**Journal of Computing and Electronic Information Management**
ISSN: 2413-1660 | Vol. 15, No. 3, 2024

# Insurance Fraud Prediction Model Based on eXtreme Gradient Boosting

**Mengda Lu***

School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, Jiangsu, 214122, China
* **Corresponding author Email:** 1193210411@stu.jiangnan.edu.cn

**Abstract:** As the volume of insurance transactions continues to increase, combating insurance fraud has become increasingly important. One of the key challenges is how to predict fraud based on high-dimensional data. In the current research landscape, deep learning methods that perform well often require large amounts of data, but their training and deployment also pose significant computational challenges. This paper, based on the Alibaba Tianchi dataset, explores the feasibility of constructing a low-cost prediction system through feature engineering and parameter tuning. Additionally, this paper verifies the robustness of the eXtreme Gradient Boosting (XGBoost) model in handling high-dimensional sparse features and imbalanced data.

**Keywords:** Auto Insurance; Fraud Prediction; XGBoost.

## 1. Introduction

As the economy rapidly develops, the number of vehicles in the country has also surged dramatically. Consequently, traffic accidents have increased, and auto insurance has become an effective means for car owners to mitigate the financial losses associated with these accidents. However, as insurance sales increase, so too does the incidence of insurance fraud, with fraudulent claims in some types of insurance accounting for up to 20% or more of total claim payouts[1]. Insurance fraud negatively impacts both the insurance industry and consumers. Therefore, predicting insurance fraud is crucial for maintaining the fairness and stability of the entire insurance system. However, relying on risk control experts for fraud prediction is costly, making the development of digital models for this purpose essential.

During insurance underwriting and claims processes, a large amount of data is collected, which can be used to construct models for clustering analysis using high-dimensional data. To address this issue, various predictive models have been proposed, including Bayesian modeling, clustering analysis, logistic regression, random forests, and neural networks [2, 3, 4, 5, 6]. Research on insurance fraud detection has evolved from traditional statistical regression to artificial intelligence. However, traditional machine learning algorithms have shown limited effectiveness in this area, and artificial intelligence algorithms require large datasets and significant computational resources for training and deployment. Optimizing traditional algorithms is a feasible approach to constructing low-cost predictive models.

This paper proposes a low-cost predictive model based on XGBoost. Through feature engineering and parameter tuning, an accurate predictive model was developed. The findings confirm that the XGBoost-based model saw a 14% gain in AUC compared to ensemble learning models constructed using SVM and logistic regression. The subsequent sections of this paper is organized as follows: Section 2 provides an overview of the pertinent research on traditional and deep learning algorithms in the field of auto insurance fraud prediction. Section 3 details the data processing, feature engineering, and model tuning processes. Section 4 will focus on presenting the results of the improved model proposed in this paper and compare them with other models. Additionally, it will discuss the practical significance of these findings.

## 2. Literature Review

Regarding anti-insurance fraud, scholars worldwide have made various explorations. Artis M [2] and colleagues utilized a nested logit model to predict car insurance fraud in Spain during 1995-1996. Their study found that the Multinomial Logit Model (MNL) performed well in predicting fraudulent behavior, achieving an overall correct classification rate of 76.3%. The Nested Multinomial Logit Model (NMNL) provided insights into the decision-making process, showing how different variables influenced various steps of the decision tree. In 2007, Caudill [3] introduced a new approach to detecting and re-evaluating fraudulent insurance claims using a multinomial logit model with missing information and the EM algorithm, successfully identifying claims that may have been misclassified.

In 2011, Ye Minghua[4] proposed integrating Logit regression metrics as refined explanatory variables into a BP neural network, combining the strengths of statistical regression and neural networks to predict motor vehicle insurance fraud. In 2018, Zhang Bo[5] compared various methods, including the Logit model, Bayesian network model, and decision tree model, achieving high accuracy in identifying insurance fraud. Jiaxi Yang and colleagues (2022) [6] proposed a multimodal learning framework (AIML) that integrates structured and unstructured data, using NLP and CV to extract information from unstructured data. The introduction of this model has significantly expanded the range of data sources that can be used for model construction. Compared to models that rely solely on structured data, this model has achieved a substantial improvement in predictive accuracy.

## 3. Method

This paper implements a prediction method grounded in the XGBoost model. The following sections will provide a detailed explanation of the specific work undertaken.

## 3.1. XGBoost

XGBoost (eXtreme Gradient Boosting) is a gradient boosting algorithm based on decision trees and is diffusely used in machine learning for classification and regression tasks. It boosts the model's predictive capability by merging several weak learners (in this case, decision trees), eventually forming a strong predictive model. The number of weak learners increases progressively during the iterative training process, with each iteration gradually reducing prediction error and improving model performance.

Suppose we have a target function $L(y, \hat{y})$ to measure the error between the model's prediction $\hat{y}$ and the true value y. In each round of training, the model uses the current residuals and trains another decision tree and adds the new prediction results to the overall model prediction to limit the error.

In the $t$-th iteration, the model's prediction can be expressed as:

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta \cdot f_t(x)$$

Where $\hat{y}^{(t-1)}$ is the model's prediction in the $t-1$ iteration, $f_t(x)$ is the decision tree generated in the

t-th iteration, and $\eta$ is the learning rate, which influences the extent of each tree's impact on the overall model.

XGBoost's objective function is structured into two sections: the loss function and the regularization term.

$$\text{Obj} = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{t=1}^{T} \Omega\left(f_t\right)$$

The difference between the predicted value $\hat{y}_i^{(t)}$ and the true value $y_i$ was reflected by the loss function $L(y_i, \hat{y}_i^{(t)})$. The model's complexity is controlled by the regularization term $\Omega(f_t)$, which is used as preventing overfitting. It can be expressed as:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2$$

where $T$ expresses the amount of leaf nodes in the tree, $\gamma$ controls the leaf node quantity, $\omega_j$ is the value of the j-th leaf node's weight, and $\lambda$ defines the regularization applied to the weights.

To achieve optimality in the objective function, XGBoost employs a second-order Taylor expansion to approximate the loss function and uses a greedy algorithm to select the optimal split point. The second-order Taylor expansion of the loss function can be expressed as:

$$L\left(y_i, \hat{y}_i^{(t)}\right) \approx L\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)$$

Where $g_i = \frac{\vartheta L\left(y_i, \hat{y}_i^{(t-1)}\right)}{\vartheta \hat{y}_i^{(t-1)}}$ is the first derivative; $h_i = \frac{\partial^2 L\left(y_i, \widehat{y_i^{(t-1)}}\right)}{\partial \widehat{y_i^{(t-1)^2}}}$ is the second derivative.

When constructing a new tree, the algorithm calculates the gain for each candidate split point and selects the split point with the highest gain to perform the split. This process is repeated until the maximum tree depth or other stopping criteria are met. The gain is calculated using the following formula:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma$$

In the formula, $G$ represents the first-order derivative, and $H$ represents the second-order derivative. $G_L$ stands for the combined total of gradients for the left subtree, while $G_R$ equals the sum of gradients for the right subtree. $H_L$ represents the sum of the second-order derivatives for the left subtree, and $H_R$ represents the sum of the second-order derivatives for the right subtree.

## 3.2. Data Processing and Feature Engineering

The data used in this paper comes from the Alibaba Cloud Tianchi platform. The dataset provides historical car insurance claim data from previous customers, including features such as age, education level, occupation, and the severity of the accident. The `train.csv` file contains 700 records and includes the ground truth labels indicating whether each claim is fraudulent. Subsequent experiments will be based on this dataset.

The provided data has already undergone data cleaning and missing value imputation. To reduce data volume and improve computational efficiency, we first performed a correlation analysis on all numerical variables in the dataset. The results, as shown in Figure 1, indicate a marked correlation between total claim amount and injury claim amount, property claim amount, and vehicle claim amount. Similarly, a high correlation was found between age and the duration of being a customer. When training the model, we consider selecting one variable from each highly correlated group to represent the entire set of related variables.

For non-numeric variables, we used one-hot encoding to convert each categorical column into integer values, making them suitable for subsequent use in machine learning models.

The dataset was then split, with 20% of which assigned for testing purposes to confirm the model's performance. The dataset was also standardized by scaling the feature data to a range with zero mean and unit variance. This standardization is crucial for improving the performance of machine learning models, particularly for gradient descent-based algorithms, as it accelerates the model's convergence and prevents certain features from dominating the entire model.

## 3.3. Model Training and Hyperparameter Tuning

In this study, the XGBoost classifier's evaluation metric is specified as logarithmic loss (log loss), which is a typical measure for binary classification task evaluation. The Randomized Search method is employed to find the optimal hyperparameters for the model. This method randomly selects and combines hyperparameters from a specified range and then trains and validates the model. The hyperparameters considered for optimization in this experiment include "n_estimators," "learning_rate," "max_depth," "subsample," "colsample_bytree," and "gamma," with the AUC score used as the evaluation metric. The meanings of these parameters are as follows:

- n_estimators: The number of trees, or the number of iterations.
- learning_rate: The learning rate, used to reduce each tree's contribution to the model.
- max_depth: The greatest depth of the trees, controlling the model's complexity.
- subsample: The proportion of samples used for training weak learners.
- colsample_bytree: The share of features utilized for training weak learners, which affects the dimensionality of the data fed into the weak learners.
- gamma: Determines whether a node will split; if the

reduction in the loss function from the split is less than this value, the split will not occur.

The learning rate ensures that the role of each tree in determining the final prediction is small, thereby improving the overall model's performance. When updating the leaf node weights, the learning rate is multiplied by a coefficient to avoid overly large steps. A larger learning rate speeds up the training process but may prevent the model from converging. Conversely, a smaller learning rate captures more subtle patterns but slows down the convergence speed.

Parameters like max_depth, subsample, and colsample_bytree help prevent overfitting. A larger max_depth allows the model to learn more specific and local trends in the data, which can lead to overfitting if the value is too high, or underfitting if too low. Subsample represents the subsampling ratio, influencing the number of samples fed into the subtrees. The value indicates the proportion of the total training set samples used by the subtrees, ranging from 0 to 1. Colsample_bytree represents feature subsampling, affecting the number of features fed into the subtrees. The value indicates the proportion of the total feature dimensions used by the subtrees, ranging from 0 to 1. Gamma acts as a regularization parameter, specifying the least reduction in loss required to trigger a node split. A larger gamma value enhances the algorithm's conservativeness, only allowing splits that result in significant loss reduction.

Finally, a suitable combination of hyperparameters was obtained and used for model training.

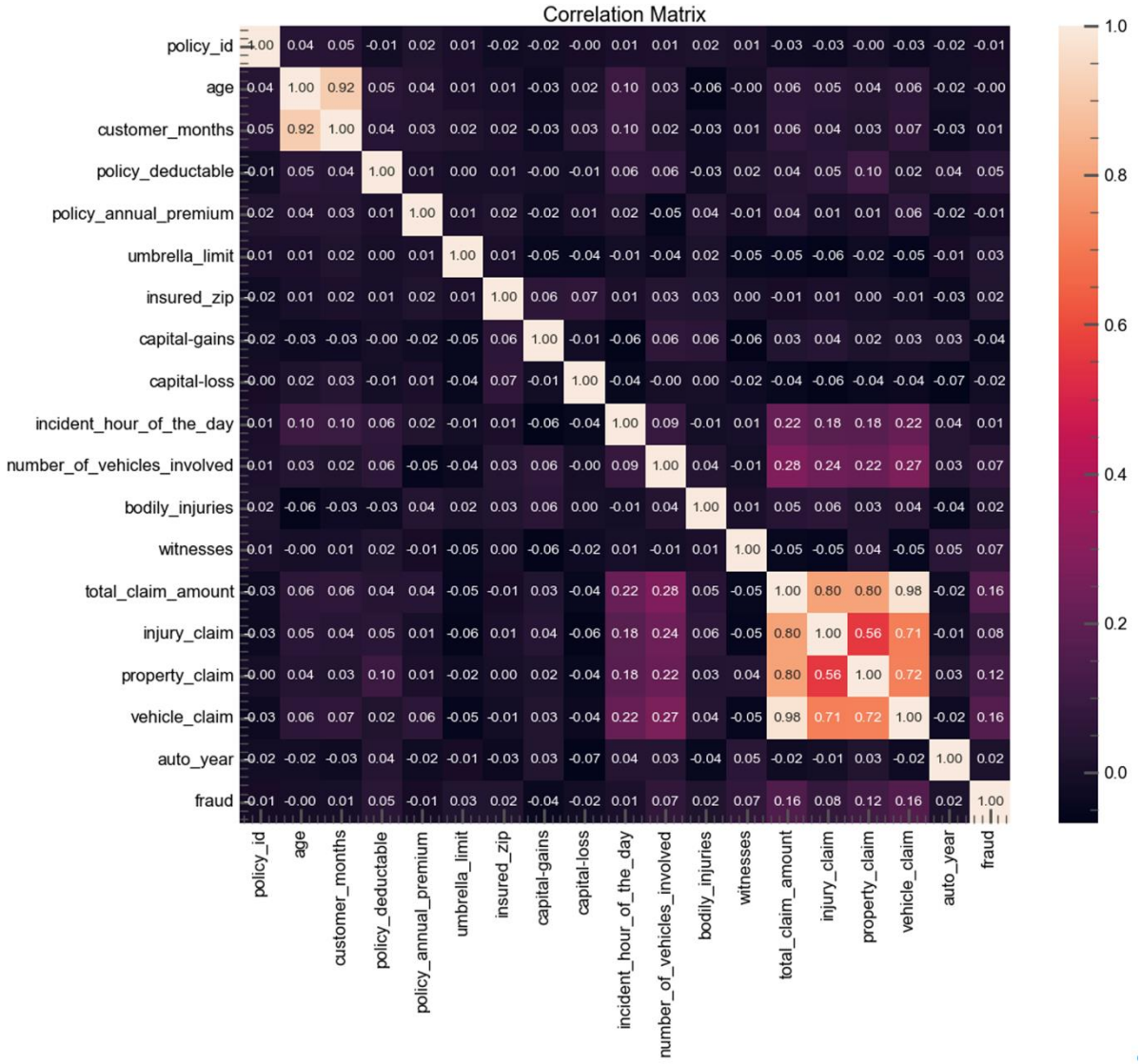


**Figure 1.** Variable Correlation Analysis.

## 3.4. Evaluation Metrics

In the field of machine learning, several algorithms are used to evaluate model performance, including Accuracy, Precision, Recall, F1 Score, and AUC. Accuracy measures the overall correctness of the model and is computed using the formula below:

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN}$$

In the above formula, the value of TP (True Positive) refers to the number of samples in cases where the actual value is positive and the prediction from the model is likewise positive. The TN (True Negative) value expresses the amount of samples when both the observed value and the model's output are negative. FP (False Positive) refers to the number of samples when the actual value is negative, and the model's prediction incorrectly indicates a positive value. FN (False Negative) indicates the quantity of samples where the actual positive value is misclassified as negative. TP and TN shows the total of samples correctly classified by the model, while FP and FN corresponds to the count of samples incorrectly classified by the model.

Precision measures the accuracy of the model's positive predictions and is calculated using the following formula:

$$\text{Precision}=\frac{\text{TP}}{\text{TP+FP}}$$

Recall assesses how well the model identifies positive cases, representing the proportion of correctly identified positive samples, and is computed using the following formula:

$$\text{Recall}=\frac{\text{TP}}{\text{TP+FN}}$$

F1-Score is defined as the harmonic average of Precision and Recall, suitable for cases with imbalanced classes, and is calculated using the following formula:

$$\text{F1-Score}=2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}=\frac{\text{TP}}{\text{TP}+\frac{1}{2}(\text{FP+FN})}$$

The ROC curve describes the trend of the False Positive Rate (FPR) as the True Positive Rate (TPR, i.e., Recall) changes. The position of each point on the curve is determined by the FPR and TPR values of the model's prediction results when a specific threshold is selected. In each iteration of the calculation, data with model output values under the specified threshold are treated as negative, while those values that reach or surpass the threshold are interpreted as positive. The x-axis is established by the following equation $\text{FPR}=\frac{\text{FP}}{\text{FP+TN}}$, and the y-axis is established by the equation $\text{TPR}=\frac{\text{TP}}{\text{TP+FN}}$. An ideal classifier should have a high TPR with a low FPR, making the ROC curve as close as possible to the top left corner. AUC measures the area under the ROC curve and examines the classifier's effectiveness as a whole. As the value approaches 1, the classifier's performance improves.

# 4. Conclusion

Training and evaluation were performed on the XGBoost classifier using the optimal parameters identified earlier. The model's classification performance on the training set is shown in Figure 2.

The model fits the training data well, achieving an AUC value of 0.9864. Additionally, the recall was 0.8844, the F1 Score was 0.9220, the Accuracy was 0.9607, and the Precision was 0.9630. The confusion matrix indicated a low level of incorrect predictions.

The model's performance on the test set is shown in Figure 3. On this data, the model achieved an AUC value of 0.8628. Observing the confusion matrix, it is evident that the model successfully accomplished the prediction task.

In comparison, the ensemble learning model using SVM and logistic regression had an AUC value of 0.72 on the test set, as shown in Figure 4. The ensemble model performed poorly on this problem, with a significant number of fraudulent samples not being correctly predicted, achieving a recall rate of only 0.0923.

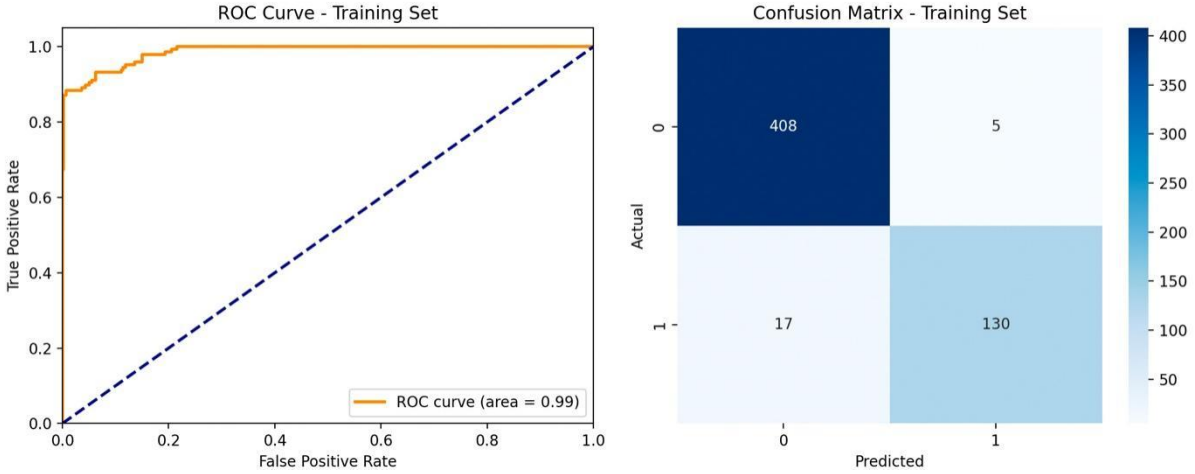


**Figure 2.** XGBoost Performance On Training Set.

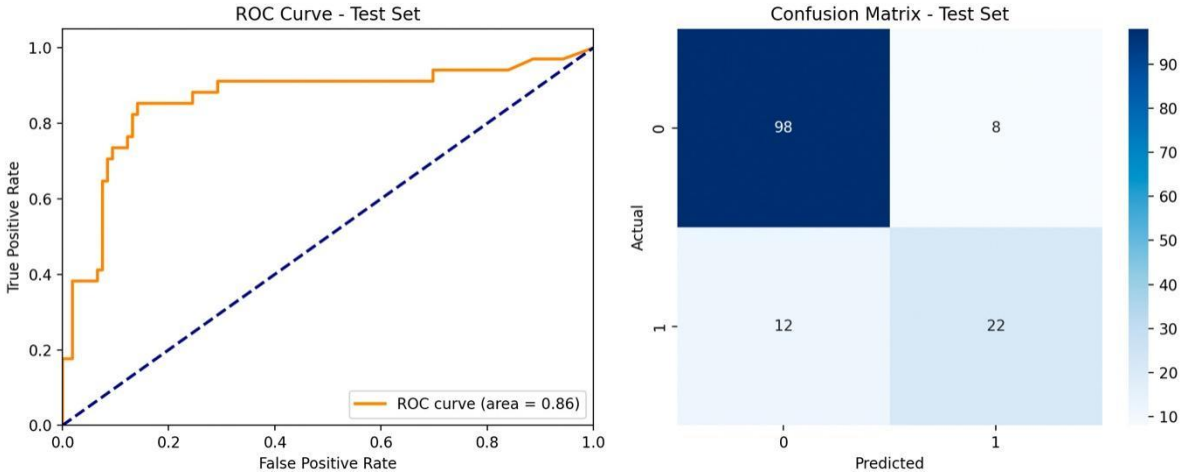


**Figure 3.** XGBoost Performance On Test Set.

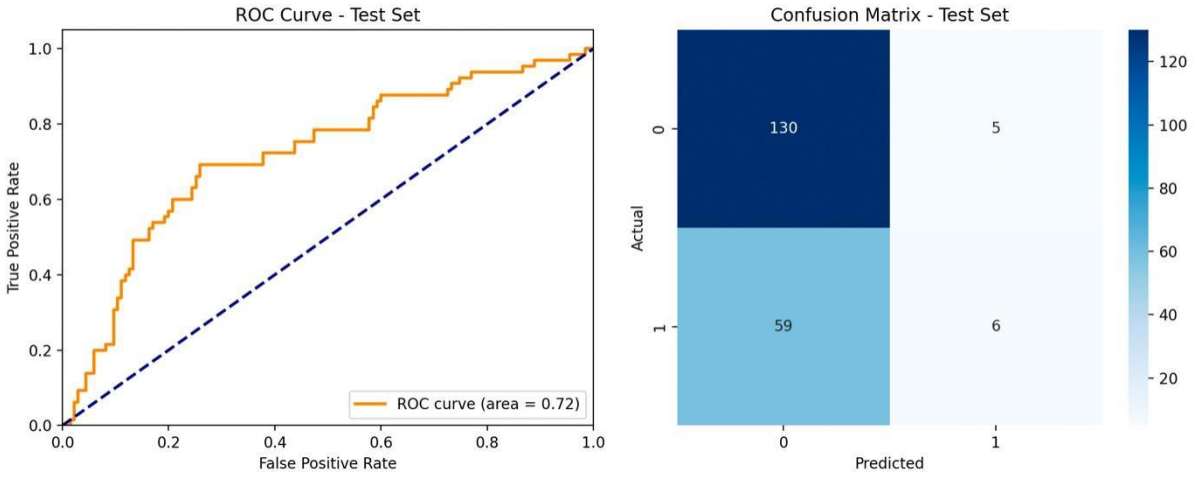


**Figure 4.** Performance of the Ensemble Model Built with SVM and Logistic Regression on the Test Set.

The fraud prediction model based on XGBoost proposed in this paper demonstrated strong performance, particularly when data and computational resources were relatively limited. Despite consuming fewer resources during training, the model still performed well on test data, accurately identifying potential fraudulent activities. This indicates that the XGBoost algorithm is both practical and efficient in addressing fraud detection issues, making it suitable for application in various real-world scenarios.

This study optimized XGBoost to operate efficiently even when data and computational resources are constrained, providing a reference framework for future research on how to implement effective machine learning models under resource limitations. The study demonstrates the robustness of XGBoost in handling high-dimensional sparse features and imbalanced data, offering new insights into feature engineering and model generalization for fraud detection. Additionally, the tree structure of the XGBoost model provides a degree of interpretability, offering potential for deeper understanding of fraud patterns and aiding in the explanation of prediction outcomes.

In practical applications, many businesses and organizations may face challenges of insufficient data or limited computational resources. The model proposed in this paper proves that even under such constraints, effective fraud detection can be achieved through appropriate algorithm selection, offering a cost-effective solution for similar real-world scenarios. Due to the model's low computational requirements, it can be more easily deployed in real-time systems for online fraud detection on large-scale transaction data. This efficient application can significantly enhance security in the financial and e-commerce sectors.

# References

[1] Yu, W., Feng, G., Zhang, W. (2017) A Research on Fraud Detection System and Gang Identification of Vehicle Insurance. Insurance Studies, 2: 63–73

[2] Artís, M., Ayuso, M., Guillén, M. (1999) Modelling different types of automobile insurance fraud behaviour in the Spanish market. INSURANCE MATHEMATICS & ECONOMICS, 24: 67-81

[3] Caudill, SB., Ayuso, M., Guillén, M. (2005) Fraud detection using a multinomial logit model with missing information. JOURNAL OF RISK AND INSURANCE, 72: 539-550

[4] Ye, MH. (2011) Insurance frauds identification research based on the BP neurological network--with China motor insurance claim as an example. Insurance Studies, 3: 79-86

[5] Zhang, B. (2018) Master of Anti-fraud research on auto insurance claims based on data mining technology (academic dissertation, University of International Business and Economics). master.

https://kns.cnki.net/kcms2/article/abstract?v=sxrP1m9hSI_o8nyh8xZTi73Cl03w8G2OyLb-t94gIvbZ9BMr15IKLycAgGQdEcctrqv8FJ50sbzKvSPK3jLNoXqmQUgMVqExZo2WKLZPFz_DJgiglXNnIXuY6ZdVzhGkq-YJI5sFEcKDBBKi5NfIMUDRNa6-kQ8J_Rzo12WUe-XPe39pAlnNc9bPuEp0VP9XT93qeA1CRKo=&uniplatform=NZKPT&language=CHS

[6] Yang, JX., Chen, K., Ding, K., Na, CN., Wang, M. (2023) Auto Insurance Fraud Detection with Multimodal Learning. DATA INTELLIGENCE, 5: 388-412