POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Comparative Analysis of Natural Language Processing and Gradient Boosting Trees Approaches for Fraud Detection

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Ludovica Lerma

Advisor: Prof. Marcello Restelli

Co-advisor: Fabrice Popineau

Academic year: 2022-2023

1. Introduction

With the increasing popularity of digital payments, fraud detection systems have become indispensable in limiting monetary losses for both customers and card-provider companies. Recognizing the significance of this issue, online payment platforms actively incorporate robust fraud detection systems into their infrastructure.

When it comes to addressing fraud detection challenges, Decision Trees have emerged as a clear and efficient approach, extensively integrated into current fraud detection systems [1]. Gradient Boosting Trees (GBT), which are a type of Decision Trees, are the focus of this study. However, this relatively simple model faces considerable difficulties effectively handling the complexities associated with fraud detection, particularly concerning data imbalance and concept drift.

Therefore, this thesis presents an original contribution by utilizing and analyzing Natural Language Processing (NLP) methodologies in the context of fraud detection, while directly comparing their performance against the Gradient Boosting Trees approach. By doing so, we aim to highlight the strengths and limitations of both approaches and uncover the potential benefits of applying NLP techniques in this specific domain.

2. Problem Formulation

Consider a dataset $D_n = (x_i, y_j)_{i=1}^n$ where $x_i \in X \subseteq R^d$ and $y_i \in Y = 0, 1$.
Each payment i is described by its features x_i *(e.g. amount, date,... etc)* and its label y_i flagging whether it is a fraud, $y_i = 1$, or not, $y_i = 0$.
Fraud detection consists of estimating a function $f : X \mapsto Y$ using D_n, i.e. a function which predicts whether a payment i is fraudulent based on its features x_i.

Fraud Detection is considered a challenging task for three main reasons [5]:

- Imbalanced learning: non-fraudulent payments are significantly more numerous than fraudulent ones.
- Concept drift: fraudsters change their behaviors due to a cat-and-mouse game.

$$D_n^{train} \sim p_{train}(x, y)$$
$$D_n^{test} \sim p_{test}(x, y)$$
$$p_{test}(x, y) \neq p_{train}(x, y)$$

- Explainability: for regulatory reasons, it is necessary to account for a model's prediction.

3. Related Work

Gradient Boosting Trees (GBT) have been widely employed in various domains for building predictive models and classifiers. In this approach, weak learners, which are represented by decision trees, are enhanced by combining them sequentially to construct more resilient models. The objective of each subsequent tree is to minimize the errors made by the previous ones, resulting in an iterative improvement of the model's performance. The effectiveness of GBT in fraud detection has been explored in several studies [7], [6].

To further enhance fraud detection, we draw inspiration from the FraudMemory study conducted by Yang et al. [8] The authors propose a hybrid fraud detection system that combines *sequential* and *memory-enhanced methods*. The core idea behind FraudMemory is to model the normal behavior of each user and identify transactions that significantly deviate from this behavior as potential frauds. To capture the valuable information from transactions, the authors employ two distinct representations: the user profile representation and the log representation. The user profile representation encapsulates user behavior by considering metrics such as transaction intervals, frequency, and monetary value. On the other hand, the log representation focuses on extracting attribute-level information using a modified version of the Continuous Bag-of-Words model, called the Continuous Bag-of-Attributes model (CBOA). To capture the sequential patterns inherent in transactional data, the FraudMemory model utilizes a Gated Recurrent Unit (GRU) model. The GRU model is designed to process sequential data and extract dependencies among transactions. Finally, a Multilayer Perceptron (MLP) is employed to evaluate the sequential representation and assign a fraud score to each transaction.

While GBT has demonstrated effectiveness in fraud detection, the NLP-inspired approach presented in FraudMemory offers a novel perspective by combining sequential analysis and memory-enhanced techniques. In our study, we aim to build upon these ideas and evaluate the performance of GBT as well as compare it with the NLP-based model inspired by FraudMemory. The evaluation encompasses both the conventional GBT approach with augmented feature engineering techniques and an exploration of its performance when augmented with NLP embeddings. Our focus lies in isolating the latent representation of transactions and utilizing an LSTM model to extract sequential patterns, with the ultimate goal of enhancing the accuracy and efficacy of fraud detection systems.

4. Background

4.1. Gradient Boosting Trees

GBT are an extension of the AdaBoost algorithm [2]. AdaBoost initially trains decision trees on equally weighted observations and iteratively adjusts the weights based on misclassified samples, leading to the construction of an ensemble of trees. The final predictions are obtained through a majority voting scheme weighted by the individual accuracies of the weak learners. GBT expands upon this approach by formulating the problem as a numerical optimization task, using a gradient descent-like procedure to sequentially add weak learners and minimize the model's objective.

4.2. Natural Language Processing

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on enabling machines to understand, interpret, and generate human language. One crucial aspect of NLP is the representation of words in a format that machines can effectively process.

Word embeddings are techniques used to transform words or phrases from their original high-dimensional input space into a lower-dimensional numerical vector space. These continuous vectors represent words and capture their semantic meaning. Notably, word embeddings can create clusters in the projection space, where similar words are located close [4].

Another important aspect of NLP is the management of sequential data. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that addresses the challenges associated with sequential data. LSTMs are structured with recurrent connections and gated memory cells in each neuron. The recurrent connectivity enables the network to maintain memory of past inputs and learn long-term dependencies. The gated memory within each cell controls the flow of information, allowing the retention or exclusion of data [3].

5. Proposed Models

The GBT model can be split into two stages:

1. In a first stage, we use features engineering to encapsulate time in new attributes. These attributes are derived with the aid of experts' knowledge and comprise of, to name a few: mean and standard deviation (std) of the amount spent, cumulative standard deviation (nbstd) of the amount spent, rolling sum of the amount spent, average number of hours -in the range 1 to 24 over a day period- and average minutes between two consecutive transactions, ...
2. In the second stage, an implementation of Gradient Boosting Tree, Light Gradient Boosted Machine (LightGBM), is utilized to classify transactions as fraudulent or legitimate.

The NLP model can also be divided into two distinct phases:

1. In the first phase, which is a preliminary phase, the attributes undergo a processing step through an embedding layer. The embedding layer is constructed using the Continuous Bag of Attributes (CBOA) dense embedding technique. This technique enables the projection of the attributes into a lower-dimensional space. By doing so, the attributes are quantified and represented as real-valued vectors that capture their semantic relationships with other attributes. During this phase, the incoming transaction is divided into a_{ij}, where $i \in A$ and $j \in A_i$ represent the attribute and its corresponding value. It is important to note that the sequence order of a_{ij} is irrelevant, and each a_{ij} relies on the other a_{mn} (where m belongs to A and m is not equal to i) as its context. The objective of CBOA is to use contextual attributes to predict the target attribute, with the ultimate aim of maximizing the logarithmic probability [8].

$$\log p(a_{ij} | \{a_{mn} | m \in A, m \neq i, n \in A_m\})$$

2. The second phase frames the model architecture.
The model compounds a LSTM layer and two dense layers on top. The LSTM is fed with sequences of vectorized attributes and extracts temporal patterns. It uses the sequence of transactions $r_1^u, r_2^u, \ldots, r_t^u$ of user u to calculate the current hidden state vector h_t^u at time-step t based on the previous hidden state vector h_{t-1}^u:

$$h_t^u = LSTM(h_{t-1}^u, r_t^u, \theta)$$

where θ denotes the LSTM parameters to learn. Finally, two dense layers are employed to evaluate the sequential representation and determine whether a transaction is fraudulent or not.

$$Prediction_{sequence} = Dense(h_t^u)$$

Finally, a third model combines and integrates the two previous models by incorporating the CBOA representation of transactions and feeding them into the LightGBM.

6. Experiments

6.1. Dataset

A large dataset of 100 million labeled transactions from a real payment service is used. Each transaction consists of several attributes, including both categorical and continuous variables. Of particular interest are the "card" attribute, which identifies the card used for the transaction and links it to the series of transactions associated with the same card, and the "fraud" flag attribute, which indicates whether a transaction is classified as fraudulent. Less than 1% of the transactions in our dataset are flagged as fraud. This makes the dataset highly imbalanced, presenting a challenge in developing an effective classification model for time-series input data. The original dataset has been modified in several ways: from dropping attributes useless to

	Loss	TP	FP	TN	FN	ACC	PREC	RECALL	AUC	Support
LSTM	0,03138	5756	17795	7264219	34342	0,9928	0,2445	0,1435	0,8112	7322112
LGBM		5843	7841	7274411	34263	0,9942	0,4270	0,1457		7322358
LGBM-2		3297	4166	7278086	36809	0,9944	0,4418	0.0822		7322358

Table 1: Comparison LSTM, LGBM and LGBM with CBOA on short history

	Loss	TP	FP	TN	FN	ACC	PREC	RECALL	AUC	Support
LSTM	0,1414	9790	12900	3966309	20473	0,9917	0,4315	0,3235	0,9077	4009472
LGBM		5843	6112	3973264	24424	0,4887	0,1930	0,1457		4009643
LGBM-2		2854	2984	3976392	27413	0,9924	0,4889	0,0943		4009643

Table 2: Comparison LSTM, LGBM and LGBM with CBOA on long history

the learning process to change the data format to boost performances. Finally, cards and merchants not associated to fraudulent transactions are dropped to slightly balance the dataset and focus the learning process on fraudulent sequences.

6.2. Metrics

Accuracy is a commonly used metric, but it's not a reliable metric for evaluating imbalanced datasets as it can be misleading. For example, if we had a model that predicts "non-fraud" for all instances in a dataset where fraud represents only 0.4% of the total instances, it would achieve a 99.6% accuracy. To gain better insight, Precision, Recall and AUC (Area Under the Curve) are used. In the fraud detection task, classifying non-fraudulent transactions as fraudulent means for the client to have their money frozen and the service suspended for undefined time. Our major interest relies on recall.

6.3. Experiments

This study presents two main objectives. Firstly, it aims to compare different window sizes and determine the optimal one for the LSTM model's performance. Here, the term "window size" refers to the number of transactions used to train the LSTM model. Secondly, it focuses on comparing the LSTM model with the LightGBM model.

To achieve these objectives, we conducted experiments in three categories:

- The first category, which results are shown in table 1, involved cards with short histories, specifically those with 8 or more transactions. This minimum value was chosen based on an average of 7.76 transactions, ensuring a substantial portion of the dataset was included. To encompass a wide range of cards, a window size of 3 was used in this experiment. The goal is to assess the model's ability to learn despite the limited number of transactions.
- The second category, which results are shown in table 2, focused on cards with longer histories, specifically those with 18 or more transactions. Although the majority of cards had fewer than 25 transactions, longer card histories provide the LSTM model with more data to learn patterns effectively. To strike a balance between adequate data for the LSTM model and a reasonable number of cards for analysis, a window size of 12 was selected.
- The third category, which results are shown in table 3, involved cards with even longer histories, specifically those with 40 or more transactions. The objective here is to evaluate the model's performance with extended card histories. Consequently, a window size of 32 was chosen to accommodate these lengthy sequences and assess the model's handling of them.

window size	Loss	TP	FP	TN	FN	ACC	PREC	RECALL	AUC	Support
3	0,03138	5756	17795	7264219	34342	0,9928	0,2445	0,1435	0,8112	7322112
12	0,1414	9790	12900	3966309	20473	0,9917	0,4315	0,3235	0,9077	4009472
32	0.0330	8994	12590	3966619	21269	0.9916	0.4167	0.2972	0.8777	161200

Table 3: Comparison window sizes for LSTM performance

6.4. Model Architecture

In our experiments, we varied the architecture of the LSTM model and discovered the following empirical findings for optimal performance:

- The number of units in the LSTM should match the maximum window size length.
- The length of the hidden state should correspond to the length of the sequences being handled to ensure proper memorization and utilization.
- To avoid excessively long training times, we limited the maximum window size to a fixed value of 40, as some cards had thousands of transactions.

Moreover, the presence of outliers can significantly impact the training process, leading to vanishing or exploding gradients. To mitigate this issue, we took the following steps:

- We set the parameter "global clipnorm" of the optimizer to 0.5. This parameter sets an upper limit to the L2 norm of gradients, ensuring smoother gradient values.
- We assigned a higher weight to instances of the positive class when computing the loss function to counterbalance the dataset imbalance.

6.5. Results

Tables 1 and 2 provide a direct comparison between the LightGBM and the NLP-based model, either on short histories and on long histories.

In the first case, we cannot see significant improvements in the use of the NLP-based model with respect to the use of the LightGBM.

On long histories, however, the NLP-based model proves to perform remarkably better than its counterpart. Indeed, we observe an improvement in Recall, that scores around 0.3235.
Based on this results, we can conclude that the LSTM-based model is superior to the LightGBM, solving the task of fraud detection on a dataset of cards with long history.

Table 3 provides an immediate comparison between the performances achieved by the NLP-model set with different values of window sizes. The three models are trained on different datasets:

1. the first one, with a window size set to 3, is trained on a database made of cards with an history made of 8 or more transactions.
2. The second one, with a window size set to 12, is trained on a database made of cards with an history made of 18 or more transactions.
3. The last one, with a window size set to 32, is trained on a database made of cards with an history made of 40 or more transactions.

The three model are trained during 30 epochs. The best performance is achieved by the second model. Although, it must be considered that most cards are related to short histories. Indeed, when selecting cards with longer and longer histories, we end up with much fewer populated databases.

7. Conclusion

In conclusion, the experimental results provide two significant findings. Firstly, it was determined that the optimal window size for the LSTM model is 12. Additionally, when compared to the LightGBM models, the LSTM model showcased superiority on a dataset consisting of cards with a long history and a window size of 12, resulting in an improvement of 0.1778 points in Recall with respect to the best result achieved by the LightGBM.

While the experimental results are presented, it is worth noting that more advanced techniques, such as transformers and attention mechanisms, hold the potential to surpass the capabilities of the current methodologies. These advanced techniques offer the complexity required to capture intricate patterns and dependencies within transactional data, which could significantly enhance the accuracy and effectiveness of fraud detection systems.

8. Bibliography

References

[1] Khaled Gubran Al-Hashedi and Pritheega Magalingam. Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Computer Science Review*, 40:100402, 2021.

[2] Leo Breiman. Prediction games and arcing algorithms. *Neural computation*, 11(7):1493–1517, 1999.

[3] Gang Chen. A gentle tutorial of recurrent neural network with error backpropagation. 10 2016.

[4] Tomas Mikolov, Quoc Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. 09 2013.

[5] Aisha Abdallah n, Mohd Aizaini Maarof, and Anazida Zainal. Fraud detection system: A survey.

[6] Altyeb Altaher Taha and Sharaf Jameel Malebary. An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. *IEEE Access*, 8:25579–25587, 2020.

[7] David G Whiting, James V Hansen, James B McDonald, Conan Albrecht, and W Steve Albrecht. Machine learning methods for detecting patterns of management fraud. *Computational Intelligence*, 28(4):505–527, 2012.

[8] Kunlin Yang and Wei Xu. Fraudmemory: Explainable memory-enhanced sequential neural networks for financial fraud detection. In *Hawaii International Conference on System Sciences*, 2019.