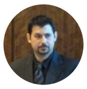
Application of Anomaly Detection Techniques to Identify Fraudulent Refunds

Article *in* SSRN Electronic Journal · August 2011

DOI: 10.2139/ssrn.1910468



Hussein Issa
Rutgers, The State University of New Jersey

Miklos A. Vasarhelyi
Rutgers Business School

Application of Anomaly Detection Techniques to Identify Fraudulent Refunds

Miklos A. Vasarhelyi, Ph.D.

KPMG Professor of AIS, Rutgers Business School

Hussein Issa

PhD Student, Rutgers Business School

ABSTRACT

Anomaly detection is a concept widely applied to numerous domains. Several techniques of anomaly detection have been developed over the years, in practice as well as research. The application of this concept has extended to diverse areas, from network intrusion detection to novelty detection in robot behavior. In the business world, the application of these techniques to fraud detection is of a special interest, driven by the great losses companies endure because of such fraudulent activities. This paper describes classification-based and clustering-based anomaly detection techniques and their applications, more specifically the application to the problem of certain fraudulent activities. As an illustration, the paper applies K-Means, a clustering-based algorithm, to a refund transactions dataset from a telecommunication company, with the intent of identifying fraudulent refunds.

Keywords: anomaly detection, classification, clustering, fraud detection

1. INTRODUCTION

Anomaly detection is a data mining technique that is applied to various areas. One problem is especially important to the business world, and that is fraud detection. Fraud in this context is best described by Phua et. al. as "*the abuse of a profit organization's system without necessarily leading to direct legal consequences"* (Phua, V. Lee, Smith, & Gayler, 2005). With the dramatic increase in the amount of data gathered in today's business world, manual verification and screening of all the transactions is extremely costly, if at all possible. Hence, it becomes essential to use automated techniques to ensure this verification, especially in the case of companies that deal with millions of external parties. With this automation, only *suspicious* activities need to be investigated (Phua, V. Lee, Smith, & Gayler, 2005). The question is which activities should be treated as suspicious, and what criteria must be considered when making such a decision? Anomaly detection techniques can help answer these questions.

Outlier or anomaly detection is not a new concept. In fact it has been studied and researched since the 19th century (Edgeworth, 1887). The difference between the past and the present is that advances in technology made it possible for more efficient applications of this approach. However, before discussing anomaly detection, it is necessary to describe what is meant by anomaly. An anomaly is an instance (record, transaction, etc.) that does not conform to a well-defined and general pattern of the expected behavior in a certain dataset (Figure 1). Anomalies are sometimes known in different areas as outliers, aberrations, exceptions, or peculiarities (Chandola, Banerjee, & Kumar, 2009).

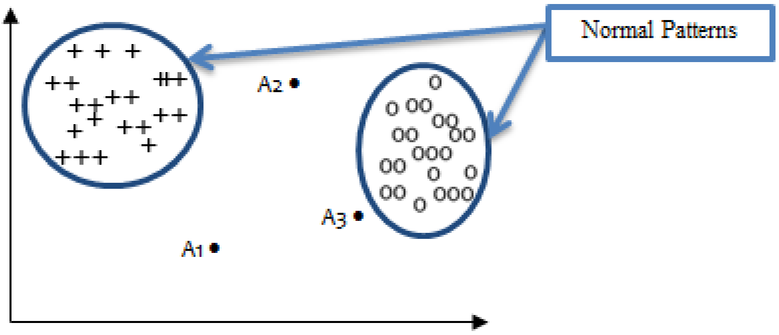


Figure 1. Anomalies and Normal patterns

The idea behind anomaly detection is simple: define a pattern of the normal behavior, and consider any instance in the dataset that does not conform to this pattern as anomalous. However, this is easier said than done. Many factors play a role in complicating this seemingly simple issue. First, it is hard to define a pattern that can represent all possible normal behaviors, and very often an anomaly lying close to the boundaries of a normal region can in fact be a normal instance (e.g. Point A3 in Figure 1 and Point A2 in Figure 2). The opposite could also be true.

Moreover, this normal behavior is, in most cases, dynamic in nature, and consequently it keeps changing (e.g. antivirus definitions). Today’s normal pattern may be anomalous tomorrow, or vice versa. In fact, the mere definition of an anomalous behavior changes from one area to another, which makes the application of a domain-specific technique hard to generalize to other domains. For example, a small deviation of blood cholesterol level from normal can indicate an anomaly, whereas the same deviation in oil prices could be considered normal. Therefore, a technique used in the medical field may not be useful for detecting oil price anomalies. When it comes to fraud detection, persons who are committing this fraud would usually camouflage these activities in a way that makes them follow the normal behavior, making them harder to detect. In addition to these challenges, datasets sometimes are contaminated with noise, which is different from anomalies, although similar in behavior.

On top of these issues, data labels are usually unavailable, making the training and testing of anomaly detection models harder to perform (Chandola, Banerjee, & Kumar, 2009). In fact, fraud detection research that uses data mining techniques is criticized for the scarcity of real data available to the public, which would be used to test the detection models. This paper applies anomaly detection techniques to detect fraudulent refunds in a real data from a telecommunications company. It also compares the results achieved using classification-based and clustering-based techniques.

The remainder of this paper is organized as follows. Section 2 discusses the different aspects of the anomaly detection problem. Section 3 presents the applications of anomaly detection. Section 4 describes different anomaly detection techniques. Section 5 describes the dataset and methodology used in this study. Section 6 discusses the results. Section 7 concludes the papers.

2. ASPECTS OF ANOMALY DETECTION

In order to properly address an anomaly detection problem, several factors have to be taken into consideration. In fact, these are the factors that lead to the wide diversity of techniques used in the anomaly detection problems.

The first of these aspects is the nature of input data, which is a collection of records (or instances, objects, events, etc.) described using a set of attributes or variables. An instance could be univariate (with one variable) or multivariate (several variables) (others, 2005). The type of these variables (categorical, continuous, and binary) affects the techniques that can be used on that dataset. The relation between records also affects the applicability of a certain technique. For example, in time series datasets, instances are related in time. Spatial datasets consist of records that are related to their neighboring records (e.g. traffic) (Chandola, Banerjee, & Kumar, 2009).

The second aspect of anomaly detection is the type of anomaly. There are three main types of anomalies: point anomaly, collective anomaly, and contextual anomaly (Figure 2).

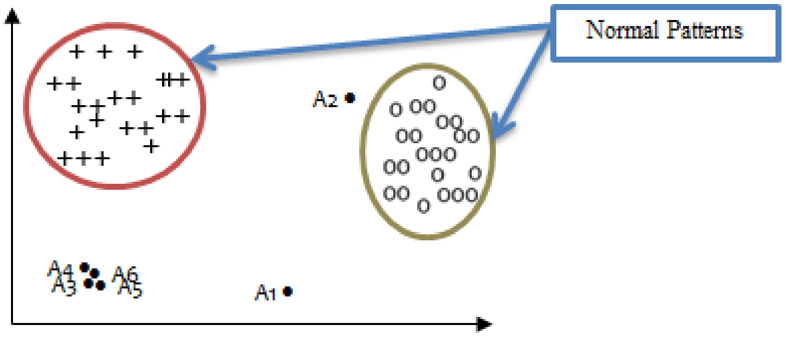


Figure 2. Types of Anomalies

Point anomalies are instances that are individually considered as anomalies to the normal behavior (Point A1 in Figure 2). For example, if an individual's spending pattern using her credit card is in the range of $20 to $100, a payment of $1500 is by itself a point anomaly and therefore worth investigation.

The second type of anomalies is collective anomalies. In this case it is not one instance that behaves anomalously, but a group of related instances that behave differently from the general pattern of the overall data (Points A3, A4, A5, & A6 in Figure 2). The instances by themselves are not anomalous, but when they occur together, they form a collective anomaly.

Contextual anomalies, on the other hand, take into consideration the context where the instance occurred. An instance can be considered normal in one situation but anomalous in another (Song, Wu, Jermaine, & Ranka, 2007). For example, while a temperature of 25 degrees Fahrenheit is treated as normal in January, it is considered anomalous if it occurs in July. The instance is then judged based on a contextual attribute (time of the year) and a behavioral attribute (temperature in degrees). This type of anomalies is mostly studied in time-series data and in special data (Salvador, Chan, & Brodie, 2004)(Kou, Lu, & Chen, 2006)(Shekhar, Lu, & Zhang, 2002).

It is worth mentioning that both point anomalies and collective anomalies can be contextual. Take the credit card spending example mentioned earlier. If the usual pattern is $200 per week, a $1000 during Christmas week is considered normal, as opposed to the same $1000 during a week in May.

The third factor affecting the formulation of the anomaly detection problem is the level of data labeling. The label of an instance can be either *normal* or *anomalous*. However, it is often very hard to get a labeled dataset to train a detection model, and even if labels are available, they may not be correct (Brockett, Derrig, Golden, Levine, & Alpert, 2002). The main reason behind that is the process of labeling: it is usually done manually by a human, hence the high cost and uncertainty of labels. This is especially true for anomalies, as it is very hard to capture all possible types of anomalous behavior, which is dynamic in nature. In fact, the level of data labeling decides on the mode of detection to be used: supervised, semi-supervised, and unsupervised (discussed in section 4).

3. APPLICATIONS OF ANOMALY DETECTION

Anomaly detection is widely applied in research and in practice. These applications include network intrusion detection, fraud detection, fault detection in safety systems, novelty detection, and enemy activities surveillance (military use) (Chandola, Banerjee, & Kumar, 2009). As the interest of this paper is the application of anomaly detection approach to capture fraudulent activities, the discussion will be limited to the domain of fraud detection.

First the notion of fraud detection needs to be defined. It is the detection of any illegitimate actions taking place in a business/commercial environment, which would end up with an unauthorized use of the organization’s resources. Persons or entities who are committing this fraud may be employees, customers, or imposters who would steal the identity of real customers. That said, it is only normal that organizations try to detect this fraud as soon as possible, as the longer the fraud continues, the greater the possible loss to the organization. Anomaly detection comes in handy in this case, as it allows for detecting customers’ behavior patterns and profiling users based on their activity patterns, and can detect fraudulent activities by monitoring users' activities and comparing them to the established profile (Fawcett & Provost, 1999).

One of the main applications of anomaly detection techniques is the area of credit card fraud, especially in the case of stolen credit cards. Suspicious transactions (high expenditures, new type of payments, etc.) are compared to that customer's profile, and if they don't follow the pattern, they are considered as anomalous, and are further investigated.

Another important area where anomaly detection was found to be effective is fraud in insurance claims. Fraud in the domain of insurance is very costly to insurance firms, hence their interest in detecting fraudulent claims. Usually insurance companies acquire labeled data from claim adjustors and investigators, who manually check claims and determine the legitimate vs. fraudulent claims. This labeled data is then used to train a model using supervised/semi-supervised fraud detection technique (such as neural networks), and new instances are tested against this model and designated either as normal or anomalous (i.e. fraudulent) (Fawcett & Provost, 1999)(Brockett, Xia, & Derrig, 1998).

4. ANOMALY DETECTION TECHNIQUES

Before discussing the different techniques used in anomaly detection, it is important to describe the modes under which those techniques can operate. As mentioned at the end of section 2, the level of data labeling determines which mode can be used.

The supervised mode operates under the assumption that both normal and anomalous instances are labeled. The anomaly detection technique then trains a model using the labeled data to setup a normal class and an anomalous one. New instances are then tested with that predictive model, and are assigned to one of those classes. The main problem with supervised learning is that it is very hard to get data that is correctly labeled (as this is done manually), especially for the anomalous instances (hard to detect). Moreover, the normal class is usually much bigger than the anomalous class, creating some kind of unbalance (Abe, Zadrozny, & Langford, 2006).

When only normal instances can be labeled, semi-supervised techniques are more efficient. This is more applicable than the supervised as the critical requirement of having labeled anomalous instances, which is hard to satisfy, is not necessary. When neither normal nor anomalous classes can be labeled, the unsupervised mode is the only one to use. Knowing that this is the case of most real-world datasets, it is understandable that unsupervised mode is the most widely used and the most popular among the three modes. It does assume, however, that the normal class is much bigger than the anomalous one. If this assumption fails, this mode ends up with a high rate of false positives.

To discuss anomaly detection techniques, it is useful to divide them into 2 broad categories:

4.1. Classification-based techniques:

The main idea behind classification is to use a labeled dataset to train a model, and classify those instances (training phase). Any instance that is tested later on will be designated as anomalous if it does not belong to any normal class (testing phase) (others, 2005) (Duda, Hart, & Stork, 2001). Classification-based techniques can be either multi-class or single-class categories. In the former case, the model contains multiple normal classes, and a classifier is used to differentiate between each of those classes against the other. If the new instance to be tested is not found to belong to any of these normal classes, it is considered as anomalous (De Stefano, Sansone, & Vento, 2000). On the other hand, single-class technique classifies *all* the normal instances into one normal class, and therefore any test instance that does not fall into that region is considered anomalous. Figure 4 and 4 illustrate single-class and multi-class classification techniques, respectively.

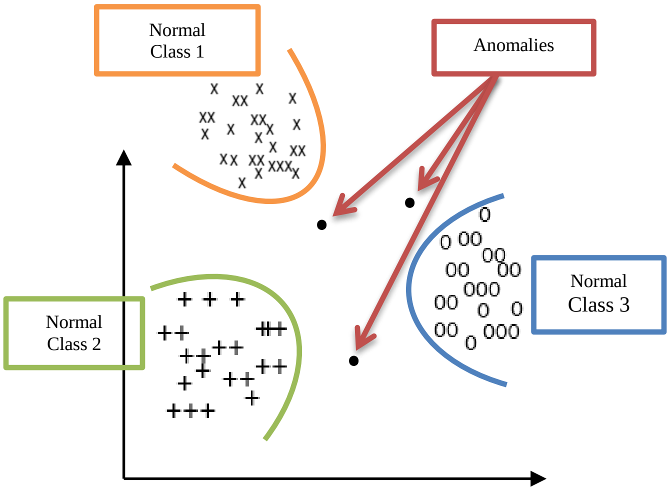


Figure 4. Multi-class classification

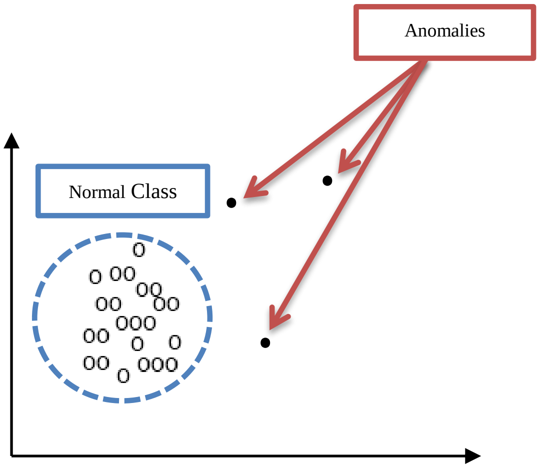


Figure 3. Single-class Classification

Below are some of the popular classification techniques that are used in anomaly detection.

Neural networks

Neural networks are multi-class classification techniques that consist of two phases. The first step is to train a neural network on the labeled data instances and learn the normal classes, and then new instances are introduced. If the instance is rejected by the neural network it is classified as anomaly (De Stefano et al., 2000).

Bayesian Networks

The concept behind this technique is estimating the posterior probability of observing a class label using a naïve Bayesian network. The predicted class of the test instance would be the class label with the highest posterior probability. This technique is popular in different applications, such as network intrusion detection, novelty detection in video surveillance, and anomalies in text data (Valdes & Skinner, 2000) (Baker, Hofmann, Mccallum, & Yang, 1999).

Support Vector Machine (SVM)

SVMs learn a region containing normal instances using a single-class learning method. If the test instance falls within the boundary of that learned space, it is normal. If it does not, it is classified as anomalous. This technique is widely used in audio signal anomaly detection, novelty detection, and system call intrusion detection (S. King et al., 2002).

Rule-based techniques

This technique defines the normal behavior pattern using rules. It trains a model using a rule-based learning algorithm. Next the test instance is tested by the model to see which rule best captures it. If no such rule could be found, the test instance is considered anomalous. Association rules is a variant of this technique that operates under unsupervised mode, where it generates the rules from the data itself. It is applied to the domains of network intrusion detection, call intrusion detection, and more importantly fraud detection (Brause, Langsdorf, & Hepp, 2002)(W. Lee & S. J. Stolfo, 1998).

Computational Complexity, advantages and disadvantages

The computational complexity of classification techniques depends largely on the algorithm that is used. In general, decision trees perform faster than the more complex quadratic SVMs. Although the training phase may be slow, testing is usually much faster as the classes are already defined. Another advantage of classification based techniques is that they can take advantage of some powerful classifying algorithms. On the other hand, these techniques depend on the accuracy of instances labels, in addition to the fact that it is very hard to get accurately labeled datasets.

4.2. <u>Clustering-based techniques:</u>

The concept behind clustering is a simple one. Instances with similar attributes are grouped into clusters (others, 2005). The next step would be to test new instances against the model, and see if they fit into any of the defined clusters. There are three types of clustering techniques that operate under different assumptions:

i. ***Category 1:*** The first category assumes that all normal instances belong to a cluster, while anomalous instances do not belong to any cluster. In other words, a clustering algorithm is used to train the model and define a cluster of normal instances, and then any instance that does not conform to those criteria is considered an anomaly.

ii. ***Category 2:*** The second category assumes that all normal instances are close to the center of their closest cluster, while anomalies are far from the center of their closest cluster. Here the distance is measured, and a score is assigned to the test instance. The advantage of this category is that it can operate under the semi-supervised mode, and the distance-based score is used to determine whether the test instance is anomalous or not, after comparing it to the normal cluster. It is mostly applicable in the areas of fraud detection, intrusion detection, and fault detection (Brockett et al., 1998).

iii. ***Category 3:*** The assumption made under the last category is that anomalous instances by themselves form small clusters. In other words, the dataset will consist of big and thick clusters, considered normal, and small or thin clusters, containing anomalous instances. In other words, all instances would belong to clusters, however the size and density of this cluster determines whether the instance is anomalous or normal. The threshold can be specified by the user or can be generated from the data (Eskin, Arnold, Prerau, Portnoy, & S. Stolfo, 2002).

Computational Complexity, advantages and disadvantages

Similar to classification, the computational complexity depends on the algorithm used to train the model. If all the instances are to be compared pairwise, the computationa will be quadratic, as opposed to linear complexity when heuristic techniques are used. Again, once the clusters are formed, the testing phase becomes much faster.

One of the most important characteristics of clustering-based techniques is that they can operate under unsupervised mode. Moreover, it is enough to plug in to right algorithm to be able to apply this technique to complex datasets.

On the other hand, some algorithms force the allocation of each instance in the training set to a cluster. That may lead to the misclassification of an anomalous test instance, as it would be assigned to a large cluster, and consequently treated as normal. Moreover, clustering techniques are not efficient when the anomalies form large clusters, as it would be hard to distinguish them from normal behavior.

5. DATASET AND METHODOLOGY

5.1. Dataset

The dataset used in this study covers a period of 2 years (July 2008 to June 2010) and consists of 13,199 records representing customer refund transactions from a telecommunication company, and includes 51 variables (before data cleaning). A complete list of all the variables (initial and transformed) can be found in Appendix A.

During the data cleansing stage, 30 records contained some errors, and had to be removed. At the end, the number of usable records went down to 13,169 records.

In order to increase the efficiency of the algorithm, redundant and irrelevant variables were omitted from the dataset. Other variables also had to be excluded due to the high percentage of missing values. Moreover, certain filters had to be applied (based on the suggestions of the company personnel).

In addition to that, certain variables had to undergo a transformation to be in a useful format. The variables DIST_CREATED_BY and DISY_UPDATE_BY (indicating the person who created and updated the record, respectively) had to be transformed to be used for clustering. The date on the other hand was first changed from standard format to Julian Date, then the number of days separating the record date from the first day in the dataset was calculated (to measure the period separating the two events). The variables that were eventually used in this study are presented in Table 1:

Table 1. Variables used in the study

Variable	Description
ORG_ID	Organization's ID number
CHECK_AMOUNT	Amount of check
CHECK_DATE_COORD	Date of check (Transformed)
DIST_CREATED_DISCR	Person who created the record (Transformed)
DIST_UPD_DISCR	Person who updated the record (Transformed)

In the absence of labeled instances, unsupervised techniques are generally preferred. In this study, a clustering technique was applied to the dataset, using an open-source data mining software called Weka.

5.2. K-Means

Simple K-Means is an iterative clustering algorithm that splits the dataset in a pre-specified number of clusters. It owes its popularity to the simplicity and easiness of its application.

The first step in the K-Means procedure is to build an initial partition by choosing the number of clusters, K, and selecting K data points to act as the centroids[1] of these initial clusters. The second step is to assign each data point to the cluster of the closest centroid. Next, new centroids are computed for the K clusters, and a new partition is generated. Steps 2 and 3 are then iterated until the clusters stabilize and the centroids stop changing (Jain, 2010).

K-Means depends mainly on three parameters specified by the user: value of K (number of clusters), initial partition (selection of initial centroids), and metric used to measure the distance

[1] A centroid is the center of a cluster.

between records and centroids. It is important to choose these parameters carefully, as the performance of the algorithm will be determined by this choice.

The main objective of the K-Means algorithm is to minimize the sum of the squared error for all the clusters. This measure was used in this experiment to compare the results of choosing different values of K. These results are discussed in the next section.

6. RESULTS

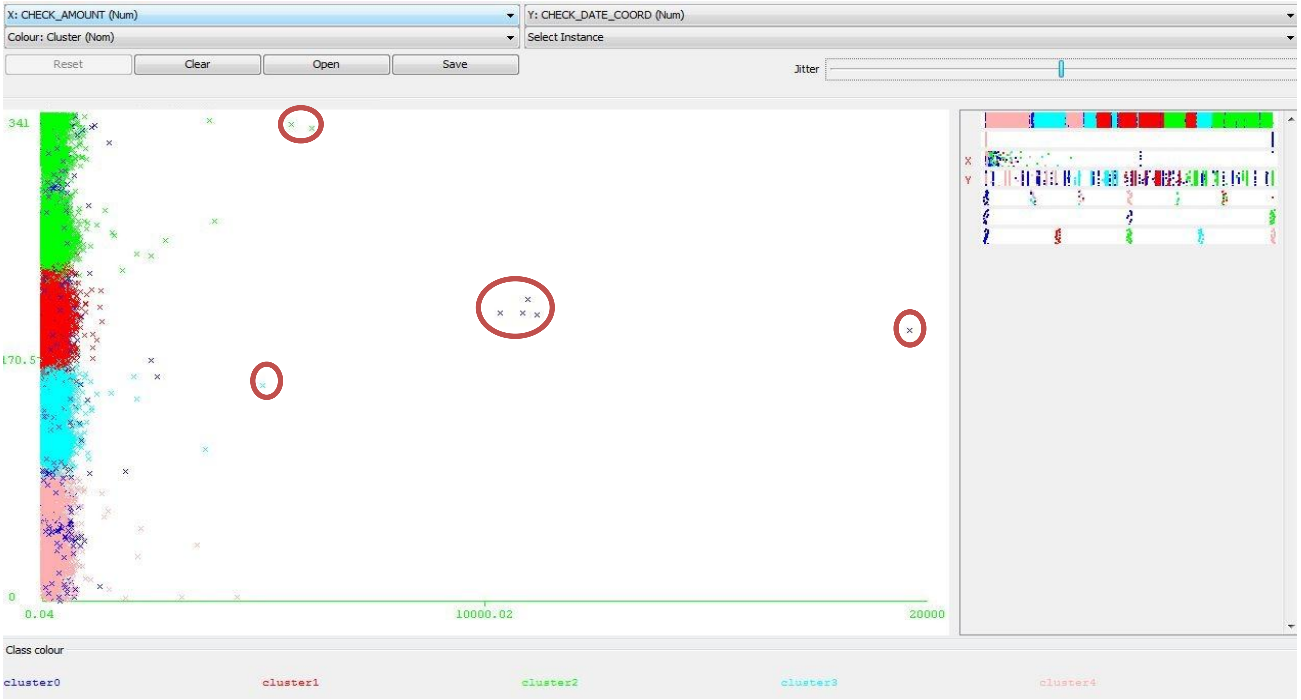
The distance metric used was Euclidean distance, which is typically used in clustering situations. In order to determine the best number of clusters, the algorithm is run independently for several values of K, and the one that yields the best partition is chosen. As mentioned before, the objective is to minimize the sum of squared errors (SSE). As expected, the higher the value of K, the lower the SSE; however after a certain point the decrease in SSE becomes insignificant, and any increase in the number of clusters (K) would be unnecessary. Table 2 shows the results of running the algorithms with K value from two to six.

Table 2. Different K values comparison

K	SSE	Iterations	C1	C2	C3	C4	C5	C6
2	1778	6	47%	53%				
3	1626	5	33%	37%	29%			
4	1567	4	24%	25%	28%	23%		
5	733	6	3%	24%	27%	23%	22%	
6	705	7	2%	24%	27%	23%	22%	1%

As we can see, the decrease in SSE was mostly pronounced for a value of K equal to 5. This value yielded 5 clusters, split into 3%, 24%, 27%, 23%, and 22%. It took the algorithm 6 iterations in total to reach these results.

Figure 5 is a graph visualizing the results obtained from running simple K-Means with (K=5) using Weka. The instances circled in red are examples of outliers (just a sample). This implies that these records do not conform to the general pattern (or normal pattern) of the records in this dataset.



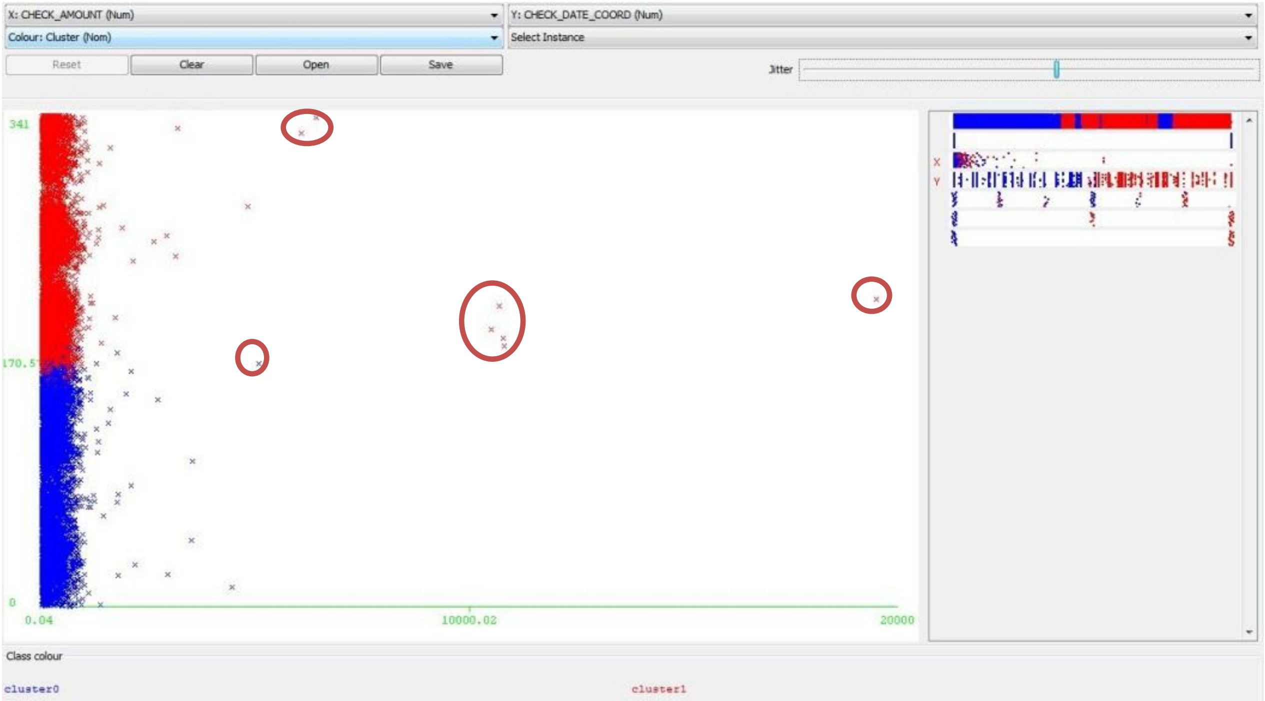
X: CHECK_AMOUNT (Num)

Y: CHECK_DATE_COORD (Num)

Colour: Cluster (Nom)

Figure 5. Results forK=5

It is noteworthy mentioning that the results obtained when K was set to 2 and 4 yielded similar results as the value 5. The circled instances in Figure 6 and 7 are the same ones that show in Figure 5

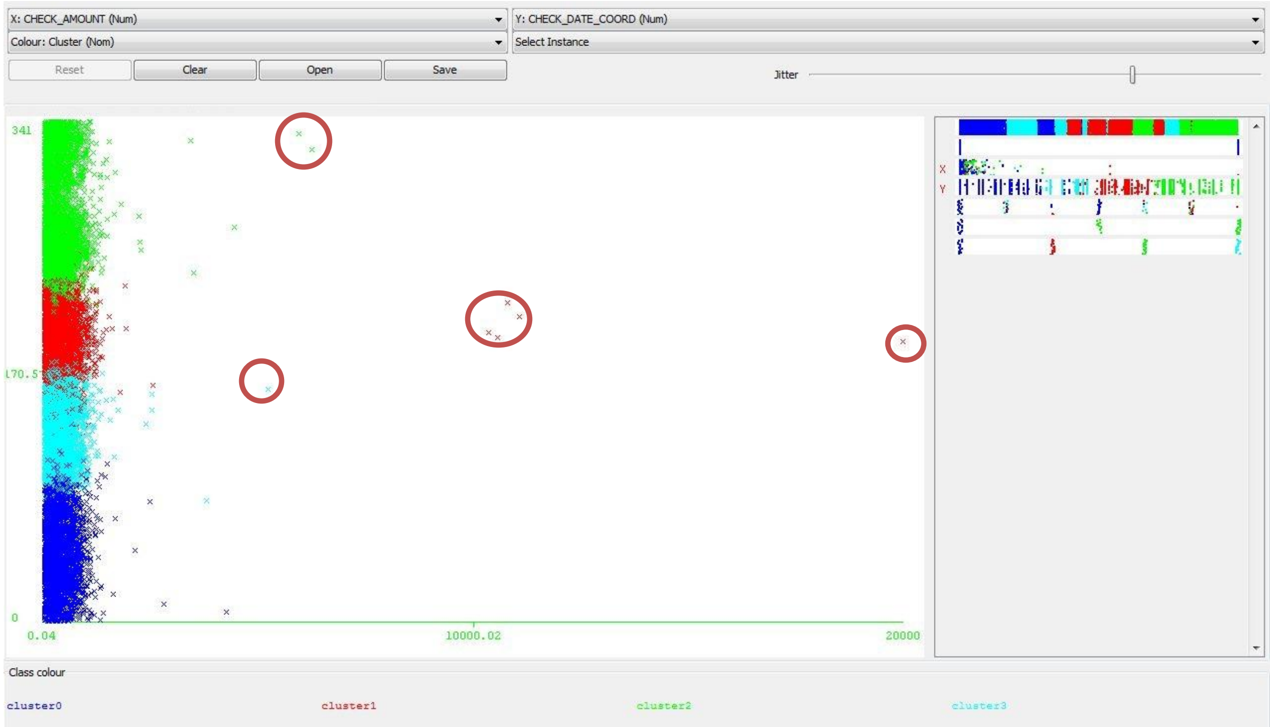


X: CHECK_AMOUNT (Num)

Y: CHECK_DATE_COORD (Num)

Colour: Cluster (Nom)

Figure 6. Results for K=2



X: CHECK_AMOUNT (Num)

Y: CHECK_DATE_COORD (Num)

Colour: Cluster (Nom)

Select Instance

Figure 7. Results for K=4

That being said, it is necessary to further examine these suspicious records by someone from the telecommunication company.

7. CONCLUSION

Anomaly detection techniques can be applied to several domains. The use of such techniques to identify fraudulent activities is particularly important in the business area. The concept of anomaly detection is to define a normal behavior and consider any instance that does not conform to it as anomalous.

There is no one superior technique to detect anomalies. The choice of the optimal technique depends on many factors related to the problem at hand, and therefore can be thought of as domain specific.

Another important factor dictating which technique to choose is the type of data (especially the availability of labeled instances). That would determine whether the mode of operation would be supervised, semi-supervised, or unsupervised.

Some of the major criticisms of data mining techniques in general are the scarcity of real-world datasets, especially labeled ones. Another criticism is the lack of published research proposing well researched methods and techniques. This paper uses an unlabeled real-business dataset. The company feedback will be used to further investigate the suspicious outliers captured by the model.

This study can be extended in several ways. Other clustering algorithms, for example DBSCAN, can be used on the dataset, and their results can be compared to the simple K-Means results from this paper. Another way to extend the this experiment is to apply cluster analysis to other types of data, such as insurance claims, and compare the performance of different clustering algorithms on these datasets. It would be interesting to compare the performance of clustering and classification algorithms on a dataset, given the possibility of procuring labeled instances.

References

Abe, N., Zadrozny, B., & Langford, J. (2006). Outlier detection by active learning. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (p. 504–509). ACM. Retrieved December 21, 2010, from http://portal.acm.org/citation.cfm?id=1150459.

Baker, L. D., Hofmann, T., Mccallum, A. K., & Yang, Y. (1999). A hierarchical probabilistic model for novelty detection in text. Citeseer. Retrieved December 22, 2010, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.6954.

Brause, R., Langsdorf, T., & Hepp, M. (2002). Neural data mining for credit card fraud detection. *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on* (p. 103–106). IEEE. Retrieved December 22, 2010, from http://ieeexplore.ieee.org/iel5/6582/17565/00809773.pdf?arnumber=809773.

Brockett, P. L., Derrig, R. A., Golden, L. L., Levine, A., & Alpert, M. (2002). Fraud classification using principal component analysis of RIDITs. *Journal of Risk and Insurance*, *69*(3), 341–371. Wiley Online Library. Retrieved December 21, 2010, from http://onlinelibrary.wiley.com/doi/10.1111/1539-6975.00027/full.

Brockett, P. L., Xia, X., & Derrig, R. A. (1998). Using Kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. *Journal of Risk and Insurance*, 245–274. JSTOR. Retrieved December 21, 2010, from http://www.jstor.org/stable/253535.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 1–58. ACM. Retrieved November 23, 2010, from http://portal.acm.org/citation.cfm?id=1541882.

De Stefano, C., Sansone, C., & Vento, M. (2000). To reject or not to reject: that is the question-an answer in case of neural classifiers. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *30*(1), 84–94. IEEE. Retrieved December 22, 2010, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=827457.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (Vol. 2). Citeseer. Retrieved December 22, 2010, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.133.1318&amp;rep=rep1&amp;type=pdf.

Edgeworth, F. (1887). On discordant observations. *Philosophical Magazine*, *23*(5), 364–375.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., & Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Applications of Data Mining in Computer Security*. Citeseer. Retrieved December 22, 2010, from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.119.5533.

Fawcett, T., & Provost, F. (1999). Activity monitoring: Noticing interesting changes in behavior. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (p. 62). ACM. Retrieved December 21, 2010, from http://portal.acm.org/citation.cfm?id=312129.312195.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, 31*(8), 651–666. Elsevier. doi: 10.1016/j.patrec.2009.09.011.

King, S., King, D., Astley, K., Tarassenko, L., Hayton, P., & Utete, S. (2002). The use of novelty detection techniques for monitoring high-integrity plant. *Control Applications, 2002. Proceedings of the 2002 International Conference on* (Vol. 1, p. 221–226). IEEE. Retrieved December 22, 2010, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1040189.

Kou, Y., Lu, C. T., & Chen, D. (2006). Spatial weighted outlier detection. *Proceedings of SIAM Conference on Data Mining*. Citeseer. Retrieved December 21, 2010, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.131.5607&amp;rep=rep1&amp;type=pdf.

Lee, W., & Stolfo, S. J. (1998). Data mining approaches for intrusion detection. *Proceedings of the 7th conference on USENIX Security Symposium-Volume 7* (p. 6). USENIX Association. Retrieved December 22, 2010, from http://portal.acm.org/citation.cfm?id=1267549.1267555.

others. (2005). *Introduction to data mining*. Pearson Addison Wesley Boston. Retrieved December 21, 2010, from http://www.pphust.cn/uploadfiles/200912/20091204204805761.pdf.

Phua, C., Lee, V., Smith, K., & Gayler, R. (2005). A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review*, 1–14. Retrieved November 26, 2010, from http://arxiv.org/pdf/1009.6119.

Salvador, S., Chan, P., & Brodie, J. (2004). Learning states and rules for time series anomaly detection. *Proc. 17th Intl. FLAIRS Conf* (p. 300–305). Retrieved December 21, 2010, from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:learning+states+and+rules+for+time-series+anomaly+detection#0.

Shekhar, S., Lu, C. T., & Zhang, P. (2002). Detecting graph-based spatial outliers. *Intelligent Data Analysis, 6*(5), 451–468. IOS Press. Retrieved December 21, 2010, from http://iospress.metapress.com/index/r0y19dqvk0vw4yc5.pdf.

Song, X., Wu, M., Jermaine, C., & Ranka, S. (2007). Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 631–645. Published by the IEEE Computer Society. Retrieved December 21, 2010, from http://www.computer.org/portal/web/csdl/doi/10.1109/TKDE.2007.1009.

Valdes, A., & Skinner, K. (2000). Adaptive, model-based monitoring for cyber attack detection. *Recent Advances in Intrusion Detection* (p. 80–93). Springer. Retrieved December 22, 2010, from http://www.springerlink.com/index/A2UYA9GQLQJK442U.pdf.

Appendix A

Table 3. Variables

Variable	Used/Omitted	Reason
BATCH_NAME	Omitted	Irrelevant
ADDRESS_LINE2	Omitted	Irrelevant
ADDRESS_LINE3	Omitted	Irrelevant
ADDRESS_LINES_ALT	Omitted	Irrelevant
ASSETT_CATEGORY	Omitted	Irrelevant
BANK_ACCOUNT_NAME	Omitted	Irrelevant
BATCH_DATE	Omitted	Irrelevant
CHECK_DESCRIPTION	Omitted	Irrelevant
CHECK_NUMBER	Omitted	Irrelevant
CHECK_STATUS	Omitted	Irrelevant
DIST_DESCRIPTION	Omitted	Irrelevant
DISTRIBUTION_ACCT_DESCRIPTION	Omitted	Irrelevant
DISTRIBUTION_LINE_NUMBER	Omitted	Irrelevant
F52	Omitted	Irrelevant
QUANTITY_INVOICED	Omitted	Irrelevant
SEGMENT1	Omitted	Irrelevant
SEGMENT2	Omitted	Irrelevant
SEGMENT3	Omitted	Irrelevant
SEGMENT4	Omitted	Irrelevant
SEGMENT5	Omitted	Irrelevant
SEGMENT6	Omitted	Irrelevant
SEGMENT7	Omitted	Irrelevant
SEGMENT8	Omitted	Irrelevant
SEGMENT9	Omitted	Irrelevant
TERM_NAME	Omitted	Irrelevant
VENDOR_PAY_GROUP	Omitted	Irrelevant
VENDOR_TYPE	Omitted	Irrelevant
DIST_UPDATE_BY,	Omitted	Needed transformation
AMOUNT_PAID	Omitted	Redundant
APPLIED_DIST_AMOUNT	Omitted	Redundant
APPLIED_INVOICE_AMOUNT	Omitted	Redundant
DIST_CREATED_BY_NAME	Omitted	Redundant
DIST_CREATION_DATE	Omitted	Redundant
DIST_GL_DATE	Omitted	Redundant
DIST_GL_PERIOD	Omitted	Redundant
DIST_UPDATE_BY_NAME	Omitted	Redundant
INVOICE_AMOUNT	Omitted	Redundant
INVOICE_DATE	Omitted	Redundant
ORG_NAME	Omitted	Redundant
PAYMENT_GL_DATE	Omitted	Redundant
PAYMENT_GL_PERIOD	Omitted	Redundant
VENDOR_NAME	Omitted	Redundant
ADDRESS_LINE1	Omitted	Unusable format/missing info
BANK_ACCOUNT_NUM	Omitted	Unusable format/missing info
INVOICE_DESCRIPTION	Omitted	Unusable format/missing info
INVOICE_ID	Omitted	Unusable format/missing info
CHECK_AMOUNT	Used	Used in the Study
ORG_ID	Used	Used in the Study
CHECK_DATE_COORD	Used	New variables- created by transforming an existing one
DIST_CREATED_DISCR	Used	New variables- created by transforming an existing one
DIST_UPD_DISCR	Used	New variables- created by transforming an existing one