2020 5th IEEE International Conference on Big Data Analytics

Personal Loan Fraud Detection Based on Hybrid Supervised and Unsupervised Learning

Hanlin Wen
Department of Artificial Intelligence and Automation
Huazhong University of Science and Technology
Wuhan, China
e-mail: 571170246@qq.com

Fangming Huang
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
Shenzhen China
e-mail: 13713663078@163.com

***Abstract*—In recent years, we have been witnessing a dramatic increase on the personal loan for consumption, due to the rapid development of e-services, including e-commerce, e-finance and mobile payments. Resulting from the lack of effective grid verification and supervision, it inevitably leads to large-scale losses caused by credit loan fraud [1]. Considering the difficulty of manual inspection and verification on the large amount of credit card transactions, machine learning methods are commonly used to detect fraudulent transactions automatically. This article has applied the Extreme Gradient Boosting(XGBoost) model for data mining and analysis, which is inspired by its brilliant reputation in various data mining contests. With people's growing concern about privacy protection, how can we apply data mining techniques while taking consideration into privacy terms is one problem. Additionally, according to current loan fraud detection studies, some features are considered to contain little information or a bit of redundancy, whereas others hold the critical information which makes things harder when feature engineering. In order to filter useless information and preserve the useful information without knowing the meaning of our data, this paper combines Kernel Principal Component Analysis (Kernel PCA) together with XGBoost algorithm and proposes a new hybrid unsupervised and supervised learning model, KP-XGBoost. We use grid search to avoid over-fitting and compare the performance of both XGBoost and P-XGBoost and other classical machine learning methods. It turns out that P-XGBoost outperforms XGBoost in fraud detection, which provides a new perspective to detecting the fraud behaviour while protecting clients' privacy.**

***Index Terms*—supervised learning, unsupervised learning, Extreme Gradient Boosting, principal component analysis**

I. Introduction

With the unparalleled enhancement of the personal consumption loan market, fraud detection and fraud prevention have become important concerns for card issuers and important research topics for researchers since detecting and preventing even a small portion of fraudulent activity would save millions of dollars. However, because of the distinct difficulty of manual verification on the large amount of credit card transactions, applying machine learning methods to detect fraudulent transactions automatically is the focus in our research. We are mainly concerned with how to predict the potential fraudulent transactions as accurate as possible through analysing clients' behaviour data.

In this paper, we propose a novel hybrid supervised and unsupervised learning method for developing a credit card fraud detection algorithm under the condition of protecting clients' privacy. An unsupervised learning based on kernel principal component analysis is proposed to decompose the dimension of the dataset. Then XGboost are incorporated into the fraud detection algorithm to perform a prediction result.

II. Theoretical Basis and Related research

A. Credit Card Fraud

Credit card fraud is divided into two types. One is application fraud [2], and the other is behaviour fraud [3].Application fraud refers to situations in which an application for a credit card is fraudulent. It occurs when a fraudster applies for a new credit card using counterfeit identity information and the card issuer approves this application. Behaviour fraud mainly includes the theft/stolen-card fraud, counterfeit-card fraud and card-not-present fraud.In the rest of this research, fraud detection refers to detecting fraudulent behaviour unless otherwise noted.

B. Related Research

Previous researches mainly focus on two aspects of fraud detection: statistical modelling methods and feature engineering methods. [4]

For the talk of statistical methods, both supervised and unsupervised learning methods are proposed to build such a fraud detection system. For supervised learning methods, Researchers have proposed artificial neural networks (ANN) [5],Bayesian belief networks [6], decision trees [7], random forest [8], and hidden Markov models [9].One problem these supervised learning methods may face is that they require one accurately-labelled dataset, whereas some of the real-life transaction records are unavoidable mislabelled. In addition, the distribution of the samples are unbalanced, where only less than than 0.1 percent of the transactions are fraudulent [4] which increases the difficulty when pre-processing before model implementation. As for unsupervised learning, they group every sample into different clusters to detect abnormal behaviours. If one transaction record deviates strongly from the normal clusters, it can thus be suspected as a fraud. Unsupervised learning methods they are peer group analysis [3], and self-organizing maps [10]. The unsupervised methods

978-1-7281-4111-4/20/$31.00 ©2020 IEEE

produce higher false alarm rates because unusual behaviour is often found to correspond to a legitimate transaction [11]. To some extent, a false alarm rate usually means the cost of detecting a real fraudulent behaviour, and a lower false alarm rate represents less misjudgement of normal transactions. Hence, most of the previous studies mainly concentrate on supervised learning methods.

Fraud detection modelling usually involves a domain-based feature engineering process. [4]Obviously, good feature give rise to the performance of machine learning methods. In this fraud detection field, keeping all the features of historical transaction records costs large amount of computing resources when training the machine learning models. Nevertheless, full features do not mean full-mark performance. However, in previous researches the meaning of every feature is clear, while this paper conducts experiments on one unknown dataset due to the privacy term. Although the choice of features in the transaction records are still one problem, especially when the meanings of features are unknown, the idea of using Kernel PCA are one reliable method to derive the feature variables in the input dataset.

III. The proposed KP-XGBoost

A. Kernel Principal Component Analysis

MIKA,S. et al. propose Kernel Principal Component Analysis (Kernel PCA) as a non-linear feature extractor which has proven powerful as a preprocessing step (e.g. data compression, reconstruction, and de-noising) for classification algorithms. [12]It can serve as a natural generalization of linear principal component analysis and solve the problem that real-world data manifolds are usually complex and highly nonlinear [13], which the classical PCA is incapable of tackling

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 $l \times l$ matrix K by

$$K_{ij} := (\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)) \tag{5}$$

we can thus get

$$l\lambda K\alpha = K^2\alpha \tag{6}$$

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

$$l\lambda\alpha = K\alpha \tag{7}$$

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

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

$$1 = \sum_{i,j=1}^{l} \alpha_i^k \alpha_i^k (\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)) = (\alpha^k \cdot K\alpha^k) = \lambda_k(\alpha^k\cdot\alpha^k) \tag{8}$$

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

$$(\mathbf{V}^k \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{l} \alpha_i^k (\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x})) \tag{9}$$

Note that neither (5) nor (9) requires the $\Phi(\mathbf{x}_i)$ in explicit form. Therefore, we are able to use kernel function for computing without actually performing the map Φ. We can thus use the kernel function in Support Vector Machines [14]

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

Substituting kernel functions for all occurrences of $(\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$ then we can get Kernel PCA

B. Extreme Gradient Boosting (XGBoost)

Zięba et al. [15] proposed using extreme gradient boosting (XGBoost) to learn an ensemble of decision trees for bankruptcy prediction. Their so-called synthetic feature is comprised of various arithmetic operations such as addition, subtraction, multiplication, and division. Each synthetic feature can be treated as a regression model and constructed in an evolutionary way.

Denote $\mathbf{x} \in \mathcal{X}$ as one sample in the dataset where $\mathcal{X} \subset \mathbb{R}^D$ and $y \in \{0, 1\}$ as the label.We use Classification and Regression Trees(CART) as the base classifier model, which can be represented by the weights associated with the leaves in the structure.

$$f_k(\mathbf{x}_n) = w_{q(\mathbf{x})} \tag{11}$$

where $q(\mathbf{x})$ is the function which takes an example $\curvearrowleft$ and returns the path id in the structure of the tree, $q : \mathbb{R}^D \rightarrow \{1, ..., T\}$, T is the number of paths or leaves. A path is ended with a leaf containing weight w_i.

Then we construct an ensemble of K CART [16]

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

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 :

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

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

The model is trained by minimizing the following criterion:

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

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

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

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

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

Therefore, we can represent (14) as

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

The model can be also regularized by setting minimal number of examples combined with each of the leaves, by setting maximal depth of the tree, by setting the percentage of features randomized for each iteration of constructing the tree or by adding the new tree with corrected influence of the trees in the committee [15]

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

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

C. KP-XGBoost

Suppose the sample set is $X = \{x_1, ...x_m\}$,from (9) the project of sample point x_i in the new feature space $\mathcal{F}$ is $\mathbf{V} \cdot \Phi(\mathbf{x}_i)$ if we want to make the projects of all the sample points as disperse as possible, we are supposed to maximize the covariance in (2).This can thus be represented in the following manner

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

$$s.t. \mathbf{V}\mathbf{V}^T = I \tag{20}$$

We thus construct the Lagrange function

$$f(\mathbf{V}) = \mathbf{V}^T \Phi(X) \Phi(X)^T \mathbf{V} + \lambda(I - \mathbf{V}^T \mathbf{V}) \tag{21}$$

We take the partial derivatives respect to $\mathbf{V}$

$$\frac{\partial f}{\partial \mathbf{V}} = 2\Phi(X)\Phi(X)^T \mathbf{V} - 2\lambda \mathbf{V} \tag{22}$$

Let (22) equal 0 we can get

$$\Phi(X)\Phi(X)^T \mathbf{V} = \lambda \mathbf{V} \tag{23}$$

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

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

IV. Empirical analysis

A. Experimental setup

We conduct the following experiment to examine the performance of the proposed algorithm based on Kernel PCA and XGBoost. We compare the results from KP-XGBoost with four popular machine learning methods in fraud detection, including random forest(RF), logistic regression and support vector machine and the original XGBoost.Moreover, we first carry out experiments to examine the performance of PK-XGBoost. Namely, XGBoost exploits the features decomposed by Kernel PCA.The process of our experiment is shown in Fig. 1. The other four models are directly trained on the dataset without any decomposition and their performance are used as benchmark.

The popular 10-fold cross-validation approach is used for model evaluation and model selection to avoid over-fitting classifiers [17],in this paper we choose to use 5-fold instead of 10-fold due to the fact that we only have 50,000 records.This means, if we divide the raw data into 10 subsets, then each subset contain only 5,000 records which is relatively too small for training.

B. Data Description

In this paper we use a real-life dataset from one of the largest banks in China, where the fraudulent transactions are labelled 1. The dataset contains 50,000 records and its distribution is imbalanced with approximately 10,000 (around 20% fraudulent transactions) (see in Fig. 2)

Our data contain large amounts of missing values and noises, so we first process the data by dropping meaningless and strongly correlated features, filling in the missing value with column median, transforming multi-class features with one-hot encoding and employing the min-max scaling method so that we can achieve the processed data which is of vital importance for statistical models and further data analysis.

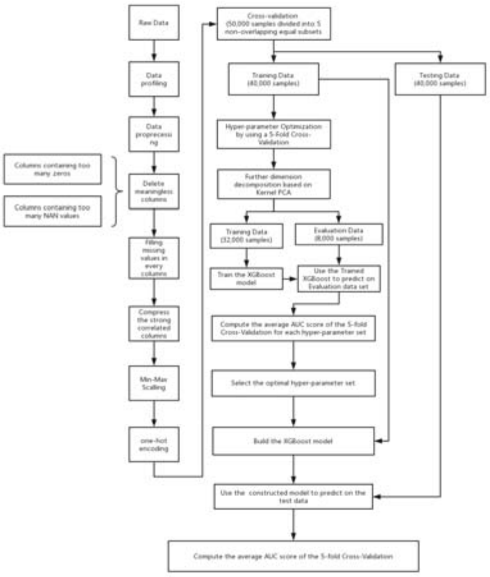
Raw Data
Data profiling
Data propreccessing
Columns containing too many zeros
Columns containing too many NAN values
Delete meaningless columns
Filling missing values in every column
Compress the strong correlated columns
Min-Max Scaling
one-hot encoding
Cross-validation (50,000 samples divided into 5 non-overlapping equal subsets
Training Data (40,000 samples)
Testing Data (40,000 samples)
Hyper-parameter Optimization by using a 5-Fold Cross-Validation
Further dimension decomposition based on Kernel PCA
Training Data (32,000 samples)
Evaluation Data (8,000 samples)
Train the XGBoost model
Use the Trained XGBoost to predict on Evaluation data set
Compute the average AUC score of the 5-fold Cross-Validation for each hyper-parameter set
Select the optimal hyper-parameter set
Build the XGBoost model
Use the constructed model to predict on the test data
Compute the average AUC score of the 5-fold Cross-Validation

Fig. 1. The PK-XGBoost algorithm

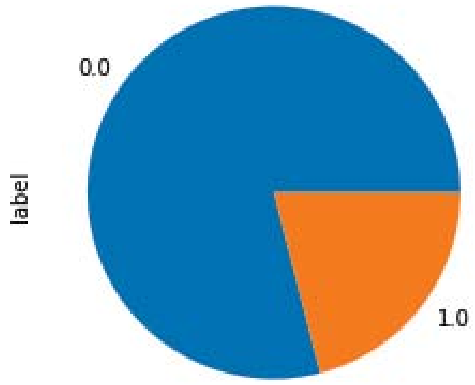

Fig. 2. The label distribution of dataset

C. Evaluation criteria

As it is mentioned in (IV-B), it one imbalanced classification task, we thus adopt the AUC (the area under the ROC curve) value as an overall performance measure. The AUC is considered to be a better overall performance measure than accuracy because it is independent of a cut-off value. [4] and it is a float value between 0 and 1, which means the closer AUC to 1 is, the better performance one model has.

D. Experimental results

The empirical experiment aims at answering two questions. One is that what the exact number of components is, that to say, how many components should be kept after Kernel PCA to achieve the best results.Normally, we keep the components with their accumulative contribution rate over 85% [18]. However, it is just from the voice of experience. Hence, we carry out grid search to determine the precise number of components which can help XGBoost achieve the best result.

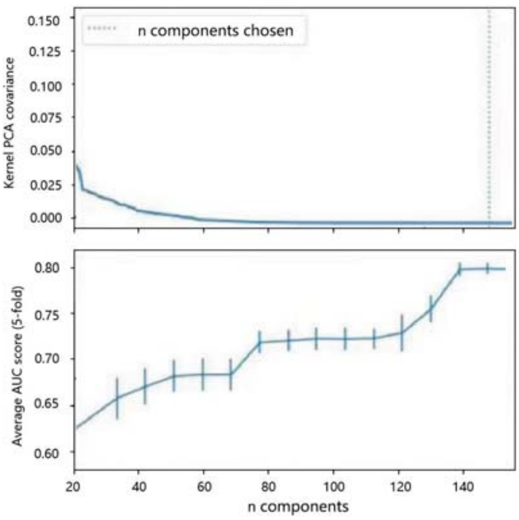

Fig. 3. Grid search on the number of components

From Fig. 3, we can see when we keep 150 components the PK-XGBoost can achieve the best AUC score(approximately 0.785)

Another question is whether our proposed algorithm can provide better fraud detection results. From Table I , we can see the results. Our proposed PK-XGBoost provides much better experimental results (compared with LR,RF SVM), and it also shows that Kernel PCA contributes to the performance of XGBoost, with the average AUC score of it 0.1 greater than that of XGBoost.

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

In summary, this paper proposes a new hybrid supervised and unsupervised learning method for credit card fraud detection. A data decomposition based on Kernel PCA projects and decomposes the feature variables for XGBoost to further detect the fraud behaviour. Compared with previous work, we take the privacy of clients into consideration. This is to say, we process the data without knowing the meaning of every feature, which poses great challenge to feature engineering.

We conduct comparative experiments to compare the performances of our proposed algorithm with traditional machine learning methods.

The empirical analysis is the carried out based on a real-life dataset.The results show that our proposed approach dramatically outperforms all the other classical machine learning methods and increases the performance of the original XGBoost. This provides an effective way to take both fraud detection and privacy protection into consideration.

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. Schölkopf, A. Smola, K. R. Müller, and G. Rätsch, "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, "High-dimensional 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.