iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** V **Month of publication:** May 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Outlier Detection Using Oversampling PCA for Credit Card Fraud Detection

Amruta D. Pawar[1], Seema A. Dongare[2], Amol L. Deokate[3], Harshal S. Sangle[4] , Panchsheela V. Mokal[5]
[1,2,3,4,5] *Computer Technology Department, Sanjivani K. B. P. Polytechnic, Kopargaon*

Abstract*: Credit card fraud detection is an important application of outlier detection in recent years. Many outlier detection techniques are available but they are working in batch mode, due to this those techniques are not applicable for applications where large amount of data is present. In this paper, a new credit card fraud detection method based on Oversampling Principal Component Analysis (osPCA) with low computation and memory requirements is presented. Principal Component Analysis (PCA) is a tool in data analysis for dimension reduction, it transforms high dimensional data into lower dimensions which contains maximum amount of information. Here we are using concept of oversampling on data and then input is given to PCA, therefore it is possible to detect credit card frauds from large amount of data. This technique is suitable for online applications which have memory or computation limitation.

Keywords*— Oversampling, Outlier, PCA.

I. INTRODUCTION

Outlier detection is a problem of finding patterns in data that do not satisfies the expected behaviour.These non- satisfying patterns are often referred to as anomalies or outliers. Outlier detection is a mechanism to find such outliers or anomalies in data. Outlier detection is used in wide variety of applications such as fraud detection for credit cards, insurance or health care, intrusion detection for cyber security, and fault detection in safety critical systems and so on.

In past many outlier detection methods have been proposed. These approaches can be divided into main three categories: Statistical, Distance and Density based methods. Statistical based approaches follow some predefined distributions and data which deviates from these standard distributions are considered as outliers. Distance- based approaches consider the distance of its neighbouring objects and objects far away from the current object is considered as an outlier. Density-based approaches calculate outlier factors to measure the outliereness of data instance. Fig. 1 illustrates outliers in two-dimensional dataset. The data has two normal regions N1 and N2 since most observations lie in these two regions. Points which are far away from these regions, for example, o1 and o2, and points in region o3 are outliers [3].

Outlier detection is an unsupervised data learning problem. It is observed that removing or adding abnormal data instance will affect the principal direction than removing or adding normal one does. Using this leave one out (LOO) strategy, one can calculate principal direction of the dataset without target data instance present. The outlierness of data instance can be determined as by the variation of resulting principal direction.

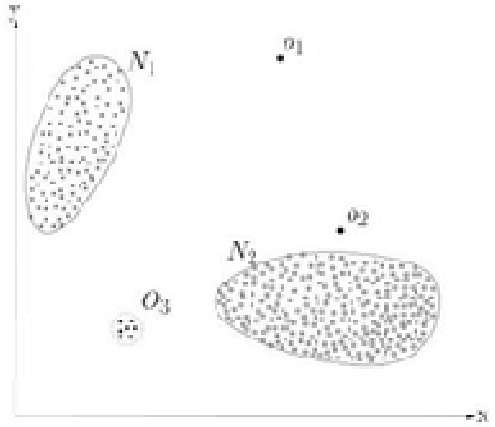


Fig. 1 A simple example of outliers in a two-dimensional data set.

©IJRASET: All Rights are Reserved

The difference between these two eigenvectors will indicate the outlier of the target instance. By ranking the scores of all data points, one can identify the outlier data by a predefined (dPCA) based approach for outlier detection. This framework is known as decremental PCA. This approach is well suited for small dataset [1].

In real-world applications such as credit card fraud, outlier detection techniques need to deal with large amount of data adding or removing one target instance only produces negligible differences in resulting eigenvectors, hence dPCA technique is not suitable. To solve this problem oversampling technique is used to duplicate the target instance and then apply oversampling PCA (osPCA) on such oversampled data. As a result, the effect of outlier instance is amplified due to its duplicates present in the principal component analysis which makes outlier detection easier. In previous approaches for each target instance there is a need to create a covariance matrix and solves the associated PCA problem. Therefore, here osPCA is proposed, which allows calculating the eigenvectors without storing data covariance matrix. Compared with other popular outlier detection algorithms, the required computational cost and memory requirements are significantly reduced, thus this method is suitable for credit card fraud detection [1].

II. LITERATURE SURVEY

In past many outlier detection techniques were proposed.

M. Breunig, H-P Kriegel, R. T. Ng and J. Sander proposed density-based Local Outlier Factor (LOF) method to find outliers. They assigned each object a degree of being an outlier. This degree is called as Local Outlier Factor (LOF) of an object. The degree depends on how object is isolated from surrounding objects. All previous methods consider outliers as binary property but this method assigns degree of being an outlier. However, this method does not work on high dimensional dataset and also requires high computation [2].

V. Chandola, A. Banerjee and V. Kumar presented a survey on outlier detection techniques. For each technique, they have given normal and anomalous behavior along with its advantages and disadvantages [3].

L. Huang, X. Nguyen, M. Garofalakis, M. Jordan, A. D. Joseph, and N. Taft proposed a method for discovering anomaly by continuously tracking the projection of data onto a residual subspace. They developed anomaly detector in which adaptive local data filters are used with pca. They used to send coordinator just enough data to enable accurate global detection [4].

Wei Wang, Xiaohong Guan, and Xiangliang Zhang proposed an intrusion detection method based on Principal Component Analysis (PCA). This method uses system call data and provides low overhead and high efficiency. PCA is applied to reduce high dimensional data vectors. This method is feasible for real time intrusion detection with high detection accuracy and low computation expenses. This method takes into account frequency property, instead of considering the transition information of the system calls or commands. However, research is in progress to mix the frequencies property with the transition information of system calls so that lower false alarms can be achieved [5]. N.L.D. Khoa and S. Chawla present a method to find outliers using 'commute distance'. Commute distance between two nodes captures both the distance between them and their local neighborhood densities. They show by analysis and experiments that using this measure can capture both global and local outliers effectively with just a distance based method. The method can also detect outlying clusters which other traditional methods often fail to capture and also shows a high resistance to noise than local outlier detection method [8].

X Song, M. Wu,C. Jermaine, Sanjay Ranka presented a way to determine whether reported anomaly is interesting or not. They make use of domain knowledge provided by user and set of environmental attributes. This conditional anomaly detection technique uses the differences between these attributes and proposes three different expectation- maximization algorithms [6].

C. Aggarwal and P. Yu presented a method to find outliers from high dimensional data space. They used the behavior of projections in order to find out outliers [7].

III. IMPLEMENTATION DETAILS

A. Proposed System

The proposed system has breakdown structure shown in fig. 2.

©IJRASET: All Rights are Reserved

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

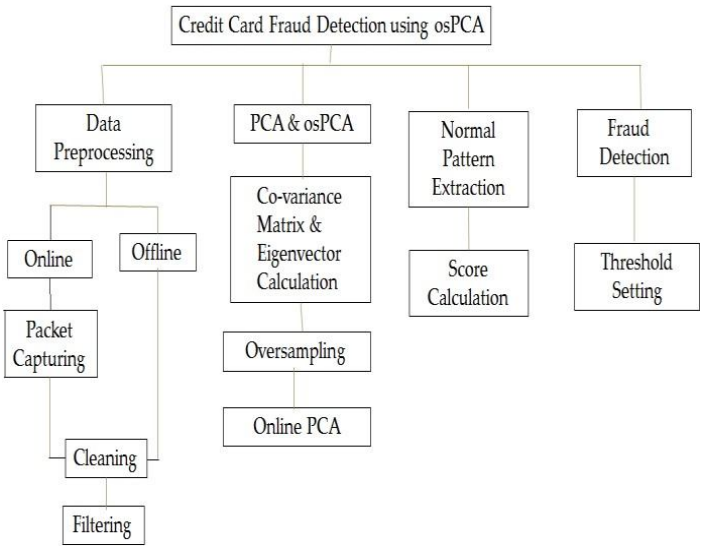


Fig. 2 The Breakdown Structure

B. Data Preprocessing

Data Preprocessing refers to remove noisy unwanted data by cleaning and filtering. The goal of this module is to filter out duplicates and noisy data. Data Preprocessing works in two phases: Online and Offline. In Online preprocessing first packet is captured and then cleaning and filtering is applied. In Offline preprocessing the dataset is loaded in ARFF file format and then filtering is applied. This module also generates summary information for all attributes. For numerical attributes it will calculate minimum, maximum, mean and standard deviation and for nominal attribute it will return count of each label.

C. Principal Component Analysis(PCA) and Oversampling PCA

1) Principal Component Analysis(PCA): Principal Component Analysis (PCA) is an unsupervised dimension reduction method. It is a mathematical process to convert a set of co-related variables into a set of un- correlated variables. These variables are called as principal components. The number of principal component is less than or equal to number of original variables. The first principal component is having highest possible variance [9]. To obtain the principal components there is need to construct the data covariance matrix and calculate its dominant eigenvectors. These eigenvectors are most informative in original data space and hence they are called as principal directions [5].

Let,

$$A = [x_1^T, x_2^T \ldots x_n^T] \in IR^{n \times p} \quad \ldots (1)$$

where each row xi represents a data instance in a p dimensional space, and n is the number of the instances.
PCA can be calculated by following formula,

Max

$$U \in IR^{n \times p} \|U\| = I(U) = \sum_{i=1}^{n} \|(x_i - \mu) - UU^T(x_i - \mu)\|^2 \quad \ldots (2)$$

Where,

$U^T(x_i - \mu)$

Determine the optimal coefficients to weight each principal direction. The PCA problem can be solved by deriving an eigenvalue decomposition problem of the covariance data matrix,

©IJRASET: All Rights are Reserved

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

$$\sum_A U = U^{\wedge} \quad \ldots(3)$$

Where,

$$\sum_A = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T \quad \ldots(4)$$

Eq. (4) is the covariance matrix, μ is the global mean. Each column of U represents an eigenvector of ∑A and the corresponding diagonal entry in ^ is the associated eigenvalue. For dimension reduction, the last few eigenvectors will be discarded due to their negligible contribution to the data distribution.

The clustered blue circles in Fig. 3 represent normal data instances, the red square denotes an outlier, and the green arrow is the dominant principal direction. From Fig. 3, we see that the principal direction is deviated when an outlier instance is added.

The presence of an outlier instance produces a large angle between the resulting and the original principal directions. But this angle will be small when a normal data point is added. Therefore, this technique uses this property to determine the oulierness of the target data point [1].

2) *Oversampling Principal Component Analysis(osPCA):* Oversampling is the process of duplicating the data instances to obtain a balanced dataset. For practical outlier detection problems, the size of the dataset is typically large, and thus it may not cause the variation of principal direction by presence of single outlier. In addition to this PCA framework for outlier detection has to perform n PCA analysis for a dataset with n data instances in a p- dimensional space, which is not computationally feasible for large scale and online problems. spa will duplicate the target instance multiple times by applying concept of oversampling. If the target instance is an outlier, this oversampling scheme allows us to overemphasize its effect on the most dominant eigenvector, and detecting outliers [1].

Suppose that we oversample the target instance times, the associated PCA problem can be formulated as follows,

$$\sum_{\tilde{A}} \tilde{u}_t = \lambda \tilde{u}_t \quad \ldots(5)$$

In this osPCA framework, this technique duplicate the target instance $\tilde{n}$ times e.g., 10 percent of the size of original dataset. PCA can be considered as a problem to minimize the reconstruction error,

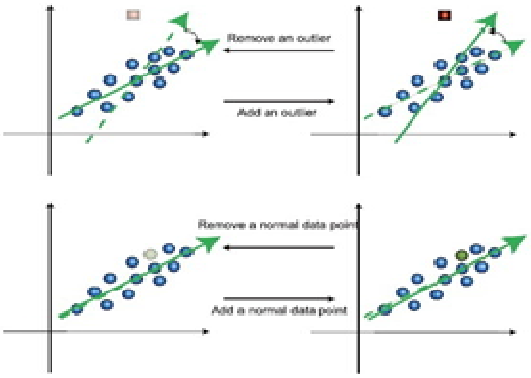


Fig. 3 Effects of adding/removing an outlier or a normal data instance on the principal directions.

Where $\bar{x}_t$ is $(x_i - \mu)$ U is, the matrix consisting of k dominant eigenvectors, and $UU^T\bar{x}_i$ is the reconstructed version of $\bar{x}_i$ using the eigenvectors in U. The above reconstruction error function can be further approximated by a least squares form,

$$\text{Min} \quad U \in IR^{p\times k}, U^T U = I \qquad J_{ls}(U) = \sum_{i=1}^{n}\|\bar{x}_i - UU'^T\bar{x}_i\|^2 \quad \ldots(6)$$

Where U' is the approximation of U, and thus $y_i = U'^T\bar{x}_i \in IR^k$ is the approximation of the projected data $U^T\bar{x}_i$ in the lower k dimensional space. Based on this formulation, the reconstruction error has a quadratic form and is a function of U, which can be computed by solving a least squares problem. The trick for this least square problem is the approximation of $U^T\bar{x}_i$ by $y_i = U'^T\bar{x}_i$ In an online setting, approximate each $U_i^T\bar{x}_i$ by its previous solution $U_{i-1}^T\bar{x}_i$ as follows,

©IJRASET: All Rights are Reserved

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Min

$$U_t \in IR^{p\times k}, U^T U = I \; J_t(Ut) = \sum_{i=1}^{t} \|\bar{x}_i - U_i y_i\|^2 \ldots (8)$$

This projection approximation provides a fast calculation of principle directions in our oversampling PCA.
This projection approximation provides a fast calculation of principle directions in our oversampling PCA.

$$\tilde{u} = \frac{\beta(\sum_{i=1}^{n} y_i x_i) + y_t x_t}{\beta(\sum_{i=1}^{n} y_i^2) + y_t^2} \ldots . (9)$$

Eq. (9) provides an effective and efficient updating technique for osPCA, which allows us to determine the principal direction of the data. This updating process makes anomaly detection in online or streaming data settings feasible. More importantly, since it only need to calculate the solution of the original PCA offline, it is not necessary to keep the entire covariance matrix in the entire updating process. Once the final principal direction is determined, to use the cosine similarity to determine the difference between the current solution and the original one, and thus the score of outlierness for the target instance can be determined accordingly.

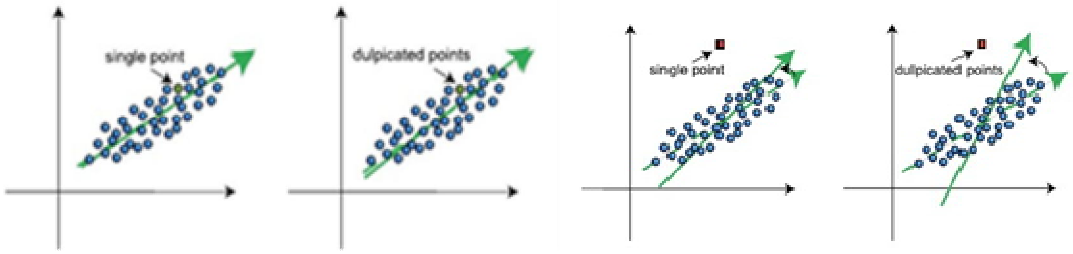


Fig. 4 The effect of an oversampled normal data or outlier instance on the principal direction.

D. Normal Pattern Extraction

Given a data set A with n data instances, we first extract the dominant principal direction u_t from it. If the target instance is at we next compute the leading principal direction $\tilde{u}_t$ without a_t present.˜ without at present. To identify the outliers in a data set, we simply repeat this procedure n time with the LOO strategy (one for each target instance).

$$\sum_{A} \tilde{u}_t = \lambda \tilde{u}_t \ldots . (10)$$

Where,

$$\hat{A} = A \backslash \{x_t\}$$

Once these eigenvectors $\tilde{u}_t$ are obtained, we use absolute value of cosine similarity to measure the variation of the principal directions,

$$s_t = 1 - \left| \frac{\langle \tilde{u}_t, u \rangle}{\|\tilde{u}_t\| \|u\|} \right| \ldots (11)$$

This s_t can be considered as a "score of outlierness," which indicates the anomaly of the target instance x_t[1].

E. Outlier Detection

This module uses the score computed in module 3 to determine the anomaly of each received data point. A higher score of s_t indicate that the target instance is more likely to be an outlier. For a target instance, if it's s_t is above some threshold, we then identify this instance as an outlier. If newly received data instance is above the threshold, it will be identified as a fraud; otherwise, it will be considered as a normal data point, and we will update our osPCA model accordingly. The online detection phase, use the dominant principal direction of the filtered training normal data extracted in the data cleaning phase to detect each arriving target instance.

©IJRASET: All Rights are Reserved

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

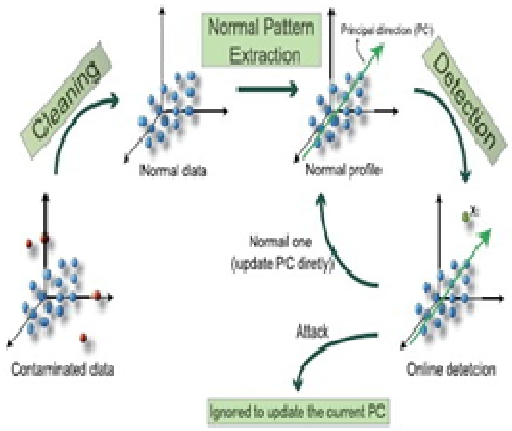


Fig. 5 The framework for Credit Card Fraud Detection.

The entire online detection phase, there is only need to keep this p dimensional eigenvector, and thus the memory requirement is less. Figure 5 shows the framework for credit card fraud detection. This framework works in two phases: Cleaning Phase and Detection Phase.

The pseudo code of online osPCA with the LOO strategy for outlier detection is described in Algorithm 1. It should note that it is required to compute x_{proj} and y once in Algorithm1, reduce the computation time when calculating $\tilde{u}$[1].

Algorithm 1		Outlier Detection using Oversampling PCA for Credit Card Fraud Detection.
Require	:	The data matrix $A = [x_1^T; x_2^T; \dots x_n^T]$ and the weight β.
Ensure	:	Score of outlierness $s = [s_1\ s_2 \dots s_n]$ If si is higher than a threshold, xi is an outlier.

Step1: Compute first principal direction u by using (8);

Step2: Keep $\bar{x}_{proj} = \sum_{j=1}^{n} y_j \bar{x}_j$ and d $y = \sum_{j=1}^{n} y_j^2$ in (9);

Step3: for i =1 to n do

$\tilde{u} \leftarrow \frac{\beta \bar{x}_{proj} + y_i \bar{x}_i}{\beta y + y_i^2}$ by (8)

Step4: $s_i \leftarrow 1 - \left|\frac{\langle \tilde{u}, u \rangle}{\|\tilde{u}\| \|u\|}\right|$ by (11)

End for

IV. RESULTS & DISCUSSION

A. Experimental Setup

To conduct the experiment, we have used Intel Pentium 5, 2.5 GHz, 4GB RAM, Windows 7 platform. All modules are developed in Java Environment 1.8.5 with Netbeans 7.4. To examine outlier detection for credit card fraud we use standard German Credit Card Fraud dataset https://archive.ics.uci.edu/ml/datasets/Statlog/German which is available on UCI Machine Learning Repository. This dataset consists of 20 attributes and 1000 instances. The attributes are both numerical and categorical. We also use two-dimensional synthetic data to test this technique. In synthetic data set we will generate 50 data instances in which 40 are normal instances and

©IJRASET: All Rights are Reserved

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

remaining 10 are outliers.

B. Results & Analysis

To verify the feasibility of this dissertation idea, we conduct experiments on both synthetic and real data sets.

1) *Real World Dataset:* We performed experiments on chosen dataset by taking different number of instances. Initially we select 30 data instances in which 29 are normal records and remaining 1 record is an outlier. We increase gradually the number of instances and number of outliers. Initially the preprocessing is performed on chosen dataset which removes duplicate data instances. The output of pre-processing is filtered dataset which is given to the next module PCA.PCA calculates eigenvalues and eigenvectors of given dataset and sort dimensions in descending order of their eigenvalues. In order to reduce dimensionality few lowest eigenvectors with small eigenvalues are discarded. Here we discard such eigenvectors whose values are almost zero. The output of PCA is reduced dimensions, which contain highest information and contribute for outlier detection.

TABLE I
OUTPUT OF PCA

Sr.No.	Attribute Name	Attribute Type
1	Duration	Numeric
2	Credit-Amount	Numeric
3	Installment-Commitment	Numeric
4	Residence-Since	Numeric
5	Age	Numeric
6	Existing-Credits	Numeric
7	Num-of-Dependents	Numeric

Table 1 shows set of attributes which contributes for outlier detection. These reduced attribute set is given as an input to OSPCA in which over sampling is applied to increase target instances by omitting records whose variance is almost zero. As an effect dataset is further reduced. For example, if we consider case of 100 data instance for testing, here 55 records are oversampled which indicates only 45 records are sufficient to test. Further score is calculated for all 45 record and appropriate threshold is set to detect an outlier or fraud user profile. The Table 8.2 shows the results obtained at the end of experiment.

TABLE III
EFFECT OF NUMBER OF INSTANCES AND DETECTION RATE ON STANDARD DATASET

Number of Instances	Actual Number of Outliers	Outliers Detected	Detection Rate (In Per)
30	1	1	100
50	2	1	50
60	4	4	100
70	5	5	100
80	8	8	100
90	9	8	88.8
100	9	9	100
200	10	10	100
400	13	13	100
500	15	15	100
1000	30	30	100

$$System\ Performance = \frac{\sum DR}{Number\ of\ Observations} \quad (4.1)$$

=

©IJRASET: All Rights are Reserved

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

94.44

Average Threshold = 0.00871

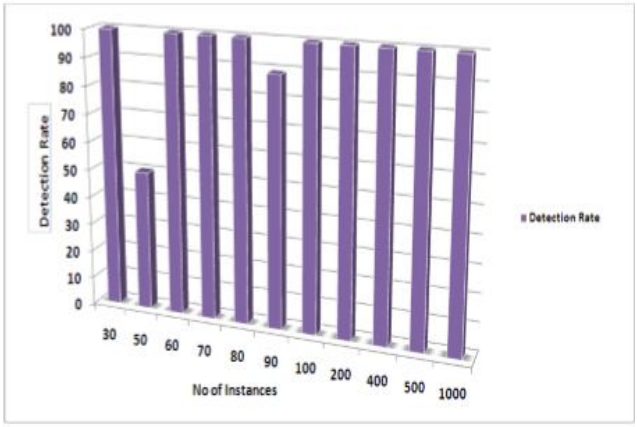


Fig. 6 Bar Chart Effect of Number of Instances and Detection Rate on Standard Dataset

From Table 2 it is clear that system is giving 100 percent performance except for data instances 50 and 90. For 50 data instances it gives 50 percent and for 90 data instances it gives 88.88 percent. Figure 6 shows the graph of Number of Instances with Detection Rate. Further we calculate the time required for classification without using oversampling and with using oversampling.

TABLE IIIII

TIME COMPARISON OF CLASSIFICATION WITH & WITHOUT OVERSAMPLING WITH STANDARD DATASET

Number of Instances	CPU Time for Classification without	CPU Time for Classification with
30	0.784837	0.023788
50	0.912178	0.022128
60	0.937357	0.018809
70	1.06446	0.018532
80	1.204813	0.0177
90	1.367021	0.0177
100	1.452	0.019639
200	2.901194	0.019086
400	16.315546	0.022405
500	19.21942	0.021852
1000	78.498815	0.014659

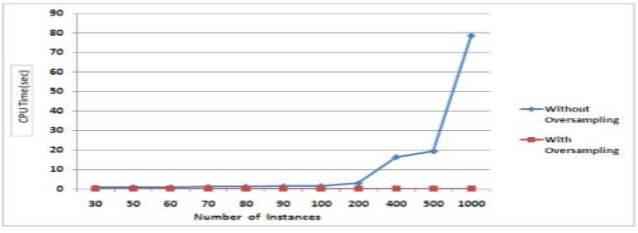


Fig. 7 Comparison of Classification Time without Oversampling and with Oversampling

Table 3 shows CPU time required for classification of dataset without using oversampling and with using oversampling for all test data instances. After observation of this table we can say that less time is required for classification with oversampling as compared

©IJRASET: All Rights are Reserved

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

with without oversampling.

Figure 7 shows graph of time for classification without oversampling and with oversampling. Here blue color line shows time without oversampling and red color line shows time with oversampling. From graph, it is clear that time required for classification without oversampling is more and increasing with increase in number of data instances, but with oversampling graph is linear.

2) *Synthetic Dataset:* We generate a 50 synthetic data records, which consists of 40 normal instances and 10 deviated instances. We apply the algorithm gradually starting with 10 synthetic data instances which contain 9 normal instances and 1outlier. After that number of normal and outlier instances are increased gradually. Further we calculate score for each data instance. By setting the proper threshold we detect 10 deviated data points as outliers.

TABLE IVV

EFFECT OF NUMBER OF INSTANCES ON DETECTION RATE ON SYNTHETIC DATA

Number of Instances	Actual Number of Outliers	Outliers Detected	Detection Rate (In Per)
10	1	1	100
20	3	2	66.66
30	5	5	100
40	8	8	100
50	10	10	100

Table 4 shows result of experiments on synthetic data. From Table 8.4 it is clear that system will give 100 percent detection rate except for one case.

TABLE V

TIME COMPARISON OF CLASSIFICATION WITH & WITHOUT OVERSAMPLING ON SYNTHETIC DATA

Number of Instances	CPU Time for Classification without Oversampling	CPU Time for Classification with Oversampling
10	0.158493	0.017149
20	0.134490	0.01797
30	0.181715	0.01770
40	0.212750	0.018532
50	1.287522	0.016873

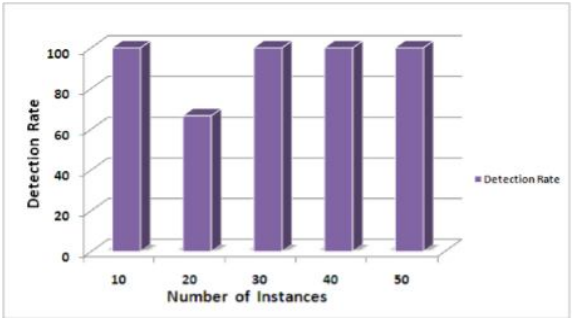


Fig. 8 Bar Chart Effect of Number of Instances and Detection Rate on Synthetic Dataset

Further we calculate the time required for classification without using oversampling and with using oversampling. The results obtained on different number of records are listed in Table 5.

©IJRASET: All Rights are Reserved

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

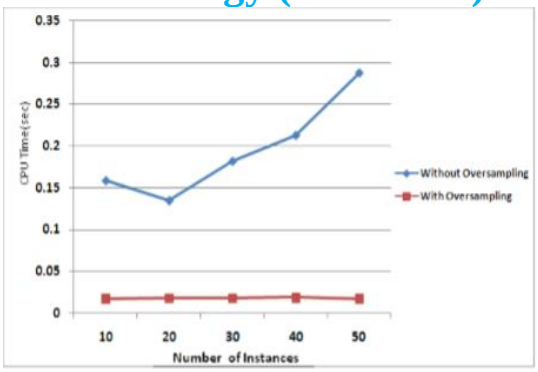


Fig. 9 Comparison of Classification Time without Oversampling and with Oversampling for Synthetic Data

System Performance for Synthetic Data=93.33
Average Threshold = 0.0169
Figure 9 shows graph of time comparison with oversampling and without oversampling on synthetic data. Blue color line shows time for classification without oversampling which increases with increase in number of instances. Red line shows classification time with oversampling which is linear in nature.

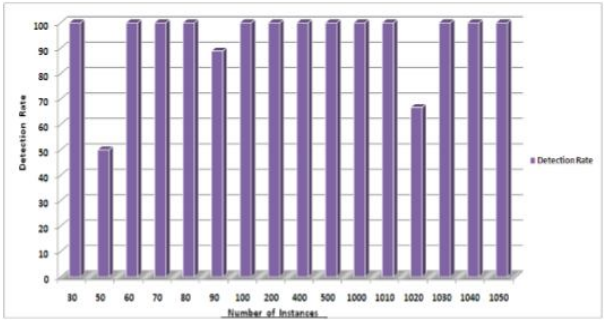


Fig. 10 Total System Performance

If we combine the result of both standard dataset and synthetic dataset, we get the overall system performance in terms of detection rate. Figure 8.5 shows combined result of the system, which shows system gives satisfactory results. We can calculate average system performance by considering performance on both dataset.

Average System Threshold = 0.0128
Total System Performance = 93.885

V. CONCLUSIONS

This paper provides credit card fraud detection method based on oversample PCA. PCA performs dimension reduction by discarding attributes with lower eigenvalues. When oversampling a data instance, this technique enables the osPCA to efficiently update the principal direction without solving eigenvalue decomposition problems. This online osPCA technique will preferable for online large scale or streaming data problems. The attempt is to improve the performance in terms of memory and time requirements.

REFERENCES

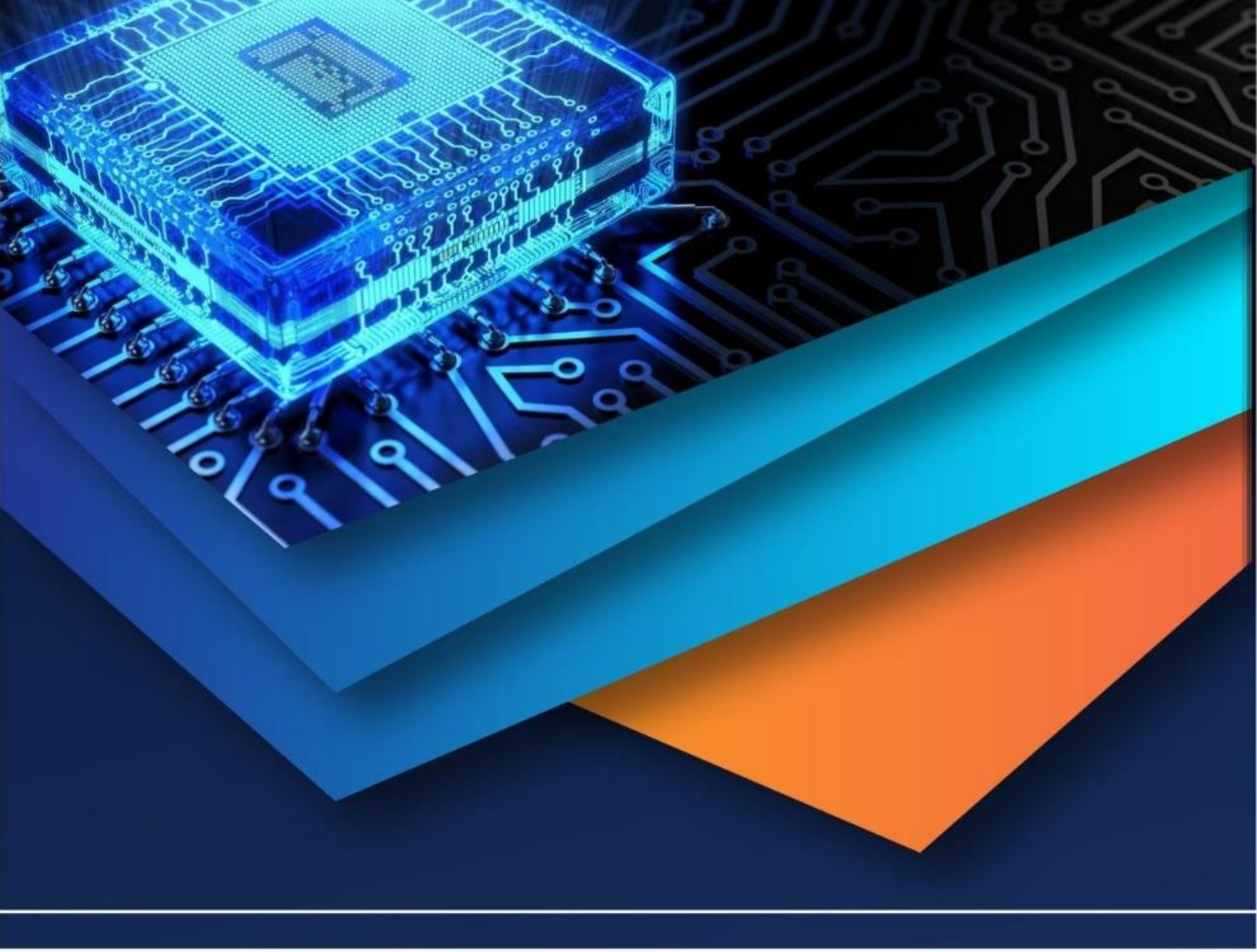
[1] Y. Lee, Y. Yeh, and Y. Wang," Anomaly Detection via Online Oversampling Principal Component Analysis", IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 7, July 2013.
[2] M. Breunig, H.-P. Kriegel, R.T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," Proc. ACM SIGMOD Int'l Conf. Management of Data, 200
[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," ACM Computing Surveys, vol. 41, no. 3, pp. 15:1-15:58, 2009
[4] L. Huang, X. Nguyen, M. Garofalakis, M. Jordan, A.D. Joseph, and N. Taft, "In-Network Pca and Anomaly Detection," Proc. Advances in Neural Information Processing Systems 19, 2007
[5] W. Wang, X. Guan, and X. Zhang, "A Novel Intrusion Detection Method Based on Principal Component Analysis in Computer Security," Proc. Int'l Symp. Neural Networks, 2004

©IJRASET: All Rights are Reserved

Volume 5 Issue V, May 2017
ISSN: 2321-9653

[6] Xiuyao Song, Mingxi Wu, Christopher Jermaine, Sanjay Ranka,"Conditional Anomaly Detection", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERIN
[7] C.C. Aggarwal and P.S. Yu, "Outlier Detection for High Dimensional Data," Proc. ACM SIGMOD Int'l Conf. Management of Data, 200
[8] N.L.D. Khoa and S. Chawla, "Robust Outlier Detection Using Commute Time and Eigenspace Embedding," Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining, 2010
[9] F. Angiulli, S. Basta, and C. Pizzuti, "Distance-Based Detection and Prediction of Outliers," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 2, pp. 145-160,2006.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)