and high bias. In practice, if the engineer can manually remove irrelevant features from the input data, this is likely to improve the accuracy of the learned function. In addition, there are many algorithms for feature selection that seek to identify the relevant features and discard the irrelevant ones. This is an instance of the more general strategy of dimensionality reduction, which seeks to map the input data into a lower-dimensional space prior to running the supervised learning algorithm.

Noise in the output values

A fourth issue is the degree of noise in the desired output values (the supervisory target variables). If the desired output values are often incorrect (because of human error or sensor errors), then the learning algorithm should not attempt to find a function that exactly matches the training examples. Attempting to fit the data too carefully leads to overfitting. You can overfit even when there are no measurement errors (stochastic noise) if the function you are trying to learn is too complex for your learning model. In such a situation that part of the target function that cannot be modeled "corrupts" your training data - this phenomenon has been called deterministic noise. When either type of noise is present, it is better to go with a higher bias, lower variance estimator.

In practice, there are several approaches to alleviate noise in the output values such as early stopping to prevent overfitting as well as detecting and removing the noisy training examples prior to training the supervised learning algorithm. There are several algorithms that identify noisy training examples and removing the suspected noisy training examples prior to training has decreased generalization error with statistical significance.

3.3.2 Semi Supervised Techniques

Semi-supervised techniques need labels just for normal instances and do not need that labeled anomalous instances.

Because they just need normal instances labels, they are more flexible than supervised methods, that need labels for both normal and anomalous data points, and are more widely applicable, specifically in applications that just have normal data and it is hard to have a pattern for anomalous instances. For example, anomaly in space craft is an accident that is not easy to model. The easy way is to have a normal data and build a model based on those normal instances. If a data instance in a test set does not follow the normal behavior, then it will be considered as anomaly.

3.3.3 Unsupervised Techniques

Unsupervised methods do not need any labeled data instances. These techniques will find the pattern in a data set without knowing which data instances are normal and which ones are anomalous. Based on that pattern the methods can separate the normal instances from anomalous instances. The assumption of these techniques is normal data instances are more frequent than anomalous ones. Clustering is an unsupervised technique that is widely used in literature and real applications. Clustering methods group objects in a way that objects that are in the same group (cluster) are more similar than objects in other groups. Hierarchical clustering, k-means algorithm and DBSCAN are examples of clustering methods. They use different measurements for clustering instances. For example, k-means algorithm uses distance as a measure for clustering. The method groups instances while minimizing the distance between points that are in a same cluster and maximizing the distance between points in different clusters.

3.4 Output of Anomaly Detection

One of important aspects of anomaly detection techniques is the output. It is important to know how each technique will produce the anomaly output. Typically, we have two kinds of outputs: scores and labels.

3.4.1 Scores

In these techniques, the output is a ranked list of anomalies. Scoring techniques assign a score to each instance of test data. The score shows how that instance is deviate from the normal pattern in training data. An analyst can analyze the instances that have higher anomaly scores or he/she can use a cut off point (based on the problem and application) to select anomalous instances.

3.4.2 Labels

Some techniques give a label (normal or anomalous) to each instances in a test set. In these techniques we cannot apply a threshold directly anymore, but we can make changes in parameter choices in each techniques.

4 ALGORITHM

In this section, I explain artificial contrast as a method that transforms unsupervised techniques to supervised ones. As discussed, artificial contrast needs a powerful learner, which can distinguish normal and anomaly instances with low error rate. In addition, it should give an accurate estimation of class probability.

4.1 Artificial Contrast

One way to convert an unsupervised problem into a supervised one is to transform a density estimation problem to function approximation and classification. (considered to be "statistical folklore") [29]. Let $f(x)$ denote the unknown probability density to be estimated and $f_0\ (x)$ be the reference distribution we want to contrast with.

Define Y to be the target value and assign $Y\ =\ 1$ and $Y\ =\ 0$ for each point drawn from $f(x)$ and $f_0\ (x)$, respectively. We draw an equal number of instances from each distribution. Applying Bayes' Theorem yields:

$$E(Y \mid x) = P(Y = 1|x) = (f(x))/(f(x) + f_0\ (x)) \qquad (1)$$

$E(Y \mid x)$ can be estimated by solving the regression problem on the sample (x_1, y_1), $(x_2, y_2), \ldots\ (x_{2n}, y_{2n})$. The calculation in Equation (1) can be easily adjusted to account for different sample sizes for the training and reference data sets or for different responses other than zero or one.

Thus, we obtain an estimate $f\hat{}(x)$ of our function$f(x)$. Essentially, the problem can now be considered as a two-class problem, and solved using an efficient supervised learner together with an appropriate reference distribution.

Hwang et al [1] applied artificial contrast to multivariate statistical process to detect normal operating conditions from abnormal conditions. Tuv [30] used artificial contrast with RF learner to select the best subset of non-redundant features in high dimensional dataset.

4.2 Random Forest

A good learner for our problem should be able to handle high-dimensional categorical data. It should also be able to capture the higher-order interactions that may be present between different categorical features. Finally, it should provide a class probability estimate. Therefore, I choose Random Forest (RF) here as the supervised classifier [31]. Ensemble models are prediction models that are combined of several simpler models. RF is an ensemble of tree predictors, each constructed from random sampled training data. The RF prediction is the unweighted plurality of class votes (or average prediction for regression). RF results could be biased towards variables with more categories. In addition, when two variables are correlated, RF randomly uses one of the variables first and so the importance of the other variable significantly decreases because the impurity that this variable could remove is already removed. To have an accurate RF results, the model should have low bias and low correlations between variables. To keep bias low, trees are grown to maximum depth. To keep the correlation low, each tree is grown on a bootstrap sample of the training set, and only m attributes are randomly sampled (out of the total of p attributes) at each node in each tree. The best split (as measured by class entropy gain) is selected among the m candidates. Here m is a tunable parameter, generally kept at a default of $p^{1/2}$(6). The number of trees used is another tunable parameter (but results are insensitive to the selected value). Each bagged trees uses about two-thirds of instances and the remaining one-third observations are called out-of-bag (OOB) observations.

We can predict the response of an instance x_i using each of the trees in which xi is OOB. Consequently, we will have multiple predictions for xi. In order to have a unique prediction for that instance, we use the majority votes of all predicted responses for classification purpose and average of the responses for regression. We can have an OOB prediction for each instance of a data set in the same way and from that we can have an overall model error called OOB error. OOB error is a valid estimate of test error because the response of each observation is predicted using only the trees that were not fit using that observation.

Let T denote the number of trees in the RF. Let $Tj(xi)$ denote the prediction for xi from tree j for j = 1, 2, …, T.

Let OOB_idenote the set of trees with xi OOB. The class probability estimate for instance xi and class k is:

$$\hat{p}_k(x_i) = \frac{\sum_{j \in OOB_i}(T_j(x_i)=k)}{|OOB_i|} \tag{2}$$

The OOB predicted class of instance xi is $\hat{Y}(x_i)$=$argmax_k\ \hat{p}_k(x_i)$. For two classes $\hat{Y}(x_i)$=0 if $\hat{p}_0\ (x_i\ )$>c for a selected constant threshold c. The error rates computed from OOB predictions are easily obtained and have been shown to be good estimates of generalization error (6). Consequently, OOB error rates and posterior probability estimates are used in this work. Finally, RF works well with categorical data sets and can also be used to the mixed case of categorical and numerical data.

4.3 Methodology

In this section, I explain the methodology and show how I prepare data set to be analyzed with the proposed model. For this study, I should have two unlabeled data sets. One original data set that have the data that I want to analyze. I call it D_0 here. Another data set that I need is artificial data set that represents the anomalous instances. I call it D_1. By contrasting these two data sets, we could identify anomalies.

Dataset D_0 consists of N_0 instances over p features. Let the number of values for feature xj be denoted as v_j. Usually we assume that the training data is free of anomalies, but in this study my proposed learning algorithm (RF) is robust enough to tolerate a small percentage of anomalous instances. The data are unlabeled, but because I am using a supervised algorithm, I should have a label for each instances of the data set. I artificially assign the label Y = 0 to the instances in D_0. In real applications, usually attributes are not independent from each other and have relationship that build a pattern. And this constructed pattern helps us to identify anomalous instances. Therefore, I expect relationships between the different features in D_0. Consider the following simple example. Suppose John usually walks to the gym on Monday. This scenario can be considered to define a pattern in the categorical attributes {person, travel type, destination, day of week}. If John drives to the gym on Monday we would like to flag this as an anomaly. In this case, the anomaly involves four attributes. Because he usually walks on Monday (and only goes once a day) we expect the cell corresponding to the attribute values {Rene, drives, gym, Monday} to be relatively empty so that an instance suggests an anomaly. It might not be unusual for John to drive to the gym on Tuesday, and it might not be unusual for him to walk to the grocery store on Monday. On the other hand, Jack may routinely drive to the gym on Monday. It is only the four-dimensional combination of attribute values that generates the anomaly. Neither the dimension, the specific attributes involved, nor the form of the pattern is known a priori. Only a training data set of (primarily) normal instances is available. Consequently, this simple example illustrates a challenging problem, but one that I handle with an efficient solution.

To construct D_1 I randomly assign values to each categorical feature (independently) to generate the reference data set. I also assign the label $Y = 1$ to the instances in reference data set D_1. For this study, I consider equal number of instances for two datasets. However, we should note that N_1 doesn't necessarily have to equal N_0, because in practice this is difficult for the case when we have a large number of categorical features, each having a large number of distinct values so that the total possible number of cells is much higher than the number of instances (= N_0). However, keeping them equal leads to balanced two-class case. If the two data sets have different number of instances, we will have imbalanced data set and we should use an appropriate algorithm to mitigate that. The overall dataset provided to the learner is the combined set of D_0 and D_1. Thus, the supervised learner can easily learn the underlying relationship among attribute sets and distinguish between patterned instances and anomalies. Actually, by contrasting D_0 and D_1our algorithm could classify the anomalies of original data set (D_0) as D_1 and separate patterned instances from outliers.

As I discussed, one of the advantages of RF is it gives class probability for each instances. Now given an input instance xi, the learner provides a class probability estimate for each class 0 and class 1, so it shows us xi is classified as which classes. $\hat{p}_0(x_i)$ is probability of instance x_i is in class 0.

Thus, we can set a probability threshold value c such that

$$\hat{p}_0(x_i) \geq c \tag{3}$$

above which the instance is assigned to class 0. An important step is to determine the value of c. To have an accurate result we should fins an optimum c that decrease false positive rate (proportion of negative instances that classified as positive) and increase true positive rate (proportion of positive instances that correctly classified as positive).

The Receiver Operating Characteristic (ROC) will guide us to choose the value of c that leads to the optimal model. ROC curve is empirically generated from the training data and illustrates the performance of a binary classifier as under various values discrimination threshold values [32]

The curve is created by plotting true positive rate against false positive rate at different level of threshold. The best threshold for the model is the one that maximize the area under the curve.

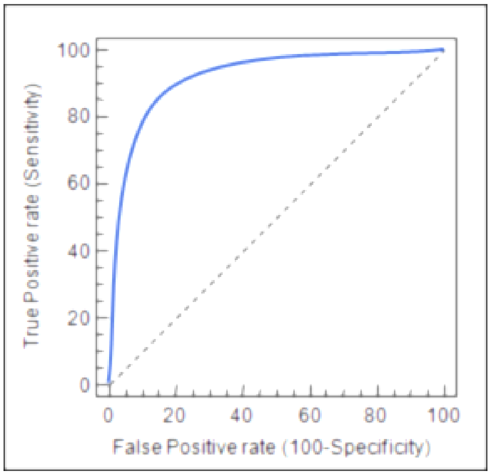

Fig. 4. Example of Roc Curve

For my problem, false positives are defined as an instances $xi\ \in D_0$ s.t. $\hat{p}_0(x_i) < c$. It means that instances that are really belong to class 0 but the probability of being in class 0 is less than the defined threshold, so it will be classified as class 1. True positives are defined as instances $xi\ \in D0$ s.t. $\hat{p}_0(x_i) > c$. It means that instances that are belong to class 1 correctly classified as class1.

When we generate random data, there is a possibility of a random instance being associated with a normal cell (in D_0). Thus, one can expect a number of instances in D_1 to be assigned to class 0. I call these instances false negatives (proportion of positive instances that classified as negative). Such false negatives are the result of the random process used to generate the data, rather than the classifier performance. The value of c is not too sensitive to the presence of pseudo-normal as can be verified experimentally.

1- Generate random data for each variables (D1)

2- Combine the original data (D0) with generated data (D1)

3- Define Class 0 for D0 and Class 1 for D1

4- Apply RF model to the final data set

5- Get prediction probability from the model

6- Check the ROC curve if it is needed to change the probability threshold

7- Get probability histograms for Class 0 and Class 1 to check the class

Fig. 5. Steps of the Algorithm

4.3.1 Experiments

To show the effectiveness of our model, I apply it to three simulated datasets and one real application. All of the analyses are carried out using R 3.3. Details of the packages and functions used for each analysis are included in Table 8 at the end of this document.

In this section I discuss each data set and how the simulated data sets have been generated and also the results.

Low Dimensional Data Set

The first data set is a low-dimensional one with two categorical features, denoted as x_1 and x_2, each having 20 distinct values. The jth value of x_1 is denoted as $x_1(j)$ for j = 1, 2, … , 20 (similarly for $x2$). I generated 500 instances for D_0 and D_1. D_0 should have a pattern so I decide to generate a pattern in 80% of D_0 as follow. I assign 20 instances to each of the 20 cells [$x_1(j)$, $x_2(j)$] for j = 1, 2,…, 20. I call such a pattern set Sp where p is the number of features over which the pattern extends (two in this case). Cells associated with pattern set Sp are called pattern cells. Examples of pattern cells are [car(1), driver(1)], [car(2), driver(2)], and so on for 80% of D_0. This pattern represents a realistic type of pattern. To test the robustness of the model to noises in the original data, I decide to generate random data for the remaining 20% instances. I assign Y=0 to the instances in D_0. I randomly generated the attribute value over p = 2 features independently. D_1 instances are generated randomly over p = 2 features (independently). I assign Y=1 to the instances in D_1 . I combine these to data sets, so we have a unique data set with two classes. I apply RF model on to this data set. The OOB estimate of error is approximately 15%. Table 1 shows the confusion matrix.

TABLE 1 Confusion Matrix of Low Dimensional Dataset

	Predicted Class 0	Predicted Class1	Class Error
Actual Class 0	409	91	0.18
Actual Class1	61	439	0.12

I use OOB estimation for calculating the class probability for each instances. I generate the ROC curve and calculate the area under the curve, which is about 0.86 and it is an acceptable result. The cutoff point to have a zero false positive rate is 0.97 (figure 6).

Figure 7 shows the histogram of class 0 probability for 80% of the instances in D_0 in the pattern set and figure 8 shows the histogram of class 0 probability for D_1. These were class probability estimates obtained from OOB. Clearly, RF is able to distinguish between the D0 and D1 data.

Note that we have a few observations in figure 7 with $\hat{p}_0(x_i)$ = 1. Upon inspecting those 29 observations, they were found to be ones associated with the pattern cells.

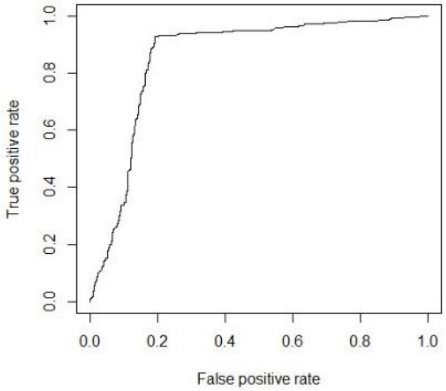

Fig. 6. Roc Curve of Applying the Model on the Low Dimensional Dataset (The Area under the Curve = 0.86)

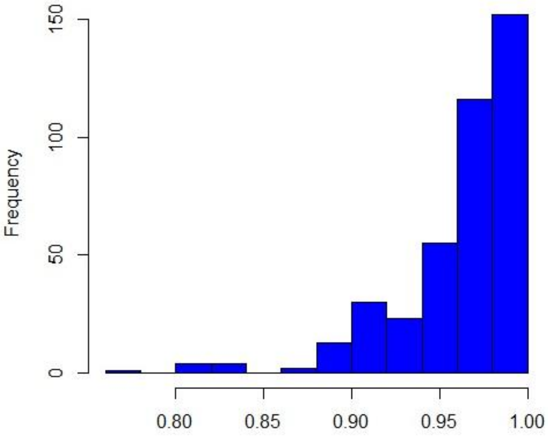

Fig. 7. Histogram for Normal Class Probability Plot in D_0 (The Model Recognizes the Patterned Records with High Accuracy)

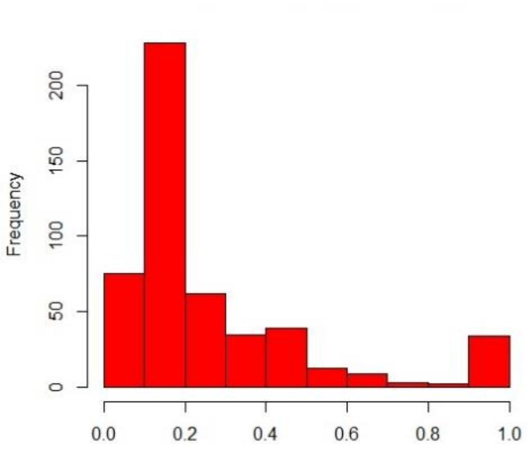

Fig. 8. Histogram for Normal Class Probability Plot in D_1 (The Model Recognizes the Random Generated Records with High Accuracy)

To ensure accuracy of the results, I generate five different D_1 data sets for the same D_0 and five different RF learner models for each data set. The class probability distributions were almost identical for each D_0,D_1 combination. Consequently, I show the figures for a single combination only.

Representative Application

The second data set is high dimensional, and simulated with characteristics that resemble a border security application. It has 20 categorical features each having 20 distinct values, and consists of 5000 instances in each of D_0 and D_1. Although the total number of features is 20, I apply the pattern generating rule (as discussed in the previous subsection) on 80% of D_0 over p = 4 features (so that we have a pattern set S4) and randomly generated attribute values for the remaining 16 features. However, for the remaining 20% of the instances in D_0, I randomly generated attribute values for all 20 features (independently) to test the robustness of RF learner. All Instances for D_1 were generated randomly over p = 20 features. The OOB estimation for this model is about 14%. Table 2 shows the confusion matrix of the model.

TABLE 2 Confusion Matrix of Model on Dataset with 20 Variables

	Predicted Class 0	Predicted Class 1	Class Error
Actual Class 0	3609	1391	0.28
Actual Class 1	0	5000	0

The area under ROC curve is 0.9 which shows our model accuracy (figure 9). Figure 10 shows the histogram of class 0 probability for 80% of the instances in D_0 in the pattern set. Figure 11 shows the histogram of class 0 probability for D_1. The two figures show how our model could well separate the pattern set from random data.

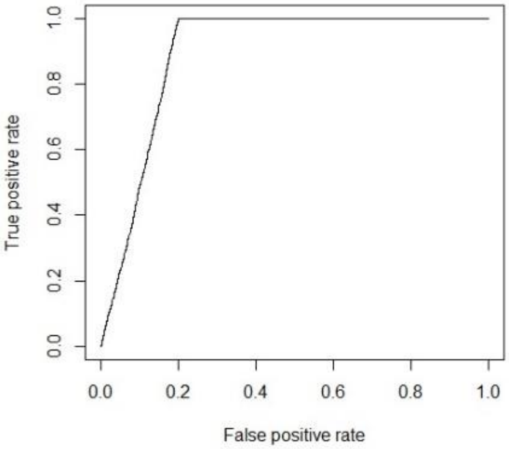

Fig. 9. Roc Curve of Applying the Model on the Dataset with 20 Variables (The Area under the Curve = 0.9)

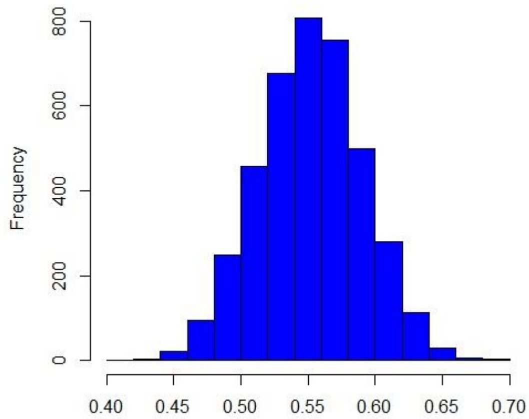

Fig. 10. Histogram for Normal Class Probability Plot in D_0 (The Model Detects the Patterned Dataset)

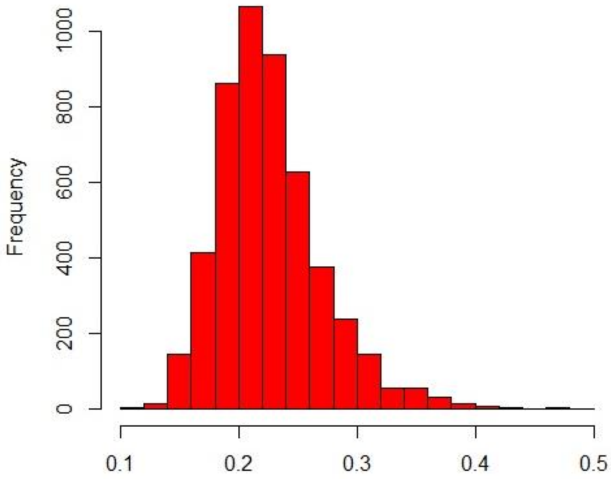

Fig. 11. Histogram for Normal Class Probability Plot in D_1 (The Model Detects the Random Generated Data Records)

Again, I generate five different D_1 data sets for the same D_0 and five different RF learner models for each data set. The class probability distributions were almost identical for each D_0,D_1 combination.

I also do another experiment on higher dimensional data set with 90 variables and 5000 records. I build a data set that has a pattern for 25 features for 80% of records (D_0). For D_1 I randomly generated data for 90 features with 5000 records. Again, I apply RF on the mixed of D_0 and D_1. Table below shows the result of RF model.

TABLE 3 Confusion Matrix of Model on Dataset with 90 Variables

	Predicted Class 0	Predicted Class 1	Class Error
Actual Class 0	5000	0	0
Actual Class 1	0	5000	0

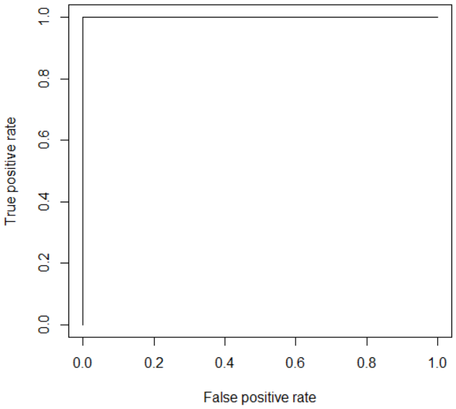

Fig. 12. Roc Curve of Applying the Model on High Dimensional Dataset (The Area under the Curve = 1)

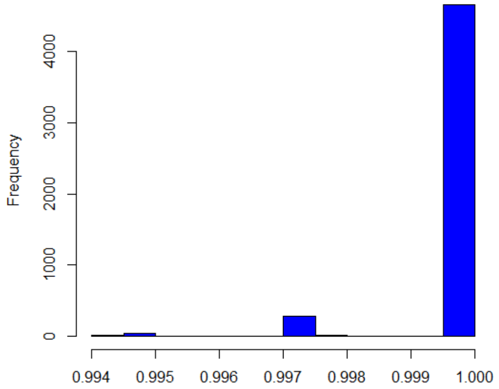

Fig. 13. Histogram for Normal Class Probability Plot in D_0 (The Model Detects the Random Generated Data Records)

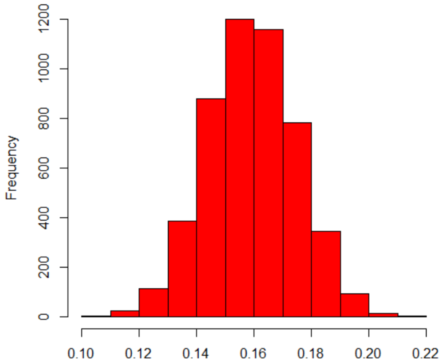

Fig. 14. Histogram for Normal Class Probability Plot in D_1 (The Model Detects the Random Generated Data Records)

I generate five different D_1 data sets for the same D_0 and five different RF learner models for each data set. The class probability distributions were almost identical for each D_0,D_1 combination. The results are almost the same for each run. To demonstrate the similarity of the results as an example, I overlay the ROC curve for one of the 5 replications in figure 15. We can see that all of the ROC are the same. For the previous two experiments, the results are similar for each replications of each dataset. In Table 4 I provide the average minimum, mean and maximum of the area under the curve for the 5 replications of the model on each dataset.

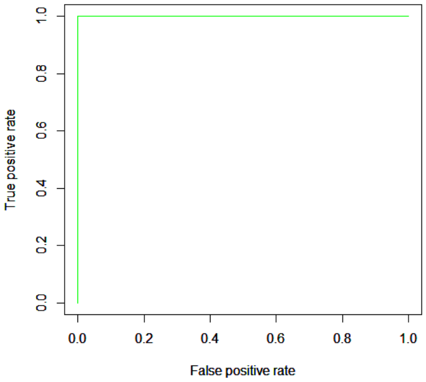

Fig. 15. Roc Curve for 5 Replications of Model for High Dimensional Simulated Dataset

TABLE 4 Average Statistics of Area under Roc Curve for 5 Replications of Model on Test Set for High Dimensional Dataset

# of records of noises in the training set (D0)	Min	Mean	Max
0	1.00	1.00	1.00
500	1.00	1.00	1.00
1000	1.00	1.00	1.00

Feature Selection

To have accurate results and powerful prediction models, our model should have the most relevant attributes. Using redundant attributes in the model could be misleading, and can bias the results. For example, k-nearest neighbor method uses small neighbors for prediction that could be significantly skewed because of redundant attributes that are in the model. Also, analyzing data with irrelevant attributes will cause overfitting problem. For example, decision tree models are based on optimal splits on attributes. Decision tree algorithm first split those attributes that are the most important ones in predicting the response value. Irrelevant attributes are the last ones that are split. If we do not omit irrelevant attributes and put all attributes in the model, we will have an overfitting problem that decrease the prediction accuracy.

Feature selection is a technique that could help us to find the best relevant attributes in a data set to have a more accurate model. There are three general classes of feature selection methods:

Filter Methods

Filter methods assign a score to each attributes of a data set. The features then will be ranked based on that scores. Those scores determine if a feature should be removed from the model or should be stayed. Chi squared test, correlation coefficient scores and information gain are examples of filter methods.

Wrapper Methods

Wrapper methods use a subset of features to build a model. Then the accuracy of each of the models are compared to each other. The subset of features that build the most accurate model is considered the best model. Recursive feature elimination is an example of wrapper method.

Embedded Methods

Embedded methods determine the features that have the most effective contribution in model accuracy. Example of embedded methods is LASSO, Ridge Regression.

Embedded methods use a penalty in a model (e.g. regression) to decrease the bias towards low complexity.

RandomForest is an efficient feature selection technique, which is efficient in categorical data sets. I used RF to select the most relevant attributes and do the same analysis with only those attributes.

The OOB estimation for this model is 10% which has been decreased from the previous model.

TABLE 5 Confusion Matrix of Model on Dataset with 20 Variables after Applying Feature Selection

	Predicted Class 0	Predicted Class 1	Class Error
Actual Class 0	4008	992	0.2
Actual Class1	35	4965	0.007

Figures 16 and 17 show the class 0 probability estimates of D_0 and D_1 data, respectively, after employing feature selection techniques. These figures show an improved separation between the two classes from that obtained from figures 10 and 11.

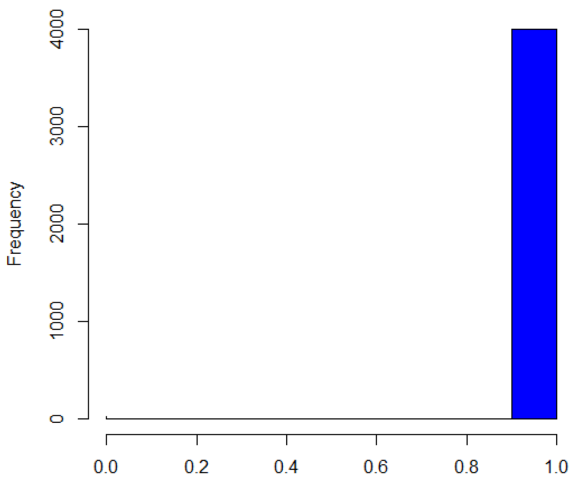

Fig. 16. Histogram for Normal Class Probability Plot in D_0 (The Accuracy of Model Improved after Applying Feature Selection on the Data Set with 20 Variables)

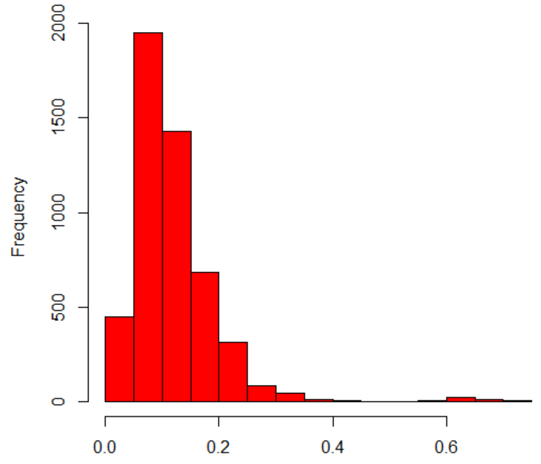

Fig. 17. Histogram for Normal Class Probability Plot in D_1 (The Accuracy of Model Improved after Applying Feature Selection on the Data Set with 20 Variables)

Finally, I conduct another experiment for a null case when both the D_0 and D_1 data are randomly generated over 20 features (N0 = N1 = 5000). Since there is no pattern in the normal data, intuitively the classifier should not be able to learn any relationship among the attribute sets.

Thus, we should not be able to distinguish between the D_0 and D_1 data. Figures 18 and 19 show the class 0 probability estimates for the D_0 and D_1 data, respectively, for this null case. They turn out to be almost identically distributed which verifies our intuition.

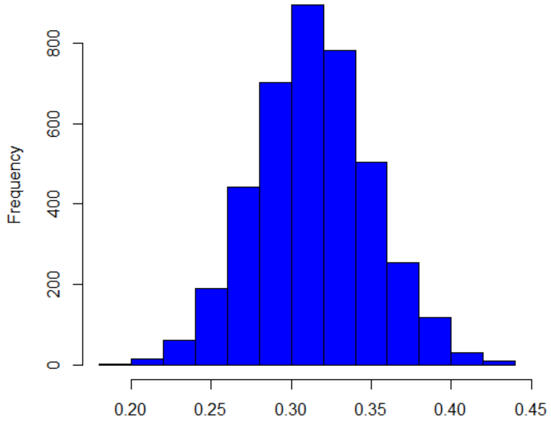

Fig. 18. Histogram for Normal Class Probability Plot in D_0 (All Data Are Randomly Generated)

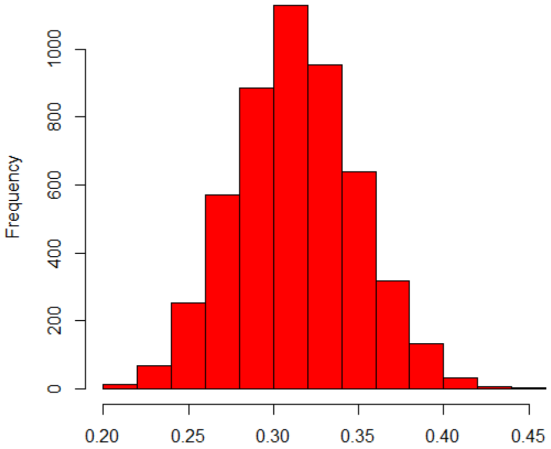

Fig. 19. Histogram for Normal Class Probability Plot in D_1 (All Data Are Randomly Generated)

4.4 Method Comparison

To evaluate the performance of the algorithm, I apply the model to a real life data set and compared the result with some other methods that have been developed for detecting outliers in categorical data sets. The methods that I use in the comparison are FindFPOF algorithm [25], FindCBLOF algorithm [22].

For all of the experiments, the two parameters needed by the FindCBLOF algorithm were set to 90% and 5 separately as done in Hawkins et al. [33]. For our algorithm, the parameter minisupport for mining frequent patterns was fixed to 10%, and the maximal number of items in an itemset was set to 5. The number of rare cases identified was used as the assessment basis for comparing our algorithm with other algorithms.

4.4.1 Data

The dataset that I use in this section is Wisconsin breast cancer data set, which has 9 categorical attributes with 699 records. The data set has two classes: benign (458 or 65.5%) or malignant (241 or 34.5%). Based on Hawkins et al.'s [33] experimental method I removed some of the malignant class to form a very unbalanced data set.

I create a training dataset with 222 records of benign class and 20 records of malignant class and set them as class 0 (D_0), I also generate random data for each attribute (D_1). I ran a RF model to the mixed of D_0 and D_1. I use a test set with 222 rows of benign records and 19 records of malignant records for prediction. I rank all the prediction records based on outlier factor (for our model I rank the records based on the probability of being outliers) and then I calculate the coverage for top n% of rows.

In Table 6 below we can see that our proposed model can detect outliers in lower percentage of records, so our model outperforms.

TABLE 6 Comparison of Methods

Top Ratio of Records	Number of Rare Classes Included (Coverage)		
	Proposed model	FindFPOF	FindCBLOF
0%	0	0	0
1%	10.53%	7.69%	10.26%
2%	21.95%	17.95%	17.95%
4%	47.37%	35.90%	35.90%
6%	68.42%	53.85%	53.85%
8%	78%	71.79%	69.23%
10%	84%	79.49%	82.05%
12%	100%	89.74%	89.74%
14%	100%	100%	97.44%
16%	100%	100%	100%
18%	100%	100%	100%
20%	100%	100%	100%
25%	100%	100%	100%
28%	100%	100%	100%

To show the accuracy of our model for different instances of malignant and benign records, we built four different training datasets with 222 benign and 0, 10, 20 and 40 malignant observations. The test set has 222 benign and 19 malignant observations. We ran RandomForest (RF) model on the training sets set 5 times for each dataset (20 models in total) and then applied the model on the test to make sure that the results are accurate. We considered the malignant records as positive class (anomalies) and benign instances as negative class (normal). The results were almost the same for all runs, we generated the ROC curves for each run on the test set and overlaid the 5 ROC curves of the model on the first dataset (dataset has 0 malignant records) in figure 20 to demonstrate their similarity as an example. For all of the other models the ROC curves were approximately the same for the 5 replications of each dataset. I provided the average minimum, mean and maximum of the area under the curve for the five runs of the model on each data set. From the ROC curves that are provided below we can see that the accuracy of the models is slightly decreasing when we increased the number of malignant observations in the training sets. It shows that when we have more anomalies on the training dataset, the power of the model to detect the anomalies is decreasing. Although, the area under the curve is quite high even for 40 malignant rows.

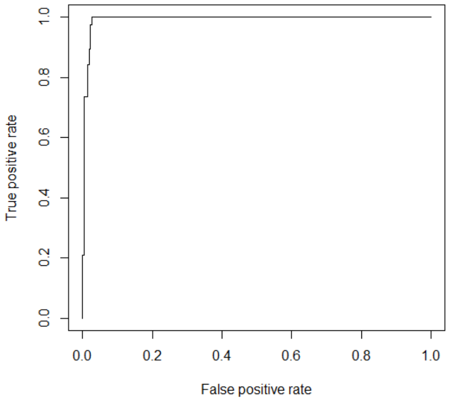

Fig. 20. Roc Curve of Test Set with 0 Malignant Records in the Training Set (Area under the Curve = 0.99)

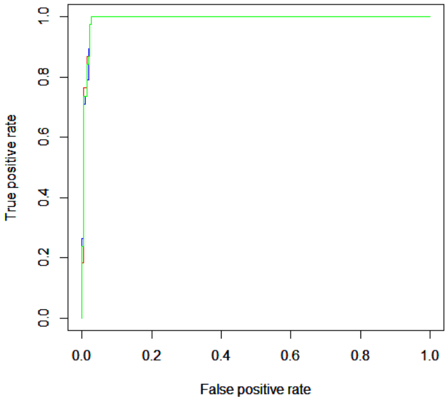

Fig. 21. Roc Curve of Test Set for 5 Replications of Model with 0 Malignant Records in the Training Set (All of the Curves Are Almost Identical)

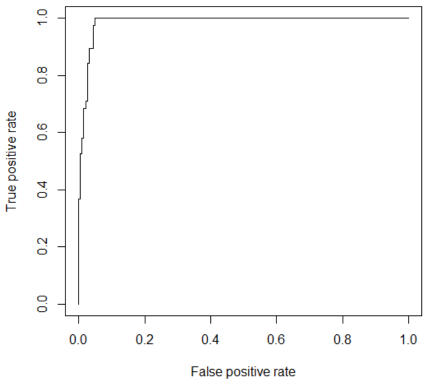

Fig. 22. Roc Curve of Test Set with 10 Malignant Records in the Training Set (Area under the Curve = 0.98)

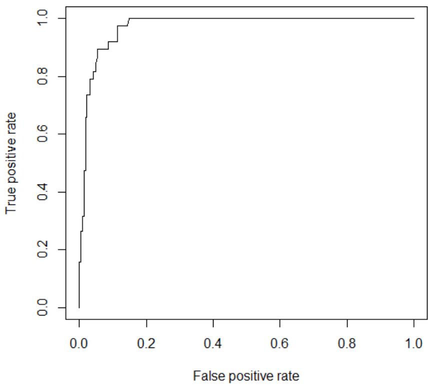

Fig. 23. Roc Curve of Test Set with 20 Malignant Records in the Training Set (Area under the Curve = 0.97)

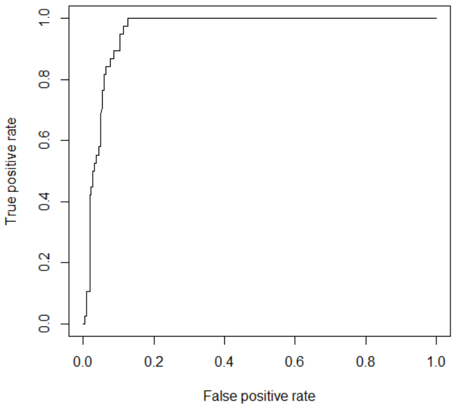

Fig. 24. Roc Curve of Test Set with 40 Malignant Records in the Training Set (Area under the Curve = 0.95)

TABLE 7 Average Statistics of Area under Roc Curve for 5 Replications of Model on Test Set with Different Malignant Records in Training Set

# of benign records in training set	# of malignant records in training set	Min	Mean	Max
222	0	0.9927	0.9929	0.993
222	10	0.9854	0.9856	0.9862
222	20	0.9736	0.974	0.9745
222	40	0.9549	0.956	0.9567

I also show that when we decrease the number of benign observations and increase malignant observations in the training set, the accuracy of model will decrease. I run models on two training datasets with 100 and 50 benign and 241 malignant observations. I repeat the model 5 times to make sure that the accuracy of the models is high. I provide ROC curve and the area under the curve of the models. I also include minimum, mean and maximum of the area under the curve for the five runs of the model on each data set. From the ROC curves that are provided below, we can see that when the malignant observations are more than the benign observations the accuracy of the model has decreased. In table 8 I also show the average minimum, mean and maximum of the area under the curve for 5 replications of model on datasets with different benign records.

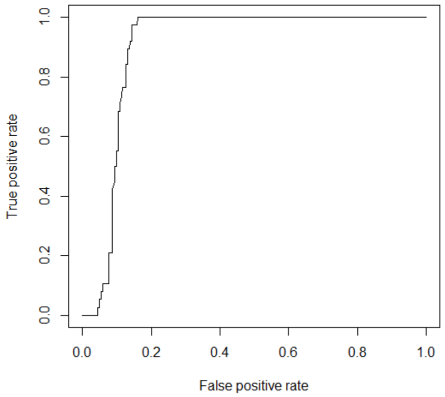

Fig. 25. Roc Curve of Test Set with 100 Benign Records in the Training Set (Area under the Curve = 0.90)

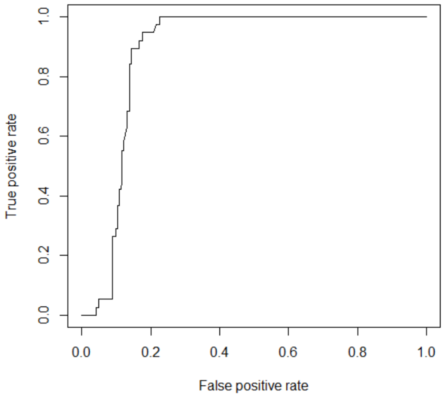

Fig. 26. Roc Curve of Test Set with 50 Benign Records in the Training Set (Area under the Curve = 0.87)

TABLE 8 Average Statistics of Area under Roc Curve for 5 Replications of Model on Test Set with Different Benign Records in Training Set

# of benign records in training set	# of malignant records in training set	Min	Mean	Max
100	241	0.90	0.90	0.90
50	241	0.87	0.87	0.88

TABLE 9 Packages and Codes

Method	**R package**	**R code**
RandomForest	randomForest	randomForest(y ~ x, data, ntree,…)
Prediction	randomForest	predict(RF model, newdata)

ROC curve	ROCR	prediction(OOB.pred,test$Class)
		performance(pred.obj, "tpr","fpr")

Area Under Curve (AUC)	ROCR	auc(actual class, predicted class)

5 CONCLUSION

I develop a novel method for detecting anomalies in categorical datasets. The model is based on Random Forest and artificial contrasts and transforms the usual unsupervised problem of point anomaly detection into a supervised one. Using Random Forest, the method was able to learn the attribute relationships in high dimensional categorical data, and predict the class probability with high accuracy. Moreover, the proposed method incorporates feature selection in case of high-dimensional data to identify the features involved in the pattern, and thus convert the problem into a low-dimensional one. I demonstrate performance of the method in two simulated data sets as well as the superiority of the model to other models in the literature when applied to real worlds datasets. The artificial contrast method is able to learn the relationship found in normal instances, and thus, to distinguish between normal and anomalous instances. Although I use a Random Forest classifier, other classifiers can easily be used. It will also be useful to identify groups of anomalous instances, sometimes known as collective anomalies to understand the mechanism behind their origin.

6 REFERENCES

[1] W. Hwang, G. Runger, and E. Tuv, “Multivariate statistical process control with artificial contrasts,” IIE Trans., vol. 39, no. 6, pp. 659–669, 2007.

[2] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv. CSUR, vol. 41, no. 3, p. 15, 2009.

[3] M. E. Otey, A. Ghoting, and S. Parthasarathy, “Fast distributed outlier detection in mixed-attribute data sets,” *Data Min. Knowl. Discov.*, vol. 12, no. 2–3, pp. 203–228, 2006.

[4] C. De Stefano, C. Sansone, and M. Vento, “To reject or not to reject: that is the question-an answer in case of neural classifiers,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 30, no. 1, pp. 84–94, 2000.

[5] 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*, 1997, pp. 220–226.

[6] J. Heggestuen, “The US Sees More Money Lost To Credit Card Fraud Than The Rest Of The World Combined,” *Business Insider*, 2014. [Online]. Available: http://www.businessinsider.com/the-us-accounts-for-over-half-of-global-payment-card-fraud-sai-2014-3. [Accessed: 03-Oct-2016].

[7] CreditCards.com, “Credit card fraud and ID theft statistics.” [Online]. Available: http://www.creditcards.com/credit-card-news/credit-card-security-id-theft-fraud-statistics-1276.php. [Accessed: 03-Oct-2016].

[8] S. Donoho, “Early detection of insider trading in option markets,” in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 420–429.

[9] C. C. Aggarwal, “On Abnormality Detection in Spuriously Populated Data Streams.,” in *SDM*, 2005, pp. 80–91.

[10] V. Bamnett and T. Lewis, “Outliers in statistical data,” 1994.

[11] L. Wei, W. Qian, A. Zhou, W. Jin, and X. Y. Jeffrey, “Hot: Hypergraph-based outlier test for categorical data,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2003, pp. 399–410.

[12] P.-N. Tan and others, *Introduction to data mining*. Pearson Education India, 2006.

[13] F. P. Preparata and M. Shamos, *Computational geometry: an introduction*. Springer Science & Business Media, 2012.

[14] I. Ruts and P. J. Rousseeuw, “Computing depth contours of bivariate point clouds,” *Comput. Stat. Data Anal.*, vol. 23, no. 1, pp. 153–168, 1996.

[15] R. Smith, A. Bivens, M. Embrechts, C. Palagiri, and B. Szymanski, “Clustering approaches for anomaly based intrusion detection,” *Proc. Intell. Eng. Syst. Artif. Neural Netw.*, pp. 579–584, 2002.

[16] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[17] V. Kumar, “Parallel and distributed computing for cybersecurity,” *Distrib. Syst. Online IEEE*, vol. 6, no. 10, 2005.

[18] S. D. Bay and M. Schwabacher, “Mining distance-based outliers in near linear time with randomization and a simple pruning rule,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 29–38.

[19] D. Barbara, N. Wu, and S. Jajodia, “Detecting Novel Network Intrusions Using Bayes Estimators.,” in *SDM*, 2001, pp. 1–17.

[20] E. M. Knorr, R. T. Ng, and V. Tucakov, “Distance-based outliers: algorithms and applications,” *VLDB Journal— Int. J. Very Large Data Bases*, vol. 8, no. 3–4, pp. 237–253, 2000.

[21] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, and others, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *Kdd*, 1996, vol. 96, pp. 226–231.

[22] Z. He, X. Xu, and S. Deng, “Discovering cluster-based local outliers,” *Pattern Recognit. Lett.*, vol. 24, no. 9, pp. 1641–1650, 2003.

[23] L. Ertöz, M. Steinbach, and V. Kumar, “Finding topics in collections of documents: A shared nearest neighbor approach,” in *Clustering and Information Retrieval*, Springer, 2004, pp. 83–103.

[24] A. Ghoting, S. Parthasarathy, and M. E. Otey, “Fast mining of distance-based outliers in high-dimensional datasets,” *Data Min. Knowl. Discov.*, vol. 16, no. 3, pp. 349–364, 2008.

[25] Z. He, X. Xu, J. Z. Huang, and S. Deng, “FP-outlier: Frequent pattern based outlier detection.,” *Comput Sci Inf Syst*, vol. 2, no. 1, pp. 103–118, 2005.

[26] Z. He, S. Deng, and X. Xu, “An optimization model for outlier detection in categorical data,” in *International Conference on Intelligent Computing*, 2005, pp. 400–409.

[27] 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*, 2007, pp. 220–229.

[28] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[29] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 63, no. 2, pp. 411–423, 2001.

[30] E. Tuv, A. Borisov, and K. Torkkola, “Feature selection using ensemble based ranking against artificial contrasts,” in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006, pp. 2181–2186.

[31] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[32] “Receiver operating characteristic,” *Wikipedia, the free encyclopedia*. 08-Sep-2016.

[33] S. Hawkins, H. He, G. Williams, and R. Baxter, “Outlier detection using replicator neural networks,” in *International Conference on Data Warehousing and Knowledge Discovery*, 2002, pp. 170–180.