American Academic Scientific Research Journal for Engineering, Technology, and Sciences

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

Detection of Rare Events: Cluster Based Preprocessing of the Training Set: The Case on Complaints for Invoice Time Series

Huseyin Carpanali [a]*, Ayse Humeyra Bilge [b], Arif Selcuk Ogrenci [c],Tarkan Ozmen [d], Ayse Tosun [e] , Kubra Cakar [f]

[a,b,c,d,e,f]*Kadir Has University, Istanbul 34083, Turkey*

[a]*Email: huseyin.carpanali@khas.edu.tr*,[b]*Email: ayse.bilge@khas.edu.tr*,[c]*Email: ogrenci@khas.edu.tr*

[d] *Email: tarkan.ozmen@gmail.com*,[e]*Email: aysetosun2018@gmail.com*,[f]*Email: kc.kubracakar@gmail.com*

Abstract

Detection of rare events is a major problem when dealing with unbalanced data. In the application of machine learning tools, data is split into training and test samples and preprocessing is applied to the training set, with the aim of obtaining a more balanced sample. In this paper we discuss preprocessing methods applied to heterogenous data clustered with respect to expected anomaly types. We propose a method for deciding on oversampling and under-sampling from each cluster, based on the variability of the items in each cluster, using Principal Component Analysis. The method is applied to the problem of detecting anomalies in a time series invoices, with an average rate of complaints of orders 10^{-4}.

Keywords: Unbalanced data; majority class; hierarchical clustering; heuristics.

1. Introduction

In many instances rare events have serious consequences. For example, rare diseases are usually deadly, and credit card frauds are rare but costly. The problem of detecting rare events is a binary classification problem; the population consists of two groups, the rare events, denoted as “positive” cases that constitute the “minority” group and “negative” cases that constitute the “majority” group. Various machine learning techniques can be applied to the binary classification problem, resulting in either correct or incorrect identification of the group an individual belongs to. As a result, one has four groups, consisting of “True-Positive”, “True-Negative”, “False-Positive” and “False-Negative” cases.

Received: 1/19/2024
Accepted: 3/19/2024
Published: 3/29/2024

----

* Corresponding author.

The performance of the machine learning algorithm is measured by “sensitivity” which is the ratio of the number of “True-Positive” cases to the total number of “Positive” cases and, by “specificity” which is the ratio of the number of “True-Negative” cases to total number of “Negative” cases. The principle of machine learning algorithms is to “train” the algorithm on a sample consisting of items with known characteristics. Intuitively, for “successful learning”, the algorithm has to be trained with a sufficient number of examples of positive and negative items. In the case of the detection of rare events, the number of positive items may not be adequate for efficient learning and “oversampling” of the minority class in the training set, may be used to increase the performance of the algorithm. In this respect, oversampling is the generation of new, “synthetic” items that have the same characteristics as the items in the minority class. The SMOTE algorithm is such an example that has proved to be useful in many applications.

If the ratio of the number of items in the minority class to the number of items in the majority class is too low, oversampling of the minority class may be insufficient to lead to a reasonably balanced training set. In that case, under-sampling of the majority class in the training set can be used to increase the balance. Random under-sampling is commonly used and gives quite successful results. In fact, if the majority class is homogeneous, random under-sampling is intuitively reasonable. In the present work we discuss the under-sampling of majority classes that display heterogenous characteristics. In the case of heterogeneous data, samples are first clustered into homogeneous subsets, with respect to the event that is aimed to be detected. If the items in a cluster are similar to each other, i.e., the variance of the set is low, then random under-sampling is applied. On the other hand, if the cluster has high variance, under-sampling should be avoided. Finally, if a cluster has too few elements, then oversampling can be applied.

The method we propose is applied to the case of the detection of customer complaints to monthly invoices of a major telecommunication company. The company's customers are divided into two categories, corporate and individual. After receiving their invoices, customers may issue a complaint either about a specific item of their invoice or the whole. A complaint might result in different scenarios according to the content. These complaints not only cause customer dissatisfaction but also cost serious amounts of money to the company due to the operational labor needed. Various predictive models are being used, and algorithms are being developed to minimize the number of complaints by predicting them beforehand. The prediction of a complaint will be helpful in preventing it by a possible correction or by placing a flag of notice which can be utilized in the case of a complaint. The monthly complaint rate for 18 million individual customers is about 1300 a month, which can be considered an extremely rare event.

When we consider complaints as positives and non-complaints as negatives, conventional predictive models won’t work properly since positives and negatives are disproportional. The models tend to label all customers as negatives to minimize the error. There are a couple of ways to approach this problem. Algorithms that detect anomalies, like Isolation Forest, can be used as a remedy. Another solution is to utilize oversampling and under-sampling techniques to create a balanced training set for predictive algorithms. Oversampling is used to increase the number of minority class data by synthesizing new data using various algorithms. A famous algorithm called SMOTE (Synthetic Minority Oversampling Technique) can be used for oversampling purposes. Under-sampling is a process where the number of observations from the majority class is reduced to a reasonable size, either by using random sampling or a systematic approach. While being the easiest to apply, random sampling might sometimes create samples that do not represent the population properly.

The dataset, which will be explained thoroughly in the following sections, consists of 9 months of invoice amount and complaint information for each customer. There are plenty of customers whose invoice sequences are similar. Those customers generally have a flat sequence of invoices. Having multiple observations similar to each other does not add any valuable information to the predictive models. Instead, it increases the run-time of the programs and creates imbalanced datasets. In order to solve this issue, the number of these observations should be trimmed to a reasonable amount. It would also be very beneficial if the data could be clustered into segments where customers with complaints would be concentrated in some segments. Unfortunately, traditional clustering techniques such as k-means and SOM did not work to obtain a separation of these two classes. In order to apply further techniques for supervised learning, we decided to cluster data into segments of similar behavior in a hierarchical manner. This can be identified as a “pre-processing” phase of large imbalanced data. The rest of this paper is organized as follows: Section 2 will have an overview of the literature about dealing with imbalanced data. Our data will be explained in Section 3, and the methodology of clustering based on hierarchical decision levels will be introduced in Section 4. Presentation of the results is given in Section 5 and finally, Section 6 will conclude the paper with the discussion of the results.

2. Literature Survey

Problems caused by imbalanced data are being faced by scientists in real-world applications such as medical diagnosis, financial crisis prediction, and e-mail filtering [1]. Another interesting topic that is affected by imbalanced data is the development of real-world drug applications [2]. Furthermore, the minority (or rare) class is typically the main class of interest in the data mining task. Learning algorithms or developed models may get overpowered by the dominant class and disregard the minority class in an attempt to reduce error, if the class imbalance issue is not considered. This led to a significant amount of research on this topic.

There are multiple ways to approach the imbalanced data problems. These methods can be separated into four categories: algorithmic-level methods, data-level methods, cost-sensitive methods, and ensembles of classifiers [2,3]. Cost-sensitive methods are used by giving a cost for misclassification. Data-level methods like under-sampling and over-sampling are widely used in literature. These methods focus on preprocessing the imbalanced dataset into a balanced dataset using various algorithms in order to train the classifier with a better data set. Furthermore, a comparison evaluation of multiple popular methodologies by Galar and his colleagues. [1] found that combinations of classifier ensembles and data preprocessing methods outperform other approaches.

SMOTE, introduced by Chawla [4], has dominated the over-sampling area by minority class synthesis. Azhar and his colleagues. [5] conducted an extensive investigation of SMOTE-based algorithms and concluded that SMOTE-based algorithms give a performance gain ranging up to 12% in classification performance compared to non-SMOTE implementations. Random under-sampling, a process where a sample is randomly selected from the majority class, is generally used for under-sampling processes. The problem with random under-sampling is the possibility of information loss caused by the arbitrary nature of the process. One way to overcome this problem is to use repetitive undersampling techniques [6]. Hasanin and Khoshgoftaar [7] conducted several simulations to test the effect of random undersampling on imbalanced datasets. They tested model powers for datasets with 100%, 10%, 1%, 0.1%, 0.01%, and 0.001% positive percentages. They showed that when class distribution is below a positive percentage of 0.1%, models tend to perform worse. They also showed that a performance boost can be achieved by partial undersampling. Seiffert and his colleagues. [8] proposed RUSBoost, combining the random under-sampling approach with a boosting procedure. Another approach, UB (UnderBagging), combines a bagging technique with a random under-sampling process. The majority class was under-sampled in Barandela and his colleagues. [9], the first study to use UnderBagging, and a balanced training data set was then utilized to build a bagging-based k-nearest neighbor ensemble (k = 1).

For credit card fraud detection, which is a widely researched imbalanced problem, Randhawa et. al [10] used Adaboost and Majority Voting to test twelve different classifiers. The results showed that the Majority Voting method gave a good performance. Zareapoor and his colleagues. [11] compared different classifiers like Naïve Bayes, K-Nearest Neighbors, Support Vector Machine, and Bagging. The researchers did not perform any under-sampling or over-sampling methods, but they proposed a different approach to measure the model. Furthermore, Awoyemi and his colleagues. [12] utilized a hybrid sampling method and tested out KNN, Naïve Bayes, and logistic regression models. The literature suggests that there is no single best method for a successful classification. As usual, the performance of the method employed may vary in different data sets. As the direct application of any classification may cause problems in a dataset with different characteristics, this work aimed to cluster the original data into similar segments which will be later processed for classification. To the best of our knowledge, the literature does not include a similar approach which will be described in the methodology section.

3. Description of Data

In this study, we worked with 2 363 378 telecommunication customers who didn't complain and 10 760 customers who complained at least once during the observation period. Data consists of a series of 9 invoices over the period 07/2021-03/2022. The numbers of invoices for each month and complaints during these months are given in Table 1. We note that objections to the invoice of a given month can be filed at a later month, the figures given in Table 1 reflect complaints to the invoice of the given month.

Table 1: Distribution of complaints (NC: No Complaint, C: Complaint

Month	1	2	3	4	5	6	7	8	9
NC	2 375 072	2 375 153	2 375 120	2 375 100	2 375 136	2 375 351	2 375 438	2 375 497	2 375 351
C	1 444	1 363	1 396	1 416	1 380	1 165	1 078	1 019	1 165

The data consists of 19 columns. These columns are,

- 1st Column: Unique ID of the customer
- 2nd – 10th Column: Invoice amounts for each month in TL (Turkish currency)

- 11th – 19th Column: Complaint information for each month (0 or 1)

In our previous work [13], we clustered 1.7 million corporate customers in order to detect retail expenditure trends, which are characterized by a sudden increase in otherwise stationary time series data. We used a hierarchical clustering approach, first splitting the data according to the mean of the invoice, then with respect to the relative standard deviation, and lastly, with respect to the relative range of monthly expenses. Based on the attributes of the data, a threshold value was used to divide each feature into four classes. In our situation, the strategy we used produced 4^3=64 groups, some of which were empty. The groupings that reflected retail expenditure were those with a high range and a low standard deviation. Clusters are labeled by a 3-digit number ijk, where i, j, and k denote respectively the levels of the mean, relative standard deviation and relative range, each ranging from 1 to 4. Certain clusters are excluded by the fact that relative range and relative standard deviation are not independent. The list of nonempty clusters, with the number of complaints and no-complaints are given below. We note that the number of complaints given in the table are the number of customers that complained at least once during the 9-month period. Thus, the average number of constraints is of the orders 10^{-4} as noted above.

Table 2: Hierarchical clusters [13] of the data (C: Complaint, NC: No complaint)

Cluster	**NC**	**C**	**Ratio**		**Cluster**	**NC**	**C**	**Ratio**
111	394764	446	0.0011		211	314995	936	0.0029
112	130701	532	0.0040		212	161830	1110	0.0068
113	30877	169	0.0054		213	55673	573	0.0102
123	8403	34	0.0040		223	18468	189	0.0123
124	3619	43	0.0118		224	12165	121	0.0099
134	527	13	0.0246		234	4541	50	0.0110
144	12	0	0.0000		244	4	0	0.0000
Total	568903	1237	0.0021		Total	567676	2979	0.0052
Cluster	**NC**	**C**	**Ratio**		**Cluster**	**NC**	**C**	**Ratio**
311	220682	928	0.0042		411	33787	181	0.0053
312	142565	1689	0.0118		412	43591	369	0.0084
313	76824	958	0.0124		413	38126	343	0.0089
322	19	0	0.0000		422	16	0	0.0000
323	68624	599	0.0087		423	39637	390	0.0098
324	36989	483	0.0130		424	12501	176	0.0140
333	3	0	0.0000		433	20	0	0.0000
334	19322	191	0.0098		434	19258	215	0.0111
344	639	1	0.0015		444	1521	21	0.0138
Total	565667	4849	0.0085		Total	188457	1695	0.0089

From Table 2, we can observe that clusters with low relative standard deviation and low relative range are less likely to complain than clusters with higher ones. For example, while cluster 411 has a C/NC ratio of 0.0054, cluster 423 has a C/NC ratio of 0.0098, and cluster 434 has a C/NC ratio of 0.0111. We also know that clusters like 1YZ (111,112 etc.) are less likely to have a complaint since the mean of the invoices is too low. Furthermore, as clustering was based on value ranges of the features hence the number of items in each cluster show a great variety. The mean, relative standard deviation and relative range, as basic features of the clustering algorithm of the reference [10], are proved to be useful for the analysis of customer complaints, but for the purposes of preprocessing the training set for better performance of machine learning algorithms, we are rather interested in the variability of the items in each cluster. In the present work, we will use the range, standard deviation and mean features in a different format, to obtain a new clustering that will result in clusters that are homogeneous with respect to inter-variability. This will be based on Principal Component Analysis, as it will be discussed in the next section. The method that we are proposing will be under-sampling of overpopulated and low diversity clusters and oversampling of those clusters with very few elements and use moderately populated clusters with sufficient diversity as is.

4. Methodology

The clusters of data are created using characteristics such as mean, the ratio of standard deviation to the mean and the ratio of the range of the data to the mean, in a hierarchical manner as given in [13]. After forming the clusters based on appropriate thresholds, we utilized the PCA (Principal Component Analysis) to calculate the similarity between items in each cluster. This similarity will be utilized in subsequent training where we will select a certain number of observations from each cluster based on this similarity measure. We decided to select fewer observations from a cluster if that cluster has mostly similar observations. By doing so, we aim to achieve a better-represented majority class in the training set for predictive models. For this purpose, 20% of the majority class data is reserved as test set, and the remaining 1 890 703 invoices are saved to be used in this work. After discussions with the experts of the invoice and complaint management departments, we have decided to apply a hierarchical clustering scheme as described below. We will for clusters in a hierarchical way in 6 stages.

Stage 1: In the first stage we select invoices with low range, here determined as below 10TL and calculate the range (max-min) value of invoice amounts for each customer. The range is one of the main characteristics for determining anomalies in the invoice sequence. A low range indicates a stable invoice sequence, which may be considered to be alike, regardless of the mean value. On the other hand, a high range can indicate different customer behaviors. For this group low range, we form 10 clusters for customers with range values 1-10 TL. It has to be noted that the range feature focuses on the variation and not on the absolute value of the invoices. The similarity of the items in a given cluster is measured by Principal Component Analysis (PCA) as follows. We recall that, PCA is based on the eigenvalue structure of the covariance matrix of the members of a given set of say *m* features. In that case the covariance matrix *S* is an mxm symmetric matrix with real eigenvalues labeled in decreasing order as, $\lambda_1 > \lambda_2 \ldots > \lambda_m$. The corresponding eigenvectors are also labeled as $X_1, X_2, \ldots, X_m$. Let $\lambda_T = \lambda_1 + \lambda_2 + \ldots + \lambda_m$ be the sum of the eigenvalues. The ratio of the largest eigenvalue λ_1 to λ_T is a measure of the similarity of the items in the cluster, denoted as PCA_1. The structure of the clusters C_{01}-C_{10} with low range, obtained at the first stage are given in Table 3. As an example, C_{02} contains 35552 customers whose range of invoices is between 1 (inclusive) and 2 (exclusive). The total number of customers in C_{01} to C_{10} is 231995, constituting approximately 12% of the population. PCA_1 values indicate that the invoices in each cluster are similar to each other as expected. The customers who have been added to a cluster are removed from the dataset, and the hierarchical clustering will continue to the second layer.

Table 3: The structure of Clusters C_{01}-C_{10}. (clusters with low range)

Cluster	C_{01}	C_{02}	C_{03}	C_{04}	C_{05}	C_{06}	C_{07}	C_{08}	C_{09}	C_{10}
Interval (Range)	[0,1)	[1,2)	[2,3)	[3,4)	[4,5)	[5,6)	[6,7)	[7,8)	[8,9)	[9,10)
# of customers	308	35552	30901	16260	34728	15707	25513	17222	21345	34459
PCA_1	1.00	1.00	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Stage 2: In the second stage, we deal with invoice series whose range is greater than the threshold value 10 TL and we evaluate the ratio of the standard deviation to the mean of the invoice sequence. At this stage we select invoice sequences whose relative standard deviation is low, the threshold chosen as 0.20, ranging from 0 to 0.20 in steps of 0.05. The same steps are followed as in the first stage of the clustering process. A total of 827266 (44% of the population) customers are grouped in clusters C_{11}-C_{14}. Again, the PCA_1 values indicate a successful similarity grouping in each cluster.

Table 4: The structure of Clusters C_{11}-C_{14} (clusters with low standard deviation)

Cluster	C_{11}	C_{12}	C_{13}	C_{14}
Interval (normalized standard deviation)	[0,0.05)	[0.05,0.1)	[0.1,0.15)	[0.15,0.2)
# of customers	48439	268619	293550	216658
PCA_1	0.99	0.98	0.97	0.95

Stage 3: At the third stage we use the maximum invoice amount as the clustering criteria. We select two groups consisting of very low and very high maximum invoice amounts. For this purpose, C_{15} is formed for customers having a maximum invoice amount less than 50 TL, and C_{16} is formed for customers above 1000 TL. Table 5 indicates that approximately 2.7% of the population is grouped into those two clusters. It has to be noted that they do not include customers who already have been clustered in the previous two layers. The PCA_1 values indicate that the similarity score within those clusters is relatively low hence these clusters should be densely sampled.

Table 5: The structure of Clusters C_{15}-C_{16}. (clusters consisting of outliers)

Cluster	C_{15}	C16
Interval (MAX)	[0,50)	(1000,)
# of customers	4,840	45,866
PCA_1	0.60	0.29

Stage 4: At this stage we select customers with an extremely high invoice amount for only one month and a low

range for the rest. Invoice sequences of this type correspond most likely to retail expenditures, as discussed in [10]. Let f_i be the invoice amount for the month i and let $M = max(f_1, f_2 \dots f_9)$. Clusters C_{17} and C_{25} are determined as follows. For each month i, we consider invoice sequences that satisfy the condition,

$$f_i = M > 0.1M > mean(f_1, f_2, ..\hat{}f\_i, \dots, f_9)$$

where $\hat{}f\_i$ means that the i'th term is omitted. Invoice sequences satisfying this condition are the ones that have a high invoice at month i and they are assigned to cluster C_{17}-C_{25}. For example, C_{17} is determined by the condition $f_1 = M > 0.1M > mean(f_2, f_3 \dots f_9)$ while C_{18} corresponds to the condition $f_2 = M > 0.1M > mean(f_1, f_3 \dots f_9)$. Customers from each of these clusters have peak in their invoice sequence at a given month and the mean of the remaining months is less than 10% of this peak value. The distribution of customers into those clusters is given in Table 6. It can be seen that the similarity measure of PCA_1 for those clusters is higher than 0.84 however a relatively low proportion of the population is clustered at this layer, namely 0.1%.

Table 6: Structure of Clusters C_{17}-C_{25} (clusters with extremely high invoice at a single month)

Cluster	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}
Peak month	1	2	3	4	5	6	7	8	9
# of customers	264	131	147	122	134	117	120	157	824
PCA_1	0.94	0.90	0.90	0.88	0.89	0.88	0.90	0.84	0.92

Stage 5: The same process is repeated with a looser criterion at the next layer. For cluster 26, the equation is

$$f_1 = M > 0.2M > mean(f_2, f_3 \dots f_9)$$

It is repeated for each month, and the distribution can be seen in Table 7 which indicates that approximately 9% of the population is clustered with a high level of similarity at this layer.

Table 7: The structure of Clusters C_{26}-C_{34} (clusters with moderately high invoice at a single month)

Cluster	C_{26}	C_{27}	C_{28}	C_{29}	C_{30}	C_{31}	C_{32}	C_{33}	C_{34}
Peak month	1	2	3	4	5	6	7	8	9
#of customers	24812	22588	16755	11714	16245	14865	14617	18442	26364
PCA_1	0.82	0.85	0.84	0.84	0.86	0.85	0.84	0.84	0.82

Stage 6: At the final stage we distinguish invoice series that contain a plateau and the ones that are scattered across their range. For this purpose, the interval between the maximum and the minimum of the invoice amounts is divided into three sectors. The first sector is the lower 20% of the interval, the second sector is the middle 60% of the interval, and the third sector is the top 20%. After that, the number of invoices in each sector is counted. The number of invoices in the middle sector is used as a clustering criterion. Table 8 displays the distribution of invoices that fall in the middle sector. and it indicates that the remaining 32% of the data cannot be clustered with

a high level of similarity as in the previous layers. C_{35} indicates that approximately 5% of customers have either "very low" or "very high" invoice amounts.

Table 8: The structure of Clusters C_{35}-C_{42} (clusters with/without a plateau)

Cluster	C_{35}	C_{36}	C_{37}	C_{38}	C_{39}	C_{40}	C_{41}	C_{42}
# invoices in middle sector	0	1	2	3	4	5	6	7
# of customers	102081	102879	101133	97002	84691	65028	41089	18415
PCA_1	0.50	0.56	0.59	0.64	0.68	0.72	0.75	0.79

5. Discussion of the Results

In order to observe cluster structures, graphs of invoice series are plotted for a selection of the clusters. First of all, randomly selected 1000 elements of Cluster 2 are plotted in Figure 1.

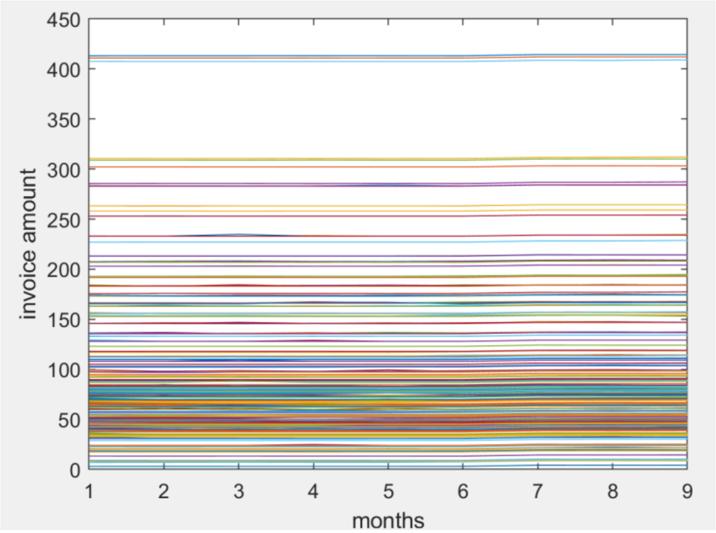

Figure 1: Invoice sequences of 1000 randomly selected elements of Cluster 2

The low-range clusters like this obviously have almost constant invoice sequences. Detection of these kinds of sequences is important because these generally can't be associated with customer complaints. If a customer gets almost the same amount in their invoices for each month, they won't have any reason to issue a complaint.

The last low-range cluster, Cluster C_{10}, includes customers with range values between 9 and 10 as seen in Figure 2. Even though fluctuations can be observed on the graph, these still can be considered stable invoice sequences.

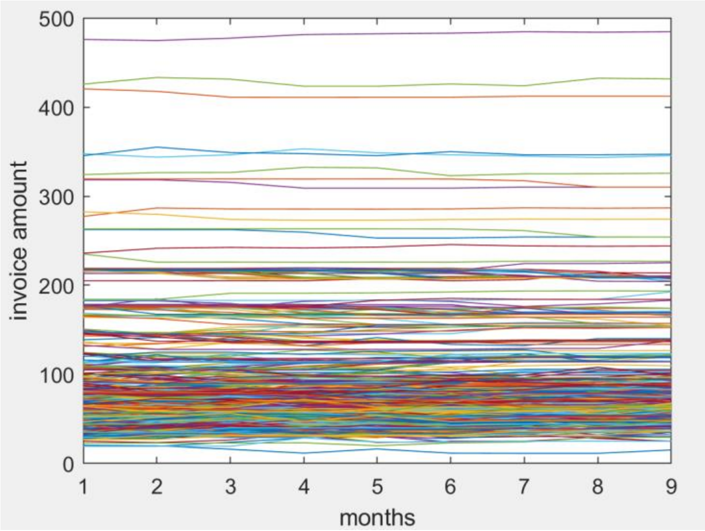

Figure 2: Invoice sequences of 1000 randomly selected elements of Cluster 10

Clusters C_{11}-C_{14} also capture somewhat stable invoice series; these clusters can also be considered low complaint risk areas of the dataset. As an example, cluster C_{11} is displayed in Figure 3,

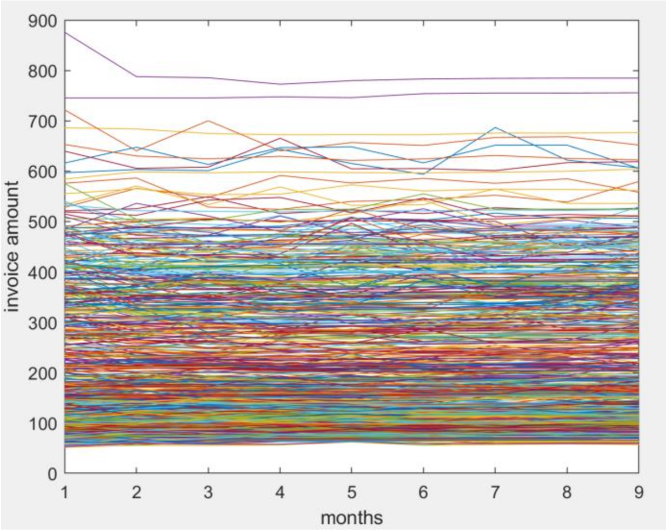

Figure 3: Invoice sequences of 1000 randomly selected elements of Cluster 11

Clusters C_{15} and C_{16} are created to select outliers. Cluster C_{15} captures extremely low mean customers, while Cluster C_{16} captures extremely high mean customers. Customers within these clusters don’t share any other characteristics than their mean values and we omit their graphs.

Clusters C_{17}-C_{25} are the realization of the invoice sequences that have one peak month and a low range for the rest of the months. The only difference between these clusters is the peak month. We expect the customers to issue a complaint at the month of peak invoice. Their distribution is displayed in Figures 4 and 5 for clusters C_{17} and C_{20}, respectively.

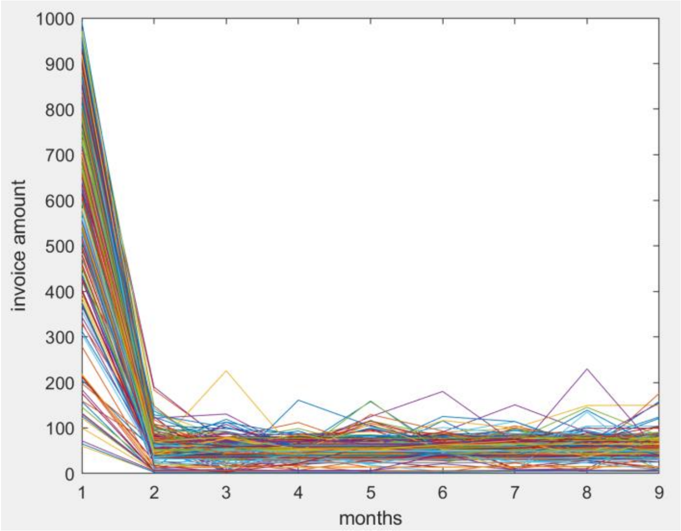

Figure 4: Invoice sequences of elements of Cluster 17

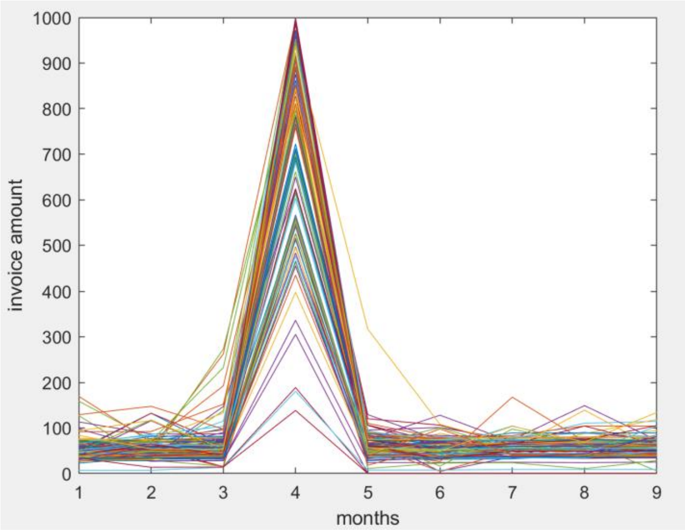

Figure 5: Invoice sequences of elements of Cluster 20

Clusters C_{26}-C_{34} are also constructed using the same logic. However, these clusters are less informative for determining whether there is a peak or not. As can be seen in Figure 6, sample customers may have relatively high values of invoices in the non-peak month.

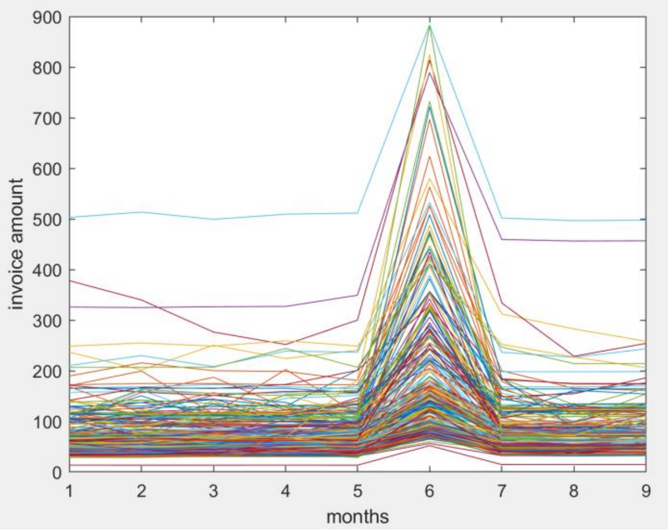

Figure 6: Invoice sequences of randomly selected 250 elements of Cluster 31

The same hierarchical clustering has been carried out for the customers who raised a complaint for their invoice. Those customers are distributed into the clusters in a similar fashion as the customers without complaints, that is we cannot identify certain clusters where the complaining customers are concentrated. This evidence is displayed in Figure 7 where it can be seen that clusters are not different among each other with respect to the complaint ratio except the first 10 clusters where there are very few complaints.

The final group of clusters, C_{35}-C_{42} have low homogeneity, as it can be seen from their low PCA_1 ratio and their graphs are omitted.

From the point of view of complaint detection, it is useful to investigate the distribution of complaints among clusters. In Figure 7, we present the percentages of complaints in each cluster.

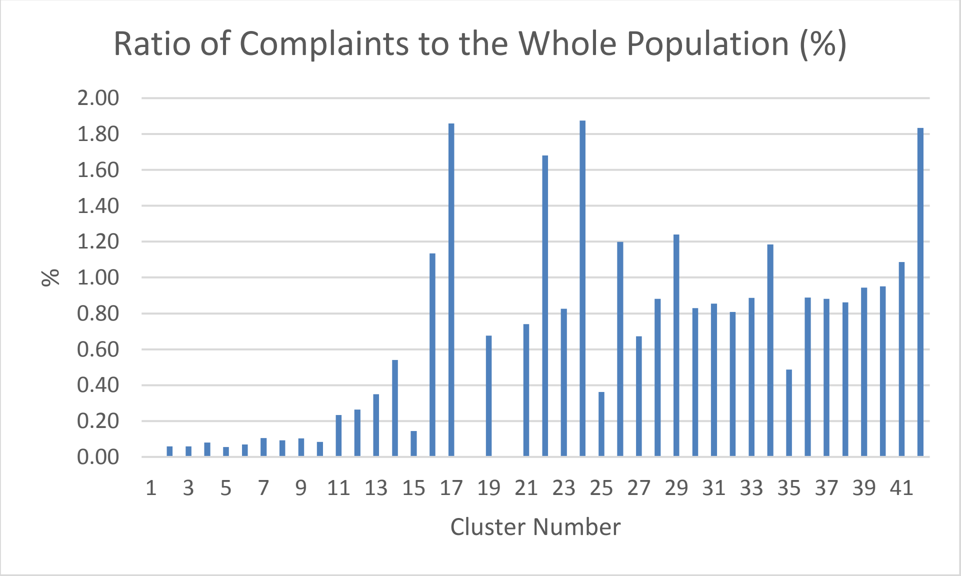

Figure 7: Ratio of complaining customers in clusters C_{01}-C_{42}

It can be seen that, as expected, invoice series with low variability, consisting of clusters C_{01}-C_{10} are less likely to lead to customer complaints. For the rest of the clusters, the ratio of the minority class is increased in general by an order of magnitude, leading possibly to an increase of the efficiency of machine learning algorithms.

For the purpose of comparison with the customers who did not issue a complaint, Figure 8 displays the distribution of customers into clusters for non-complaining versus complaining customers where each point represents one cluster.

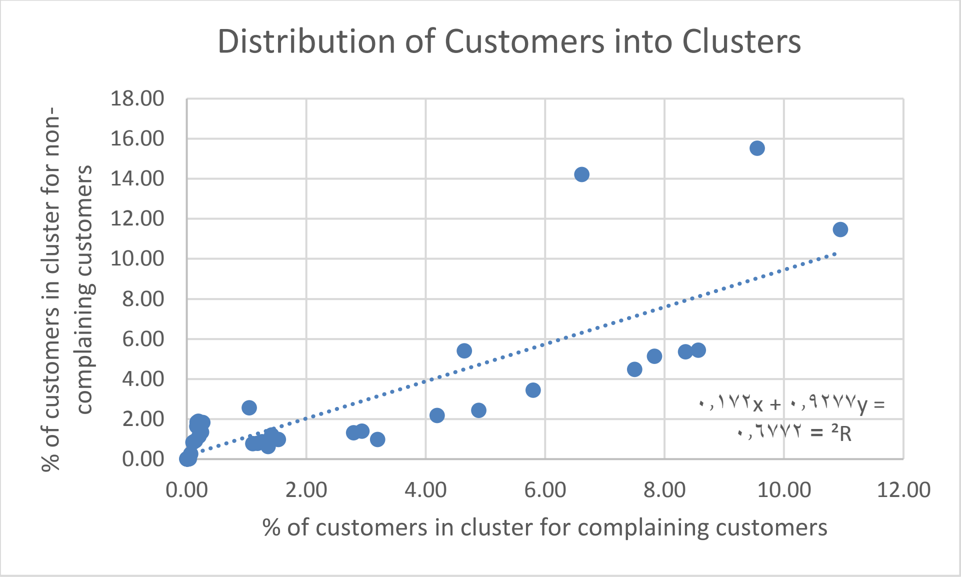

Figure 8: Distribution of customers into clusters: non-complaining versus complaining (each point represents one cluster of C_{01}-C_{42})

In Figure 8, a linear regression is also included which implies that a linear relationship seems to exist if the outliers are dismissed. Therefore, we can conclude that this hierarchical clustering approach based on heuristic supported by domain knowledge can create segments of higher similarity for both complaining and non-complaining customers.

The clustering scheme that has been introduced can be used for designing an appropriate preprocessing methodology for the majority class. Namely, clusters with low variability, characterized a-by a PCA1 ratio closed to 1 need not be densely sampled, while clusters with high variability should be densely sampled even oversampled if they have low population.

6. Concluding Remarks

In this work, we proposed a methodology for preprocessing imbalanced data with the aim of detecting rare events. We recall that in the application of all machine learning algorithms, data is split into training and test sets and one can preprocess the training set in order to improve the performance of the algorithm. With the purpose of detecting rare events, described in terms of minority and majority classes, oversampling of the minority class is a well-known and successful technique. On the other hand, preprocessing of the majority class is usually restricted to random under-sampling.

In this work we propose a decomposition of the majority group into subsets, based on the similarity of the items in each set. The similarity is determined by Principal Component Analysis, the similarity measure being the percentage of the variance explained by the largest eigenvalue of the covariance matrix, denoted as PCA_1.

This methodology is applied to the complaints to a series of invoices in telecommunication sector, where the events to be detected, i.e., the minority class is about 10^{-4} of the total population. The clustering of the training set is based on extracting homogeneous subsets, based on the range, the standard deviation, the mean and other application specific features of the data.

The preprocessing scheme that we propose is random under-sampling from each cluster at different selection rates. For example, fewer samples will be selected from clusters with higher PCA_1 values. As items in these clusters are very similar to each other, selecting too many of these will not add any new information to the models, but it will increase the computational power needed, also increasing the running time of the models. For the clusters with lower PCA_1 values, we propose selecting more observations to represent the variability of the items. Finally, if a cluster does not have enough observations to start with, oversampling algorithms like SMOTE can be used to enhance the number of observations. Application of the method proposed here and the investigation of its performance in terms specificity and sensitivity and run-time improvement is planned for future work.

References

[1] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, “A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches”, IEEE Trans. Syst. Man Cybern. – Part C, 42 (4), 463–484, 2012

[2] C. Beyan and R. Fisher, “Classifying imbalanced data sets using similarity-based hierarchical decomposition,” Pattern Recognition, vol. 48, no. 5, pp. 1653–1672, 2015.

[3] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst. Man Cybern. – Part C 42 (4) (2012) 463–484.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[5] Azhar, N.A., M.S.M. Pozi, A.M. Din, and A. Jatowt. “An Investigation of SMOTE Based Methods for Imbalanced Datasets with Data Complexity Analysis.” IEEE Transactions on Knowledge and Data Engineering, Knowledge and Data Engineering, IEEE Transactions on, IEEE Trans. Knowl. Data Eng 35, no. 7 (July 1, 2023): 6651–72. doi:10.1109/TKDE.2022.3179381.

[6] Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano. “An empirical comparison of repetitive

undersampling techniques.” In Information Reuse & Integration, 2009. IRI’09. IEEE International Conference

on, pages 29–34. IEEE, 2009.

[7] Hasanin, T., & Khoshgoftaar, T. (2018). “The Effects of Random Undersampling with Simulated Class Imbalance for Big Data.” 2018 IEEE International Conference on Information Reuse and Integration (IRI), Information Reuse and Integration (IRI), 2018 IEEE International Conference on, IRI, 70–79. https://icproxy.khas.edu.tr:2071/10.1109/IRI.2018.00018

[8] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, “RUSBoost: a hybrid approach to alleviating class imbalance”, IEEE Trans. Syst. Man Cybern. – Part A 40 (1), 185–197, 2010

[9] R. Barandela, R.M. Valdovinos, J.S. Sanchez, “New applications of ensembles of classifiers”, Pattern Anal. , Appl. 6 ,245–256, 2003.

[10] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, “Credit Card Fraud Detection Using AdaBoost and Majority Voting”, IEEE Access, Vol. 6, pp. 14277–14284, 2018.

[11] M. Zareapoor and P. Shamsolmoali, “Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier”, Procedia Computer Science, Vol. 48, pp. 679–685, 2015.

[12] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, “Credit card fraud detection using machine learning techniques: A comparative analysis”, In: Proc. of 2017 International Conference on Computing Networking and Informatics (ICCNI), pp. 1–9, 2017.

[13] Bilge, A. H. ., Ogrenci, A. S. ., Carpanali, H. ., Aktunc, E. A. ., Atas, F., Ozmen, T. ., & Kaya, B. E. . (2022). Detection of Expenditure Trends in the Telecommunication Sector. American Scientific Research Journal for Engineering, Technology, and Sciences, 90(1), 340–350.