www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Hybrid Supervised and Unsupervised Learning for Detecting Personal Loan Fraud

[1] B.Kethana, [2] Ch.Keerthana,

[1]Assistant Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.
[2] MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

Abstract—

As a result of the explosion of e-services like online shopping, online banking, and mobile payment systems, the demand for personal loans for consumption has skyrocketed in the last few years. Massive losses due to credit loan fraud are an inevitable consequence of ineffective grid verification and monitoring [1]. Due to the high volume of credit card transactions and the inherent difficulty of manually inspecting and verifying each one, machine learning approaches are increasingly being used to automatically identify fraudulent transactions. The XGBoost model has been used for data mining and analysis in this paper, drawing inspiration from its stellar reputation in numerous data mining challenges. It's becoming more difficult to employ data mining tools without violating people's privacy, which is a major worry. Also, certain features are thought to have little information or some redundancy, while others store the crucial data, which makes feature engineering more difficult, according to recent research on loan fraud detection. This study presents KPXGBoost, a novel hybrid unsupervised and supervised learning model that integrates Kernel Principal Component Analysis (Kernel PCA) with the XGBoost method. The goal is to filter out irrelevant information while preserving important data without understanding the meaning of the data. In order to prevent overloading, we evaluate the performance of XGBoost and P-XGBoost with other traditional machine learning techniques using grid search. As it turns out, P-XGBoost is more effective than XGBoost in detecting fraudulent behavior, which offers a fresh angle on the problem while still keeping customers' personal information safe.

Index Terms—

principal component analysis, supervised learning, extreme gradient boosting, and unsupervised learning

I. INTRODUCTION

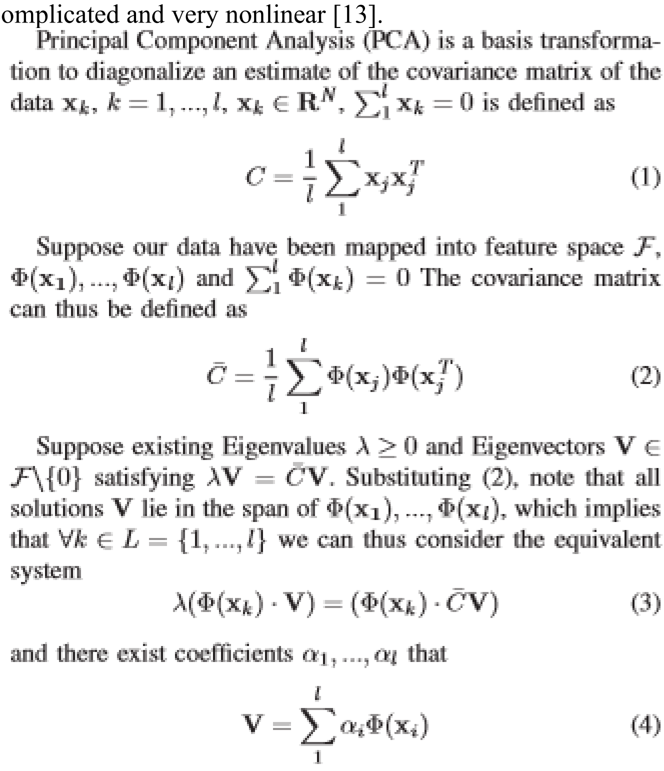
Due to the unprecedented growth of the personal consumption loan market, both card issuers' and researchers' focus on detecting and preventing fraud has grown in recent years. This is because even a small amount of fraudulent activity could result in millions of dollars saved. We are focusing on using machine learning techniques to automatically identify fraudulent transactions due to the obvious challenge of manually verifying such a high volume of credit card transactions. Our major focus is on developing highly accurate methods for predicting fraudulent transactions by analyzing client behavior data. To build a credit card fraud detection algorithm while keeping customers' privacy in mind, we provide a new approach that combines supervised and unsupervised learning. We provide an unsupervised learning method that uses kernel principal component analysis to break down the dataset's dimensions. The fraud detection algorithm then uses XGboost to make a forecast.

II. THEORETICAL BASIS AND RELATED RESEARCH

A.The Faux Pay Scandal Two distinct forms of credit card fraud exist. There are two types of fraud: application fraud [2] and behavior fraud [3]. When someone submits a false credit card application, this is called application fraud. It happens when the card issuer gives the go-ahead to a fraudulent applicant who uses false identification to get a new credit card. The most common types of behavioral fraud are card-not-present, counterfeit-card, and theft/stolen-card fraud. The term "fraud detection" is used to describe the process of identifying fraudulent activities throughout the next sections of this study. B. Research in This Area Statistical modeling approaches and feature engineering methods have been the primary foci of prior fraud detection research. [4] As far as statistical methodologies are concerned, a fraud detection system like this might be built using either supervised or unsupervised learning techniques. Researchers have offered a variety of supervised learning approaches, including decision trees, random forests, hidden Markov models, artificial neural networks (ANN) [5], Bayesian belief networks [6], decision trees [7], and random forests [8]. There is a potential issue with these supervised learning approaches in that they rely on a single dataset that has been correctly tagged, which isn't always the case in real-life transaction records. Also, pre-processing before model implementation is more difficult due to the imbalanced sample distribution; fewer than 0.1% of transactions are fraudulent [4]. In order to identify out-of-the-ordinary behaviors, unsupervised learners cluster all samples. So, it is possible to infer fraud if a single transaction record significantly differs from the typical clusters. Two examples of unsupervised learning techniques are self-organizing maps [10] and peer group analysis [3]. Due to the fact that suspicious activity is often discovered to be associated with valid transactions, unsupervised approaches result in greater false alarm rates [11]. The expense of identifying actual fraudulent behavior is roughly equivalent to the false alarm rate, and a lower rate indicates less misjudgement of legitimate transactions. Therefore, supervised learning approaches have been the primary focus of most prior research. Feature engineering that is domain-based is often used in fraud detection models. [4] The success of machine learning techniques is directly correlated to high-quality features. When training machine learning models for this fraud detection field, it requires a huge number of computational resources to maintain all the attributes of previous transaction data. However, full-mark performance does not imply full-feature. This work does trials on one unknown dataset because of the privacy term, however in earlier research, the significance of every characteristic is evident. Using Kernel PCA is a dependable way to extract the feature variables from the input dataset, even if selecting features for the transaction records is still an issue, particularly when the features' meanings are uncertain.

III. THE PROPOSED KP-XGBOOST

Analysis using Kernel Principal Components A non-linear feature extractor called Kernel Principal Component Analysis (Kernel PCA) was proposed by MIKA, S. et al. as an effective preprocessing step for classification methods, including data reduction, reconstruction, and de-noising. [12] As an extension of linear principal component analysis, it may help with the issue that classical PCA can't handle—namely, that real-world data manifolds are often complicated and very nonlinear [13].



Principal Component Analysis (PCA) is a basis transformation to diagonalize an estimate of the covariance matrix of the data $\mathbf{x}_k$, $k = 1, ..., l$, $\mathbf{x}_k \in \mathbf{R}^N$, $\sum_1^l \mathbf{x}_k = 0$ is defined as

$$C = \frac{1}{l}\sum_1^l \mathbf{x}_j\mathbf{x}_j^T \tag{1}$$

Suppose our data have been mapped into feature space $\mathcal{F}$, $\Phi(\mathbf{x_1}), ..., \Phi(\mathbf{x}_l)$ and $\sum_1^l \Phi(\mathbf{x}_k) = 0$ The covariance matrix can thus be defined as

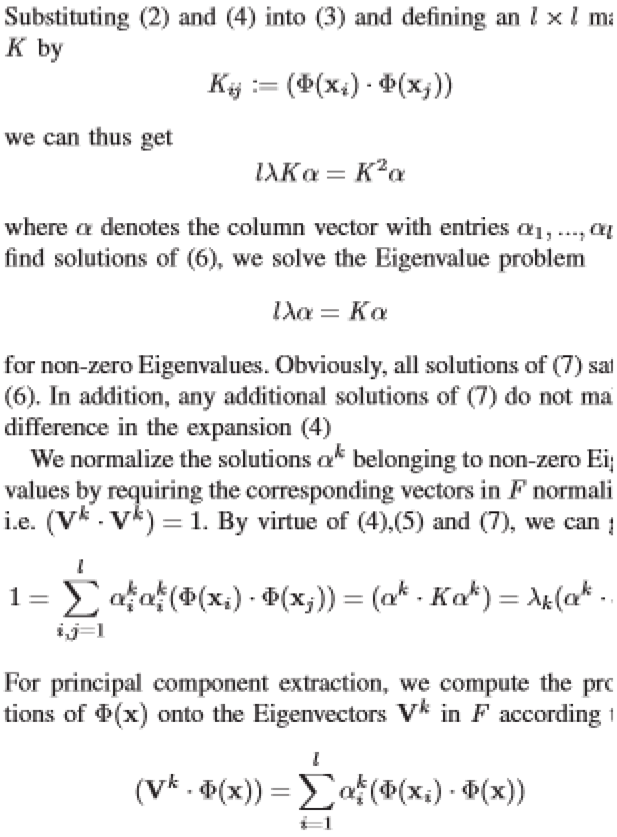
$$\bar{C} = \frac{1}{l}\sum_1^l \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j^T) \tag{2}$$

Suppose existing Eigenvalues $\lambda \geq 0$ and Eigenvectors $\mathbf{V} \in \mathcal{F}\backslash\{0\}$ satisfying $\lambda\mathbf{V} = \bar{C}\mathbf{V}$. Substituting (2), note that all solutions $\mathbf{V}$ lie in the span of $\Phi(\mathbf{x_1}), ..., \Phi(\mathbf{x}_l)$, which implies that $\forall k \in L = \{1, ..., l\}$ we can thus consider the equivalent system

$$\lambda(\Phi(\mathbf{x}_k) \cdot \mathbf{V}) = (\Phi(\mathbf{x}_k) \cdot \bar{C}\mathbf{V}) \tag{3}$$

and there exist coefficients $\alpha_1, ..., \alpha_l$ that

$$\mathbf{V} = \sum_1^l \alpha_i \Phi(\mathbf{x}_i) \tag{4}$$



Substituting (2) and (4) into (3) and defining an $l \times l$ ma
K by

$$K_{ij} := (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j))$$

we can thus get

$$l\lambda K\alpha = K^2\alpha$$

where α denotes the column vector with entries $\alpha_1, ..., \alpha_l$
find solutions of (6), we solve the Eigenvalue problem

$$l\lambda\alpha = K\alpha$$

for non-zero Eigenvalues. Obviously, all solutions of (7) sat
(6). In addition, any additional solutions of (7) do not ma
difference in the expansion (4)

We normalize the solutions α^k belonging to non-zero Ei
values by requiring the corresponding vectors in F normali
i.e. $(\mathbf{V}^k \cdot \mathbf{V}^k) = 1$. By virtue of (4),(5) and (7), we can

$$1 = \sum_{i,j=1}^{l} \alpha_i^k \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) = (\alpha^k \cdot K\alpha^k) = \lambda_k(\alpha^k \cdot$$

For principal component extraction, we compute the pro
tions of $\Phi(\mathbf{x})$ onto the Eigenvectors $\mathbf{V}^k$ in F according t

$$(\mathbf{V}^k \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{l} \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}))$$

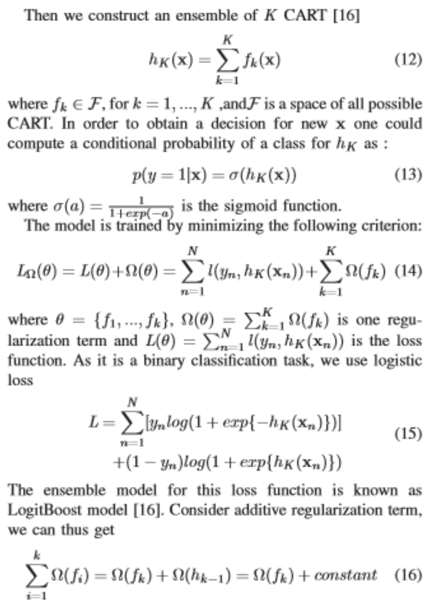
It should be noted that paragraphs (5) and (9) do not explicitly need the Φ(xi). Hence, computational usage of the kernel function is possible even in the absence of the map Φ. So, the kernel function is available for usage in SVMs. [14]

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^d \qquad (10)$$

Kernel PCA Extreme Gradient Boosting (XGBoost) may be obtained by replacing kernel functions for every instance of (Ŧ(x)•Ŧ(y)). Zie¸ba et al. [15] suggested learning a set of decision trees for bankruptcy prediction using extreme gradient boosting (XGBoost). Mathematical operations like adding, subtracting, multiplying, and dividing make up their so-called synthetic characteristic. It is possible to build each synthetic feature using an evolutionary approach and consider it as a regression model. Let x be a sample in the dataset where X⊂R D and y be a label in the set {0,1}. We describe the underlying classifier model, which is Classification and Regression Trees (CART), by the weights associated with the structure's leaves.

$$f_k(\mathbf{x_n}) = w_{q(\mathbf{x})} \qquad (11)$$

the function that accepts an example and returns the path id in the tree structure, where T is the number of routes or leaves, and q(x) is defined as q: RD → {1,...,T}. A leaf bearing weight wi marks the end of a trail.



Then we construct an ensemble of K CART [16]

$$h_K(\mathbf{x}) = \sum_{k=1}^{K} f_k(\mathbf{x}) \qquad (12)$$

where $f_k \in \mathcal{F}$, for $k = 1, ..., K$,and$\mathcal{F}$ is a space of all possible CART. In order to obtain a decision for new $\mathbf{x}$ one could compute a conditional probability of a class for h_K as :

$$p(y = 1|\mathbf{x}) = \sigma(h_K(\mathbf{x})) \qquad (13)$$

where $\sigma(a) = \frac{1}{1+exp(-a)}$ is the sigmoid function.

The model is trained by minimizing the following criterion:

$$L_\Omega(\theta) = L(\theta) + \Omega(\theta) = \sum_{n=1}^{N} l(y_n, h_K(\mathbf{x}_n)) + \sum_{k=1}^{K} \Omega(f_k) \qquad (14)$$

where $\theta = \{f_1, ..., f_k\}$, $\Omega(\theta) = \sum_{k=1}^{K} \Omega(f_k)$ is one regularization term and $L(\theta) = \sum_{n=1}^{N} l(y_n, h_K(\mathbf{x}_n))$ is the loss function. As it is a binary classification task, we use logistic loss

$$L = \sum_{n=1}^{N} [y_n log(1 + exp\{-h_K(\mathbf{x}_n)\})] \\ +(1 - y_n) log(1 + exp\{h_K(\mathbf{x}_n)\}) \qquad (15)$$

The ensemble model for this loss function is known as LogitBoost model [16]. Consider additive regularization term, we can thus get

$$\sum_{i=1}^{k} \Omega(f_i) = \Omega(f_k) + \Omega(h_{k-1}) = \Omega(f_k) + constant \qquad (16)$$

Therefore, we can represent (14) as

$$L_\Omega(\theta) = \sum_{n=1}^{N} l(y_n, h_{k-1}(\mathbf{x}_n + f_k(\mathbf{x}_n))) \\ + \Omega(f_k) + constant \qquad (17)$$

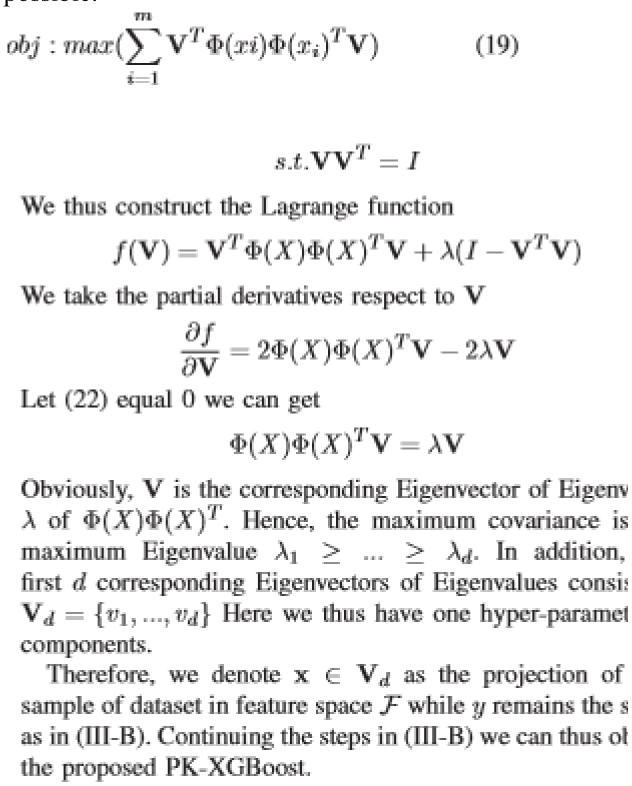
Optional regularization methods for the model include determining the minimum number of examples to be used with each leaf, the maximum depth of the tree, the percentage of features to be randomly selected for each tree construction iteration, or the addition of a new tree with corrected committee tree influence [15].

$$h_k(\mathbf{x_n}) = h_{k-1}(\mathbf{x_n}) + \epsilon f_k(\mathbf{x_n}) \qquad (18)$$

where $\epsilon \in [0, 1]$ is called step-size or shrinkage.

The Advanced KP-XGBoost System Assuming X = {x1,...xm} is the sample set, we may deduce from equation (9) that the project of sample point xi in the new feature space F is V•Ŧ(xi). To achieve maximum dispersion in the projects of all the sample points, we should maximize the covariance in equation (2).

Therefore, the following way of representing it is possible:



$$obj : max(\sum_{i=1}^{m} \mathbf{V}^T \Phi(xi) \Phi(x_i)^T \mathbf{V}) \qquad (19)$$

$$s.t. \mathbf{V}\mathbf{V}^T = I$$

We thus construct the Lagrange function

$$f(\mathbf{V}) = \mathbf{V}^T \Phi(X) \Phi(X)^T \mathbf{V} + \lambda(I - \mathbf{V}^T \mathbf{V})$$

We take the partial derivatives respect to **V**

$$\frac{\partial f}{\partial \mathbf{V}} = 2\Phi(X)\Phi(X)^T \mathbf{V} - 2\lambda \mathbf{V}$$

Let (22) equal 0 we can get

$$\Phi(X)\Phi(X)^T \mathbf{V} = \lambda \mathbf{V}$$

Obviously, **V** is the corresponding Eigenvector of Eigenv λ of $\Phi(X)\Phi(X)^T$. Hence, the maximum covariance is maximum Eigenvalue $\lambda_1 \geq ... \geq \lambda_d$. In addition, first d corresponding Eigenvectors of Eigenvalues consi: $\mathbf{V}_d = \{v_1, ..., v_d\}$ Here we thus have one hyper-paramet components.

Therefore, we denote $\mathbf{x} \in \mathbf{V}_d$ as the projection of sample of dataset in feature space $\mathcal{F}$ while y remains the s as in (III-B). Continuing the steps in (III-B) we can thus ol the proposed PK-XGBoost.

IV. EMPIRICAL ANALYSIS

S. Experimental apparatus To test how well the suggested technique, which uses Kernel PCA and XGBoost, performs, we do the following experiment. In this study, we evaluate KP-XGBoost in comparison to four well-known ML approaches for fraud detection: XGBoost, logistic regression, support vector machine, and random forest (RF). Also, we test PKXGBoost's functionality in an experimental setting initially. In particular, XGBoost takes use of the characteristics that Kernel PCA decomposes. In Fig. 1 we can see the steps that our experiment took. Instead of using decomposition, the other four models are trained directly on the dataset and their results serve as a benchmark. Although 10-fold cross-validation is often used to evaluate and choose models in order to prevent overfitting classifiers [17], we opted to utilize 5-fold in this research since our dataset is much smaller at 50,000 records. Thus, if we partition the raw data into ten groups, each of those subsets will only have five thousand records—far too few to be useful for training. B. Information Outline The fraudulent transactions are labeled as 1 in the real-life dataset used in this work, which comes from one of the biggest banks in China. There is an imbalance in the dataset's distribution, with roughly 10,000 records (nearly 20% of the total) representing fraudulent transactions (see Fig. 2) out of 50,000 records. Due to the high volume of noise and missing values in our data, we perform a series of operations on it before achieving the processed data—essential for statistical models and subsequent analysis—by removing irrelevant and highly correlated features, filling in missing values using column median, transforming multi-class features with one-hot encoding, and finally, applying the min-max scaling method.

Fig. 1. The PK-XGBoost algorithm

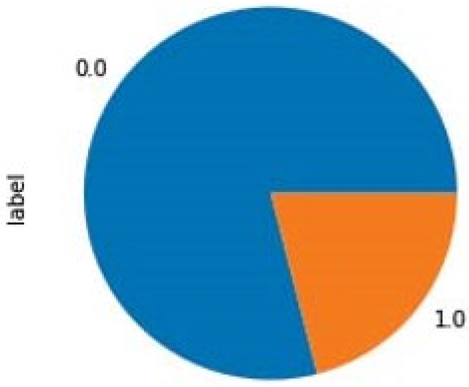


Fig. 2. The label distribution of dataset

(A) Criteria for evaluation Therefore, we use the area under the ROC curve (AUC) value as an overall performance metric for the intravenous mentionedin (IV-B) imbalanced classification test. Being unaffected by a threshold value, the AUC is thought of as a superior overall performance metric than accuracy. According to [4], AUC is a float number between 0 and 1, with a closer value indicating greater performance for a given model. The outcomes of the experiments There are two issues that the empirical experiment hopes to address. The first concerns the precise amount of components; more specifically, the amount of components that should be retained after Kernel PCA.

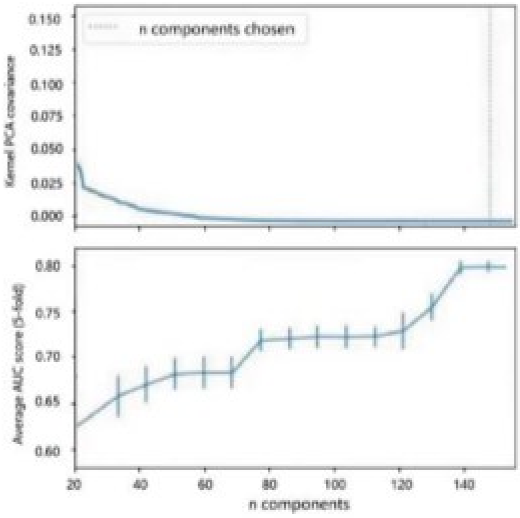


Fig. 3. Grid search on the number of components

to get the most out of it. Components having an accumulative contribution rate greater than 85% are often retained [18]. Nevertheless, it is solely based on first-hand accounts. Therefore, we use grid search to find out how many components XGBoost needs to perform optimally. Figure 3 shows that the PK-XGBoost achieves the best area under the curve (about 0.785) when 150 components are retained. We also need to know whether our suggested algorithm can improve the outcomes of fraud detection. The results are shown in Table I. The experimental findings of our suggested PK-XGBoost are much superior than those of LR,RF SVM. Additionally, the results demonstrate that Kernel PCA improves XGBoost's performance, with an average AUC score that is 0.1 higher than XGBoost's.

TABLE I PERFORMANCE OF CLASSIFIERS

Classifiers	Average AUC score (5-fold)
PK-XGBoost	0.785
Logistic Regression	0.577
Random Forest	0.556
Support Vector Machine	0.577
XGBoost	0.775

V. CONCLUSION

To sum up, this research suggests a novel approach to credit card fraud detection that combines supervised and unsupervised learning. To aid XGBoost in its fraud detection efforts, a data decomposition technique based on Kernel PCA is used to project and breakdown the feature variables. We prioritize customer privacy compared to our past efforts. Feature engineering is very complicated since we can not understand the meaning of each feature while processing the data. To evaluate how well our suggested algorithm works in comparison to more conventional machine learning techniques, we do experimental comparisons. An actual dataset is used to do the empirical analysis. According to the findings, our suggested strategy not only improves upon the original XGBoost's performance but also significantly surpasses all other conventional machine learning approaches. This offers a practical solution that addresses the dual concerns of preventing fraud and safeguarding personal information.

REFERENCES

[1] N. Carneiro, G. Figueira, and M. Costa, “A data mining based system for credit-card fraud detection in e-tail,” Decision Support Systems, vol. 95, 2017.

[2] C. Phuaacbd, “On the communal analysis suspicion scoring for identity crime in streaming credit applications,” European Journal of Operational Research, vol. 195, no. 2, pp. 595–612, 2009.

[3] R. J. Bolton and D. J. Hand, “Statistical fraud detection: A review,” Statistical Science, vol. 17, no. 3, pp. 235–249, 2002.

[4] X. Zhang, Y. Han, W. Xu, and Q. Wang, “Hoba: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture,” Information Sciences, 05 2019.

[5] E. Aleskerov, B. Freisleben, and B. Rao, “Cardwatch: a neural network based database mining system for credit card fraud detection,” in Computational Intelligence for Financial Engineering, 1997.

[6] S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar, “Credit card fraud detection: A fusion approach using dempster–shafer theory and bayesian learning,” Information Fusion, vol. 10, no. 4, pp. 354–363, 2009.

[7] A. Shen, R. Tong, and Y. Deng, “Application of classification models on credit card fraud detection,” in 2007 International Conference on Service Systems and Service Management, June 2007, pp. 1–4.

[8] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, “Data mining for credit card fraud: A comparative study,” Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.

[9] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, “Credit card fraud detection using hidden markov model,” IEEE Transactions on Dependable and Secure Computing, vol. 5, no. 1, pp. 37–48, Jan 2008.

[10] J. T. S. Quah and M. Sriganesh, Real-time credit card fraud detection using computational intelligence, 2008.

[11] M. Krivko, “A hybrid model for plastic card fraud detection systems,” Expert Systems with Applications, vol. 37, no. 8, pp. 6070–6076, 2010.

[12] S. Mika, B. Sch¨olkopf, A. Smola, K. R. M¨uller, and G. R¨atsch, “Kernel pca and de-noising in feature spaces,” in Conference on Advances in Neural Information Processing Systems II, 1999.

[13] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, “Highdimensional and large-scale anomaly detection using a linear one-class svm with deep learning,” Pattern Recognition, vol. 58, no. C, pp. 121– 134, 2016.

[14] B. Scholkopf, C. Burges, and V. Vapnik, “Extracting support data for a given task,” in International Conference on Knowledge Discovery & Data Mining, 1995.

[15] M. Zieba, S. K. Tomczak, and J. M. Tomczak, “Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction,” Expert Systems with Applications, vol. 58, no. C, pp. 93– 101, 2016.

[16] T. Chen and H. Tong, “Higgs boson discovery with boosted trees,” in International Conference on High-energy Physics & Machine Learning, 2014.

[17] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009.

[18] H. J. R and L. Z. Z, “Model validation method with multivariate output based on kernel principal component analysis,” Journal of Beijing University of Aeronautics and Astronautics, vol. 43, no. 7, pp. 1470– 1480, 2017.