Comparison of Tree-Based Learning Methods for Fraud Detection in Motor Insurance

David Paul Suda[a], Mark Anthony Caruana[b] and Lorin Grima
Department of Statistics and Operations Research, Faculty of Science, University of Malta, Msida, Malta

Keywords: Insurance Fraud Detection, Random Forests, Gradient Boosting, Data Imbalance.

Abstract: Fraud detection in motor insurance is investigated with the implementation and comparison of various tree-based learning methods subject to different data balancing approaches. A dataset obtained from the insurance industry will be used. The focus is on decision trees, random forests, gradient boosting machines, light gradient boosting machines and XGBoost. Due to the highly imbalanced nature of our dataset, synthetic minority oversampling and cost-sensitive learning approaches will be used to address this issue. A study aimed at comparing the two data-balancing approaches is novel in literature, and this study concludes that cost-sensitive learning is overall superior for this application. The light gradient boosting machine using cost-sensitive learning is the most effective method, achieving a balanced accuracy of 81% and successfully identifying 83% of fraudulent cases. For the most successful approach, the primary insights into the most important features are provided. The findings derived from this study provide a useful evaluation into the suitability of tree-based learners in the field of insurance fraud detection, and also contribute to the current development of useful tools for correct classification and the important features to be addressed.

1 INTRODUCTION

Motor insurance is an essential component of the automotive industry, as it offers financial protection to vehicle owners against potential losses stemming from accidents, theft, or other unexpected incidents. As the auto insurance market continues to expand, it inevitably attracts fraudulent activities, leading to a surge in the number of motor insurance fraud cases (Hashmi et al., 2018). Insurance fraud is a deceptive practice that often carries the false perception of being a victimless crime. In reality, it adversely affects not only the insurance industry, but all policyholders. When insurance companies incur losses due to fraudulent claims, they often have to increase premiums to compensate for the financial impact. This results in higher costs for all policyholders, including those who have never engaged in fraudulent activities.

In recent years, motor insurance fraud has become a significant concern, with statistics demonstrating the growing scale of the problem. According to the Federal Bureau of Investigation, in the United States, insurance fraud, excluding health insurance, amounts to approximately $40 billion per year, increasing the average family's insurance premium by $400 to $700 annually, (Gomes et al., 2021). This highlights the significant impact of motor insurance fraud on the industry and policyholders, necessitating effective fraud detection and prevention strategies to protect honest policyholders and create a sustainable market (Hargreaves and Singhania, 2015).

Traditional fraud detection methods, which relied heavily on extensive auditing and manual investigation, have been demonstrated to be both costly and inefficient (Nian et al., 2016). As a result, insurance companies are increasingly adopting statistical and data analysis techniques to enhance their fraud detection capabilities (Kemp, 2010). Statistical learning theory emerged in the 1960s, but practical algorithms developed in the 1990s (Vapnik, 2000). These advanced methods offer innovative fraud detection solutions, aiding insurers in mitigating losses and protecting policyholders and offer promising solutions for fraud detection, surpassing classical statistical methods like logistic regression. (Aslam et al., 2022), (Al-Hashedi and Magalingam, 2021)), (Phua et al., 2004) introduced a novel fraud detection method for skewed data, employing NN, Naïve Bayes (NB) and DT algorithms on minority oversampled data. The

[a] https://orcid.org/0000-0003-0106-7947
[b] https://orcid.org/0000-0002-9033-1481

Suda, D. P., Caruana, M. A., Grima and L.
Comparison of Tree-Based Learning Methods for Fraud Detection in Motor Insurance.
DOI: 10.5220/0013513900003967
In *Proceedings of the 14th International Conference on Data Science, Technology and Applications (DATA 2025)*, pages 390-397
ISBN: 978-989-758-758-0; ISSN: 2184-285X

approach combines stacking and bagging to enhance cost savings, using a fixed cost matrix. Stacking-bagging techniques achieved marginally higher cost savings compared to other widely used techniques. (Bhattacharyya et al., 2011) used logistic regression, support vector machines and random forests to detect credit card fraud .

With the advent of big data, a number of reviews demonstrate that classification algorithms, particularly supervised techniques, have been widely utilized in motor insurance fraud detection. The review by (Ngai et al., 2011) has shown that while logit and probit regression models remain popular, more complex algorithmic methods like neural networks, tree-based methods, and Bayesian belief networks are increasingly being used. This trend is further corroborated by the more recent review from (Al-Hashedi and Magalingam, 2021), which highlights the growing prominence of statistical learning techniques such as random forest, naïve Bayes, support vector machines, K-Nearest Neighbour (KNN) and Gradient Boosting Machine (GBM) learners. Furthermore, the imbalance problem is well described by (He and Garcia, 2009), who identified cost-sensitive learning and synthetic minority oversampling methods as viable solutions.

The main aim of this study is to focus on a number of tree-based learning classification methods to a data set provided by an anonymous insurance company which contains 159045 countrywide insurance claims of which 3199 (2.01%) are fraudulent. The original dataset has 95 variables (including the target variable) which include, age and gender of claimant, date and time of accident, and province of claim and policy holder - however information such as the provenance of this dataset and the associated misclassification costs cannot be published due to identifiability reasons and commercial sensitivity. This dataset was collected between 2011 and 2015. Tree-based methods were found to be among the most popular techniques in fraud-detection literature due to their superior classification abilities. In this study, apart from comparing the tree-based classification approaches, the aim is to address data imbalance via both cost-sensitive learning approaches and synthetic minority oversampling, and finally provide a comparison to identify which tree-based learner combined with which learning approach is most promising. Which of the mentioned approaches for data imbalance is more successful will also be postulated.

The rest of this paper is structured as follows. In Section 2, the core concepts and characteristics of decision trees are discussed, together with the tree-based learning techniques which will be used. In Section 3, the results are presented, where a comparative study of the techniques and learning approaches described is presented, and a discussion of the more prominent features of the most successful method included. Finally, in Section 4, a discussion of the results will ensue with concluding remarks on the study and an overview of limitations together with some recommendations for future work.

2 METHODOLOGY

Statistical learning methods bring together a range of techniques and algorithms, all designed to learn from input data, with the goal of emulating human learning and making predictions. In this paper we will primarily focus on supervised learning techniques and thus we define the input space $\mathcal{X}$ as a set of all possible instances that need to be labelled from the output space $\mathcal{Y}$. In the context of motor insurance fraud detection, the instances are claims and have to be labelled as 'fraud' or 'not-fraud'. Let $\mathbf{x}^{(j)} = (x_1^{(j)}, \ldots, x_p^{(j)})$ be the vector of observed entries of the input space of the j^{th} observation with corresponding observation $y^{(j)}$ from the output space. In our case, the output space is a binary categorical variable taking values from $\{0,1\}$, where 0 implies that the claim is legitimate, while 1 implies that the claim is fraudulent. In supervised learning, we split the given data set in two: the training set and the test set. In the following pages, the training set will be dented by $D = \{\mathbf{x}^{(j)}, y^{(j)}\}_{j=1}^{N}$, where N represents the number of claims in the training set. The performance of the models with be tested through the use of the confusion matrix and related metrics on the test set. A 90-10 split ratio is taken , given the size of the dataset, 10%still yields a sufficiently large test set for evaluative purposes. Correct predictions of no fraud will be considered as true negative (TN) while correct predictions of fraud will be considered as true positive (TP). Incorrect predictions of no fraud and incorrect predictions of fraud will be considered false positive (FP) and false negative (FN) respectively.

The rest of this section will be structured as follows. In Section 2.1 will discuss the data imbalance issue, and how we can address it through synthetic minority oversampling and cost-sensitive learning. In Section 2.2 will introduce decision trees - the base learner for all techniques which will be covered in this paper. Finally, in Section 2.3, tree-based learning techniques such as random forests and boosted trees will be discussed.

2.1 Addressing Data Imbalance

Data imbalance will be an issue with our dataset due to the fact that non-fraudulent claims compose the overwhelming majority. SMOTE-NC by (Chawla et al., 2002) is a synthetic minority oversampling technique that aims to address the issue of imbalanced datasets by generating synthetic samples from the minority class. SMOTE-NC is a variant of the SMOTE technique with modifications to account for nominal features. However, in the context of motor insurance fraud, a crucial aspect of the model's performance evaluation is to consider the costs of different misclassification errors. Specifically, FNs can have substantially higher consequences than FPs. In this case, a standard loss function, such as the 0-1 loss function, may not accurately reflect the true risk of the model's predictions. To address this, a cost-sensitive loss function could be used to take into account the different costs of these errors. The focus of statistical learning, under a cost-sensitive loss function, shifts to minimizing the total cost. Let the subsets of fraudulent and legitimate training samples be denoted as D^+ and D^- respectively. Adjusting the 0-1 loss function to cater for this cost factor, results in the following empirical risk of some classifier $\hat{f}$:

$$\hat{R}(\hat{f}) = \frac{a}{N} \sum_{\mathbf{x}^{(i)} \in D^+} 1_{\{f(\mathbf{x}^{(i)}) \neq \hat{f}(\mathbf{x}^{(i)})\}} + \frac{1}{N} \sum_{\mathbf{x}^{(i)} \in D^-} 1_{\{f(\mathbf{x}^{(i)}) \neq \hat{f}(\mathbf{x}^{(i)})\}} \quad (1)$$

where a indicates the cost-sensitive factor (He and Garcia, 2009). If the cost-sensitive factor $a > 1$, it os apparent that a false negative outcome will result in a higher loss. On the other hand, if $a < 1$, a false positive outcome will result in a greater loss. In the insurance fraud context, a is taken to be a ratio of the average fraudulent claim loss to the cost of investigating a claim and will be greater than 1 in our context. In Section 3, we will compare the results for various tree-based learners using both the cost-sensitive loss function and the synthetic minority oversampling technique SMOTE-NC.

2.2 Decision Trees

The concept behind decision trees is intuitive when dealing with a classification problem. The algorithm uses a tree-like framework to make predictions by dividing the input data into smaller subsets, each corresponding to a specific class. The process of dividing the input space can be formalised as a recursive algorithm that starts with the entire input space $\mathcal{X}$ and repeatedly splits it into smaller subsets. The splits in the input space are based on the values of the predictors, resulting in the dataset becoming increasingly homogeneous with each split. The final result of this process is a tree-like structure where each node represents a subset of the input data. The end nodes of the tree, known as leaf nodes, are assigned a class based on the distributions of the classes of the training cases. This tree structure can be used to make predictions for new data points by traversing the tree and arriving at a final prediction based on the class assigned to the leaf node that the data point belongs to. The decision tree algorithm provides a visual and hence intuitive representation of the relationships between the predictors and the classes, making it a useful tool for understanding and interpreting the data. Advances in decision tree theory lead to various different algorithms for constructing decision trees. In this paper we will primarily use the CART algorithm (Breiman, 1984) to construct trees.

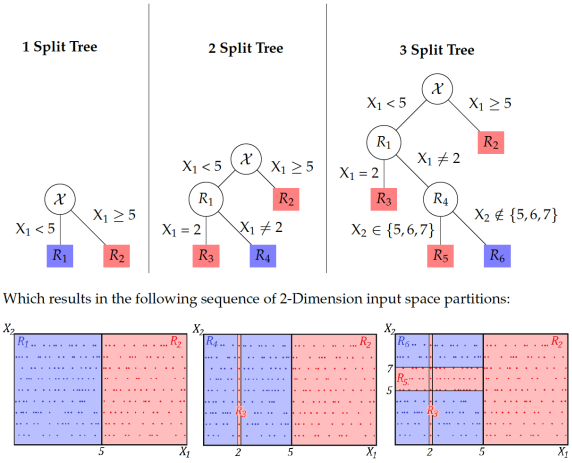

Figure 1: Illustration of a decision tree model's sequential division of the input space.

The algorithm is primarily divided into three parts: selection of splits, pruning the tree and assigning the leaf node labels. Figure 1 demonstrates the sequential division of the input space $\mathcal{X}$, using continuous predictor X_1 and discrete predictor X_2. Each tree in the diagram represents one partition of the input space, illustrating how the model splits the data based on these predictors. This depiction of decision trees takes a top-down approach, starting from the root node (input space) $\mathcal{X}$, which is split into two nodes R_1 and R_2 based on ranges of X_1. Discrete variable X_2 then splits the tree twice, first splitting R_1 into R_3 and R_4 and then splitting R_4 into R_5 and R_6. Each node represents a disjoint subset of $\mathcal{X}$.

2.3 Tree-Based Learners

In this section we explore tree-based learners, which comprise techniques that utilise multiple decision trees to tackle a learning task. These form part of the ensemble learning umbrella of techniques, where the aim of ensemble learning is to construct a prediction model by combining the strengths of a set of simpler base models. This process can be divided into two steps: creating a set of base learners and combining them to form a composite predictor. Tree-based learners are often able to achieve stronger generalisation abilities than individual decision trees due to the combination of multiple models. Tree-based learners can be largely classified into two categories, depending on the approach used to generate the individual learners:

1. **Bagging and Random Forests:** These methods generate individual learners independently. Bagging (Bootstrap Aggregating) is a technique where the same model is trained on different bootstrapped samples of the data, and then their outputs are averaged to obtain a final prediction. Random forests is a variant of bagging that constructs a collection of decision trees using random feature subsets.
2. **Boosting:** Boosting methods generate learners with strong correlations and create them in a sequence. The idea behind boosting is to sequentially train models on weighted versions of the data, where the weights are adjusted to emphasize the instances that were misclassified by the previous models. In this study, we consider GBM, XGBoost and LightGBM.

The main difference between these categories is how the base learners are generated. Bagging and random forests generate learners independently, while boosting methods generate learners in sequence with strong correlations. Despite this, both bagging and boosting methods use similar techniques to combine the multiple base learners. In Section 2.3.1 and Section 2.3.2 we describe these two main approaches in more detail.

2.3.1 Bagging and Random Forests

One approach to generating different base learners is to divide the original training data into a number of distinct subsets and use each subset to train a different base model. To improve the quality of each base model, it is often necessary to allow some overlap between the subsets, such that each one contains an adequate number of training samples. Several randomization approaches have been proposed to build independent base learners. Bootstrap aggregating, commonly known as *bagging*, is one example of a resampling method for classifier design as it utilizes the bootstrap sampling method.

Given the training set $D = \{\mathbf{x}^{(j)}, y^{(j)}\}_{j=1}^{N}$, a bootstrap sample would be a subset $D_m \subseteq D$ of the full learning set each created by randomly drawing $N' \leq N$ instances from D with replacement. In bagging, the final prediction is obtained by combining the outputs of all the base learners in the committee through an aggregation method, such as taking their average in case of regression or taking the mode in case of classification. Since each sample is drawn from the same distribution, the base learners are considered to be identically distributed. As a result, the expected value of the average of multiple base learners is equivalent to the expected value of a single learner. Therefore, the bias of bagged base learners is identical to that of individual learners. This means that bagging can only improve performance by reducing the variance.

Bagging is known to be highly effective for low-bias, high-variance methods like trees (Hastie et al., 2009). Random forests, are a significant modification of bagging proposed by (Breiman, 2001). They enhance the latter technique by building a large ensemble of uncorrelated trees and find the mode to obtain the final classification. While in traditional decision trees, the feature that is used to split a node is chosen from the entire set of features, in a random forest tree, a subset of K features is randomly selected from the feature set at each node. This introduces a degree of randomness, controlled by the parameter K, with $K = p$ resulting in the selection of features being the same as traditional trees, and $K = 1$ resulting in a completely random selection. In order to achieve optimal outcomes, it is a frequently adopted practice to use $K = \log_2 p$ (Breiman, 1984) or $K = \sqrt{p}$ (Hastie et al., 2009). However, it should be noted that this approach may not be universally applicable, and alternative values may be necessary in certain situations.

2.3.2 Boosting

Boosting is an ensemble technique that involves combining the outputs of many weak classifiers in series to create a strong committee. The key idea behind boosting is to adjust the distribution of training samples based on the errors made by weak base learners. This adjustment of the distribution is what makes boosting fundamentally different from other ensemble methods. Boosting grows the base learners in an adaptive way that removes bias by adjusting the distribution of training samples. This approach means that the base learners in boosting are not identically distributed (Hastie et al., 2009). By iteratively adapting

the distribution of training samples, boosting aims to reduce both the bias and variance of the base learners. This particular feature of boosting makes the method especially useful for real-life application scenarios, such as in predicting, detecting, and ultimately preventing motor insurance fraud.

Boosting algorithms start by training a base learner and adjusting the distribution (through reweighting) of training samples based on the base learner's output. Instances that were incorrectly classified by the base learner are given more importance by subsequent base learners. The following base learner utilizes the modified training data, and this cycle continues until a predetermined number, represented by M, of base learners are created. Finally, this results in a sequence $\hat{f}_m(\mathbf{x}), m = 1,2,\ldots,M$ of base learners which in turn are combined to form the final classifier $\hat{f}(\mathbf{x})$. In this research a variety of boosting algorithms were used. These include GBMs (Hastie et al., 2009), XG Boosting (XGBoost) (Chen and Guestrin, 2016) and Light Gradient Boosting Machines (LightGBM) (Ke et al., 2017). GBMs are learners that utilise gradient descent to minimise the loss function. Moreover, XGBoost and LightGBM are both powerful Gradient Boosting frameworks that leverage the second-order Taylor expansion for approximating the objective function, optimizing the performance of decision tree ensembles. Despite their similarities, these frameworks employ different strategies for growing trees. XGBoost grows trees level-wise, ensuring balanced tree structures, while LightGBM grows trees leaf-wise, prioritizing the most significant splits to achieve faster convergence and higher accuracy. Furthermore, LightGBM implements two algorithms to accelerate the training process: Gradient-Based One Sided Sampling (GOSS) algorithm and Exclusive Feature Bundling (EFB). The former selectively samples instances from the dataset based on the absolute values of their gradients, ensuring that instances with larger gradients, which contribute more to the information gain, are included, while the latter enables LightGBM to process large datasets more efficiently. Regularization techniques are crucial in preventing overfitting in Gradient Boosting models. The Gradient Boosting techniques employ the $L1$ and $L2$ regularization terms in their loss functions. These are terms that can be added to the loss function during training to prevent overfitting.

3 RESULTS

The description of the dataset under study has been provided in Section 1. The insurance company that provided the data did so under condition of anonymity, hence no information regarding the country of origin of the claims as well as the name of the company will be provided. The justification for the 90-10 split ratio has been given in Section 2. Any model validation that occurs during the preprocessing and building up to the optimal model is performed solely on the training set. The test set is reserved for a single use at the end with the aim of determining the optimal model, to avoid introducing any bias in the evaluation process.

Primarily, feature extraction was implemented. This included extracting information from date variables, creating indicator variables for specific conditions, and addressing high cardinality issues in nominal variables. Furthermore, the dataset contained missing data. A median imputation approach was implemented, with the mean imputation and k-Nearest Neighbors (KNN) imputation also attempted but not affecting the results obtained. Feature selection was required to improve the computational efficiency in the training of the tree-based learners. To conduct feature selection, a LightGBM model was used on the unbalanced dataset to obtain feature importance scores due to its speed and efficiency compared to other approaches. A plot of the mean ROC AUC vs the feature importance threshold was then obtained to determine the ideal threshold. The Receiver Operating Characteristic curve, abbreviated as ROC curve, is a tool for plotting the true positive rate $\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$ against the false positive rate $\text{FPR} = \frac{\text{FP}}{\text{FP+TN}}$, and AUC stands for area under the curve which quantifies model ability to discriminate between the two categories.

To select the optimal subset of features, an iterative process was then employed. The process began with a feature importance threshold of 0, progressively increased by increments of 1, and at each step, the model re-evaluated with the variables that met or exceeded this threshold using 5-fold cross-validation. Figure 2 was then used to determine a cut-off point using the elbow method, with a threshold of 22.0 being identified. This resulted in a selection of 28 variables, as this is the threshold beyond which the mean ROC AUC resulted in a sudden shift downwards. The most important variables (together with their level of importance) are shown in Figure 3 It can be seen that the province in which the claim was made and the province of the policy holder were the most important features by a huge margin, but an interpretation of feature importance values will be given later due to the fact that SMOTE-NC or cost-sensitive learning have not yet been implemented and feature importance scores can change once this is the case.

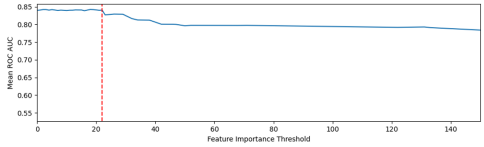

Figure 2: Mean ROC vs Feature Importance Threshold.

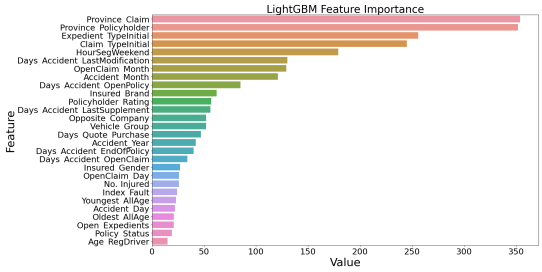

Figure 3: LightGBM feature importance scores of the selected 28 variables.

Following feature extraction, missing data imputation and feature selection, the final dataset is created and randomly divided into a training set and a test set. Decision trees, random forests, GBM, LightGBM and CatBoost are implemented using both a SMOTE-NC approach and cost-sensitive learning on the training set. During the training phase, model parameter tuning also occurs via 5-fold cross-validation on the training set. Finally, the fitted models are implemented on the test set for comparative purposes. A flowchart illustrating the entire model implementation and evaluation process is presented in Figure 4. The fine tuned parameters for decision trees were the maximum depth of the tree, the minimum number of rows to create a leaf node, and the minimum relative improvement in impurity reduction for a split to happen. These were optimised on ranges $\{3,6,9\}$, $\{3,6,9\}$ and $\{0,0.2,0.4\}$ respectively. The fine tuned parameters for random forests were the proportion of rows to be randomly sampled for each tree, the proportion of columns to randomly select at each tree node split, and the number of trees in the random forest model. These were optimised on ranges $\{0.5,0.7,0.9\}$, $\{0.5,0.7,0.9\}$ and $\{50,100,200,500,1000\}$ respectively. The fine tuned parameters for the boosting methods were maximum depth of the tree, the minimum sum of instances weights are needed in a child node, the fraction of data used for training each iteration and the fraction of features used to build each tree during training. These were optimised on ranges $\{3,5,7,9\}$, $\{1,3,5,7\}$, $\{0.6,0.7,0.8,0.9\}$ and $\{0.6,0.7,0.8,0.9\}$ respectively. Finally the $L1$ and $L2$ regularization terms were found to progressively decrease the performances when increased and were set to the default of 0.

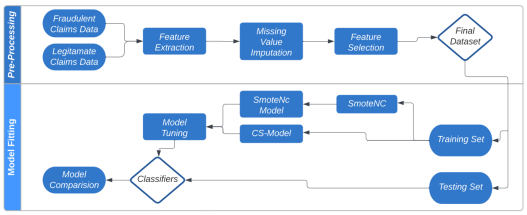

Figure 4: Flowchart illustrating the model implementation and evaluation process.

In Table 1 and 2, the results obtained when fitting decision trees, random forests, GBM, XGBoost and LightGBM are shown for both the SMOTE-NC approach and the cost-sensitive learning approach. For the SMOTE-NC approach, in Table 1 we denote these models by SNC-DT, SNC-RF, SNC-GBM, SNC-XGB and SNC-LGBM. For the cost-sensitive learning approach, in Table 2 we denote these models by CS-DT, CS-RF, CS-GBM, CS-XGB and CS-LGBM respectively. The metrics we consider for model evaluation are the following:

1. Recall $= \frac{TP}{TP+FN}$
2. NPV $= \frac{TN}{TN+FN}$ (negative predicted value)
3. TNR $= \frac{TN}{TN+FP}$ (true negative rate)
4. ROC AUC (defined earlier)
5. Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$
6. Balanced Accuracy $= \frac{TPR+TNR}{2}$

In Table 1, when implementing synthetic minority oversampling, it can be seen that random forests and XGBoost have the best recall, with the other methods performing poorly. Nonetheless, XGBoost also performs worst in terms of accuracy and second worst in terms of balanced accuracy. Only random forests appear to provide consistently good metrics throughout. In Table 2, on the other hand, when applying cost-sensitive learning, the recall for decision trees deteriorates while the recall for all the other methods improve throughout, as does the balanced accuracy (albeit only marginally for random forests). Indeed one can see that recall is best for random forests with LightGBM second best, while TNR, ROC AUC and balanced accuracy are the best for LightGBM, where random forests also yield the worst performance for TNR. Random forests and LightGBM tie when it comes to NPV while GBM is the best when it comes to accuracy (with LightGBM also second best here). Nonetheless, recall, TNR, accuracy and balanced accuracy are all above 0.7 for random forests and the boosting algorithms, while the the ROC AUC is above 0.8 throughout. The NPV is also consistently close to 1 or 1. Since LightGBM under cost-sensitive

Table 1: Comparison of different tree-based learners using SMOTE-NC.

	SNC-DT	SNC-RF	SNC-GBM	SNC-XGB	SNC-LGBM
Recall	0.66	0.82	0.53	0.91	0.55
NPV	0.99	1.00	0.99	0.99	0.99
TNR	0.83	0.73	0.86	0.27	0.87
ROC AUC	0.77	0.84	0.81	0.69	0.83
Accuracy	0.83	0.73	0.85	0.28	0.86
Balanced Accuracy	0.75	0.78	0.70	0.71	0.81

Table 2: Comparison of different tree-based learners using cost-sensitive learning.

	CS-DT	CS-RF	CS-GBM	CS-XGB	CS-LGBM
Recall	0.23	0.86	0.79	0.81	0.83
NPV	0.98	1.00	0.99	0.99	1.00
TNR	0.96	0.72	0.80	0.74	0.79
ROC AUC	0.84	0.85	0.86	0.83	0.87
Accuracy	0.95	0.72	0.80	0.74	0.79
Balanced Accuracy	0.6	0.79	0.79	0.77	0.81

learning is the best or second best throughout all considered metrics, this is considered to be the best model overall when it comes to successfully detecting fraudulent cases without compromising heavily TNR. Furthermore, cost-sensitive learning has shown to be a considerable improvement over SMOTE-NC for all approaches except decision trees in increasing recall and balanced accuracy.

To further illustrate the performance of the cost-sensitive LightGBM model, an ROC curve and a Precision-Recall (PR) curve are presented in Figure 5. The ROC curve demonstrates the model's performance on both the training and test sets, showcasing how closely they align. This indicates a good balance between complexity and performance, and also signifies the model's generalizability to new data. Furthermore, the PR curve indicates that to obtain a good recall, a poor precision is unavoidable. Note that $Precision = \frac{TP}{TP+FP}$ which indicates that a large percentage of false positive non-fraudulent claims is required to have a good recall. Indeed, for the LightGBM under cost-sensitive learning which has achieved a recall of 0.83, the precision lies just at 0.08. This, however, is the drawback that comes with detecting more fraudulent cases. Overall, metrics between training and tests sets were comparable, indicating that overfitting was not an issue.

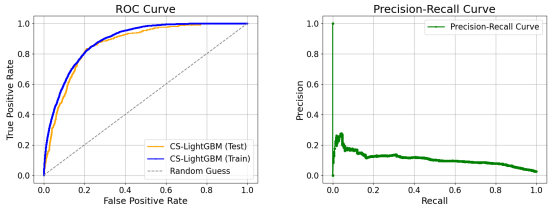

Figure 5: ROC and PR curves for the final cost-sensitive LightGBM model.

In Figure 6, the feature scores by order of importance for LightGBM using cost-sensitive learning are given. It can be seen that claim type has been by

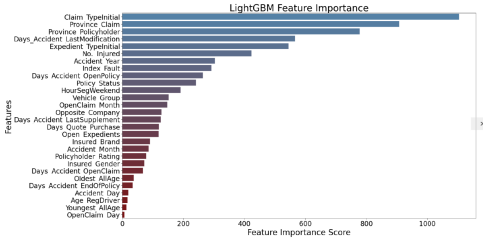

Figure 6: Feature importance scores of the selected variables for LightGBM with cost-sensitive learning.

far the most important feature in this case, superceding the province in which the claim was made and the province of the policy holder, which were originally the most important in the absence of any synthetic minority oversampling or cost-sensitive learning. Through further investigation, it was found that certain claim types such as injury claims were more prone to fraud than others. Province-related variables now place second and third in terms of importance. The number of days between accident occurrence and last insurance policy modification was the fourth most important feature, with shorter periods being more likely to be associated with fraud. The variable related to claim processing (Expedient TypeInitial) was the fifth most important variable, with the MREC category (standing for Maximum Reasonable Estimate of Claim - related to providing a reasonable estimate of damages rather than detailed assessment) being the most likely associated with fraud. The sixth most important variable was the number of injured individuals, where there is evidence that claims with a higher number of injured individuals were more likely to be fraudulent.

4 DISCUSSION

In this study, it has been shown that the more complex tree-based learners outperformed decision trees when implementing cost-sensitive learning. These included random forests, GBM, XGBoost and LightGBM. The comparison with decision trees turned out to be more of a mixed bag when applying SMOTE-NC. Nonetheless, the results obtained for the more complex tree-based learners were best when implementing cost-sensitive learning, with a stark improvement in the poorly performing metrics related to the synthetic minority oversampling is applied. Indeed, only random forests had an overall good performance under SMOTE-NC, which still showed a slight improvement under cost-sensitive learning, and which still did not exceed the capabilities of LightGBM under cost-sensitive learning. When compared to random

forests, LightGBM yielded more balanced results and although random forests produced a marginally higher recall, this came at the expense of a significantly increased number of FPs - a trade-off which is deemed unfavourable, as high false positives can lead to increased operational costs and potential customer dissatisfaction. Hence, the LightGBM model with cost-sensitive learning emerges as the preferred choice due to its enhanced fraud capturing capabilities and balanced performance. Furthermore, cost-sensitive learning emerged as the superior way to address data imbalancing on this dataset. Also, when comparing feature importance under an imbalanced dataset with feature importance when data imbalancing is addressed, one can see that variables related to the claim type, claim processing and the number of injuries became more promintent in the latter. Indeed, injury-related claims have been found to be more prone to fraud.

It is important to recognise the limitations of the dataset, particularly with regard to the accuracy and completeness of the fraud labels. As the data is sourced from a single motor insurance company, it is subject to the specific methods and procedures used by that company to identify and report fraudulent claims. Consequently, the dataset may not be fully representative of the true incidence of fraudulent claims within the wider motor insurance industry. This could potentially impact the reliability and generalisability of any results obtained from the data, particularly if the sample is biased in any way towards certain types of claims or customers. Nonetheless, further research can be done to determine which classification techniques are useful for correctly identifying fraud in a cost-effective manner, and whether cost-sensitive learning is truly a more superior approach to addressing data imbalance when compared to synthetic minority oversampling. Furthermore, this research can be further enhanced by possibly incorporating principal component analysis to reduce the dimensionality of the data set. This would allow us to use information from all the features and improve predictability, however this could come at the expense of interpretability of the features.

REFERENCES

Al-Hashedi, K. G. and Magalingam, P. (2021). Financial fraud detection applying data mining techniques: A comprehensive review from 418 2009 to 2019. *Comput. Sci. Rev.*, 40, . 419:100–402.

Aslam, F., Hunjra, A. I., Ftiti, Z., Louhichi, W., and Shams, T. (2022). Insurance fraud detection: Evidence from artificial intelligence and 416 machine learning. *Res. Int. Bus. Financ.*, 62, . 417:101–744.

Bhattacharyya, S., Jha, S., Tharakunnel, K., and Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decis. Support*, 50(3):602–613. 424 Syst.,, . 425.

Breiman, L. (1984). Classification and regression trees. *Chapman and Hall/CRC: New York*, 431:18–58.

Breiman, L. (2001). Classification and regression trees. *Mach. Learn*, 45, . 434:5–32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority oversampling technique. *J. Artif. Intell.*, 429 Res., 16, . 430:321–357.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International 436 Conference on Knowledge Discovery and Data Mining, San Francisco, USA, 13-27 August 2016*, pages 785–794.

Gomes, C., Jin, Z., and Yang, H. (2021). Insurance fraud detection with unsupervised deep learning. *J. Risk Insur.*, 88:591–624.

Hargreaves, C. A. and Singhania, V. (2015). Analytics for insurance fraud detection: An empirical study. *Am. J. Mob. Syst. Appl. Serv.*, 1(410):3. 227–232. 411.

Hashmi, N., Shankaranarayanan, G., and Malone, T. W. (2018). Is bigger better? a study of the effect of group size on collective intelligence 407 in online groups. *Decis. Support Syst.*, 107:88–98.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). The elements of statistical learning: Data mining, inference, and prediction. In *432*, pages 261–288. Springer-Verlag, New York, 2 edition.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Weidong, M., Ye, Q., and Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting 438 decision tree. In *Proceedings of the 31st Conference of Neural Information Processing Systems, Long Beach, USA, 4-9 December 439 2017*, pages 3149–3157.

Kemp, G. (2010). Fighting public sector fraud in the 21st century. *Comput. Fraud Secur.*, 11, . 414:16–18.

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., and Sun, X. (2011). The application of data mining techniques in financial fraud detection: A 426 classification framework and an academic review of literature. *Decis. Support Syst.*, 50(3):559–569.

Nian, K., Zhang, H., Tayal, A., Coleman, T., and Li, Y. (2016). Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *412 J. Financ. Data Sci.*, 2, . 413:58–75.

Phua, C., Alahakoon, D., and Lee, V. (2004). Minority report in fraud detection: Classification of skewed data. *ACM SIGKDD Explor.*, 6(420):1. pp. 50–59. 421.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*. New York, pp, Springer-Verlag, 2 edition. 1–16.