Analyzing Fraud Rate Based on Insurance Contract Reports Using Meta-Deep Stacking Approach

Abstract

Insurance fraud in automobiles poses significant challenges for insurers globally. Policyholders engage in fraudulent activities like document falsification to unlawfully obtain funds, causing substantial financial losses. Many insurers rely on traditional methods, insufficient in detecting fraud, especially in complex schemes or high claim volumes. This study aims to develop an effective fraud detection model tailored to real-world data using machine learning algorithms. The dataset includes 1000 insurance claims from car collisions across seven US states in 2015. Results show that a meta-deep stacking model, integrating SVM and Random Forest, outperforms traditional models. Building on these findings, the objective is to improve fraud detection in the Vietnamese insurance industry, reducing contract fraud. Future research will focus on leveraging deep learning models globally and utilizing diverse datasets to refine the model further.

Keywords: Insurance fraud, fraud prevention efforts, meta-deep stacking model

1 Introduction

Insurance is a contract in which an insurance company commits, in exchange for a fee, to provide cash insurance to the insured party in the event of losses resulting from uncertain events, after an official claim is submitted by the claimant [1]. Automobile insurance is a type of contract in which the insurance company compensates the insured for losses resulting from vehicle-related events, including accidents, theft, and damage from various hazards. It covers legal liability, vehicle damage, and sometimes medical expenses, with specific insurance coverage details varying depending on policy and location. The demand for automobile insurance increases with the rising number of vehicles on the road because insurance provides financial protection against driving-related risks such as accidents, theft, and damage. However, as the auto insurance industry develops, it becomes a more common target for fraudulent activities. This includes falsified or exaggerated claims for financial gain, affecting both insurance companies and honest policyholders [2, 3]. Models for diagnosing insurance claim fraud have been developed to combat this issue. However, for imbalanced datasets, this poses a challenge for traditional models. To address this, we develop a meta-deep model to select and combine effective models using a stacking approach that improves project performance shown in Fig.1. Results indicate that the proposed model operates efficiently and demonstrates stability.

The paper will be divided into four main sections: introduction, proposed method, experiment, and conclusion. The introduction will provide an overview of the content and objectives of the study. The proposed method section will detail the proposed model. The experiment section will present specific experimental strategies and results. The conclusion will summarize the entire research process in the paper.

2 Proposed method

2.1 Overview

Traditional models, although consistently achieving high performance with small datasets, encounter several challenges due to the constant need for feature selection and perform poorly on datasets with a large number of observations. Therefore, a significant challenge for models like Random Forest and SVM is the requirement for complex data processing steps [4], which slows down progress and desired performance when handling datasets of varying sizes. In contrast, for deep learning neural network models, processing large-sized data is not difficult and does not require complex data transformation or feature selection, but they are prone to instability when working with small datasets.

Hence, we decided to create a meta-deep model that combines both of these model groups shown in Fig.1. Specifically, the proposed model will be constructed from two main baseline models: the group of traditional models and a meta-deep model. We employ an ensemble stacking approach, a form of early fusion method, to combine all the output results of the baseline models. Subsequently, they are stacked vertically to serve as input for the meta-deep model. Through the training processes, we strengthen the advantages of each model group and gradually optimize their limitations.

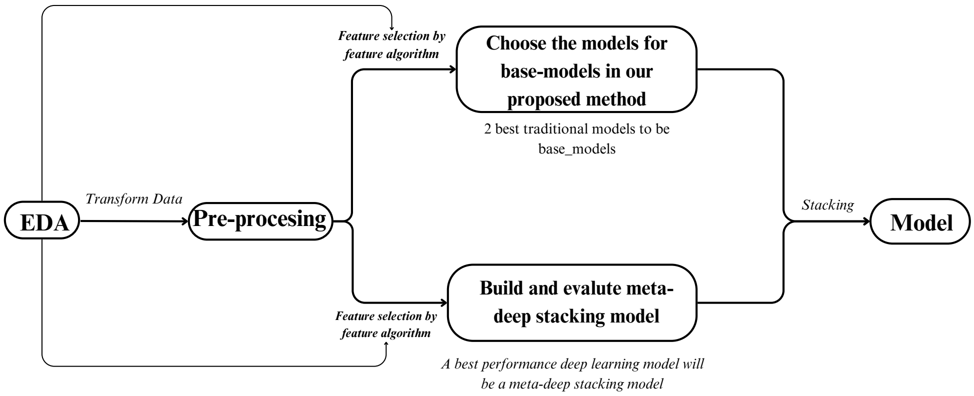

Fig. 1 Our proposed method

2.2 Details

Let X be the feature matrix of automobile insurance claim requests, with a size of D×N, where D represents the number of features and N is the number of samples. Let Y be a vector of size N×1 containing the actual labels of the samples (1 for fraud, 0 for non-fraud).

We can construct a meta-deep model from conventional deep learning models in image using Conv1D, Batch-normalization, and Dropout. It is used to predict the likelihood of fraud in automobile insurance claims as follows:

Step 1: Train traditional base models: Each base model will be trained on the feature matrix X and predict on the test or validation dataset. The prediction results will form a vector p_i of size $N \times 1$, where i is the index of the base model. In total, we have M base models, resulting in M prediction vectors: p_1 ,p_2 ,...,p_M.

Step 2: Evaluate meta-deep stacking model: each meta-deep model will be trained on the feature matrix X and predict on the test or validation dataset. We will choose the best model for stacking approach.

Step 3: Train meta-deep stacking model: The meta-deep stacking model (e.g., a logistic regression model) will be trained on the prediction matrix P to predict the final likelihood of fraud. Based on this structure, the prediction formula [5] can be represented as follows:

$$\hat{Y} = \sigma\left(\sum_{i=1}^{M} \omega_i P_i\right) \tag{1}$$

where $\hat{Y}$ is the final prediction vector of the Stacking Ensemble model (predicted likelihood of fraud). $\sigma\,(.)$ is the activation function (e.g., sigmoid function in the case of logistic regression model). ω_i is the weight vector of size M x 1 used to combine predictions from base models. P_i is the prediction of the i-th base model.

3 Experiment

3.1 Experiments setup

Dataset. Our study utilized a dataset from a competition held on Kaggle [6], which garnered significant attention from the global scientific community. This dataset includes about 1000 insurance claims for various car accidents across 7 states in the USA in 2015. However, a portion of these claims was identified as fraudulent and subsequently denied. The information and aspects targeted by the dataset are easily extracted from insurance companies, such as contract-related information, collision details, etc. During our exploration, we noticed several underlying issues with the dataset, such as the imbalance of labels deemed fraudulent compared to other labels, and the excessive number of observed variables while lacking correlation among them. From this, it is clear that the dataset is highly suitable for the topic, aiming to increase the reliability of the model through the complexity of the data.

Data Enhancement. During the experiment, we will split the dataset into a training set and a test set with the common ratio of 66.7/33.3. It is observed that the labels are uneven, with label 1, which indicates contracts considered fraudulent, accounting for less than 30%. The input data will be augmented using the SMOTE method aimed at enhancing labels of class 1 to improve the model's performance, helping the model to perform better [7].

Parameter Tuning. To select appropriate parameters for each sub-model to achieve the best effectiveness [8], we conducted parameter tuning techniques for both traditional models and deep learning models that we experimented with. The results of the parameter tuning process were directly applied to the research topic.

Training process. It was realized that when the number of epochs was limited to around 120, although data augmentation had been performed for the minority class, the model still predicted outputs as the majority class, which pushed the accuracy performance up to 73.3% but in reality, the model could not detect any fraudulent contracts. Therefore, we increased the number of epochs to 600. We used Adam as the optimization function for the topic with an initial learning rate set at 0.0001. The study also applied the Learning rate schedule technique after every 20 epochs without performance improvement with a minimum learning rate of 0.00001, along with Early Stopping to halt the training process after 30 epochs without any improvement. The entire experiment was conducted on an AMD Ryzen 7 6800H with Radeon Graphics CPU, along with Keras and Scikit-learn libraries.

Loss and Evaluation Metrics. The Binary Cross-Entropy loss function is a commonly used loss function in our paper. This function measures the difference between two probability distributions, including the predicted distribution and the actual distribution. The formula for Binary Cross-Entropy is presented as follows:

$$L\left(y_{true}, y_{pred}\right) = -\frac{1}{N}\sum_{i=1}^{N} y_{true}\log y_{pred} + \left(1 - y_{true}\right)\log\left(1 - y_{pred}\right) \tag{2}$$

where N is the number of samples in the dataset, y_{true} is the actual label of the i-th sample (with a value of 0 or 1), and y_{pred} is the predicted probability of the i-th sample belonging to class 1.

Thus, this function calculates the sum of the loss values for each sample, where the loss value is calculated using the logarithm of the predicted probability corresponding to the actual label. Additionally, as mentioned before, relying solely on Accuracy could lead to misleading evaluation results, making the training process ineffective. Therefore, we also use Recall, aiming to increase the ability to detect fraudulent contracts in reality. The purpose of using both of these metrics is to evaluate the model achieving the highest performance while proposing the most effective method in classifying fraudulent contracts [9].

3.2 Result

After evaluating to select suitable traditional models shown in Table.1, we decided to choose SVM (Support Vector Machine) and Random Forest as their performance based on the assessment was truly impressive. Having demonstrated the effectiveness of these two models, we proceeded to use parameter tunning methods to create the most appropriate model for the topic.

Table 1 Experiments on traditional models

	Accuracy	**Recall**	**F1 Score**
SVM	76.5%	75.1%	80.0%
Random Forest	80.8%	**81.0%**	**80.8%**
Linear Discriminant Analysis	74.4%	69.5%	70.1%
Decision Tree	**82.0%**	73.0%	74.0%

Moving to the meta-deep stacking model, we decided to build and test conventional deep learning models. Initially, the model showed poor effectiveness when operating on this dataset. However, after applying tuning techniques above, the model also performed relatively well in Table. 2.

The final part involves constructing the proposed model of the project, using the Stacking technique. We built it using SVM and Random Forest as base models and the deep learning model we tested as the main model, as presented in the Proposed Method section. Through the training process, the results showed that the model was extremely effective on this dataset, with the fraud detection rate reaching about 84%. This indicates that our method has achieved quite good performance in Table. 3.

Table 2 Experiments on meta-deep stacking models

	Accuracy
InceptionV2	**75,45%**
InceptionV3	74,55%
MobileNet	69,70%
Restnet50	72,73%
VGG16	70,61%
VGG19	74,85%
Neural Network	70,20%

Table 3 Experiments on stacking approach

Approach	**Meta-Model**	**Models**	**Accuracy**	**Recall**	**F1 Score**
Voting		SVM, RandomForest	82.7%	81.8%	83.0%
Stacking	InceptionV2	SVM, Random Forest	83.3%	85.2%	**84.0%**

4 Conclusion

In this study, we proposed an advanced model compared to traditional methods for predicting insurance fraud. We use a fraud dataset [6], which closely resembles real-world data, and discover deep-seated issues such as imbalance and lack of correlation in the data. After exploring and processing the data, our project had stages: to evaluate traditional models, to enhance results using the meta-deep model with stacking approach. With the complexity of the dataset, we have achieved a performance of 84%, demonstrating the effective operation of the project. This archieved not only addresses the common instability issues encountered by deep learning models but also enhances performance, enabling the model to operate effectively with complex and extremely large datasets. It proved to be more efficient and optimized compared to traditional financial learning methods.

Future research directions will focus on optimizing the model and improving its performance. Additionally, a more sophisticated Keras network model will be proposed to adapt well to different datasets influencing various aspects of fraud conditions.