Hyphatia: a Card-Not-Present Fraud Detection System based on Self-Supervised Tabular Learning

Josue Genaro Almaraz-Rivera[1], **Jose Antonio Cantoral-Ceballos**[1], **Juan Felipe Botero**[2], **Francisco Javier Muñoz**[3], **Brian David Martinez**[3]
[1]Tecnologico de Monterrey [2]Universidad de Antioquia [3]Aligo Defensores Informaticos S.A.S.
`{a00821189, joseantonio.cantoral}@tec.mx, juanf.botero@udea.edu.co, {francisco.munoz, brian.martinez}@aligo.com.co`

Abstract

Card-Not-Present fraud uses the payment card information of a victim to buy in e-commerce platforms and later shows in the form of chargebacks. In 2024, it is expected to reach losses in the United States of 10 billion dollars. In the state of the art, the IEEE-CIS dataset has emerged as a strong option for creating smart detection systems against this problem. In this work, we create a solution that we call Hyphatia, where the novel Self-Supervised Learning paradigm is implemented in the tabular data domain using SubTab, outperforming XGBoost by 2.14% AUROC, detecting 67.44% of the fraud cases in the IEEE-CIS. This pioneering experimentation prioritizes those features that are not obfuscated, and beyond providing just classification metrics, we also provide time performance and feature importance calculations for explainability. To the best of our knowledge, this is one of the first works in the literature using the Self-Supervised Tabular Learning approach for the problem of credit card fraud detection.

1 Introduction

In this research, the IEEE-CIS dataset [1] is selected due to its characteristics of real-world e-commerce transactions and the availability of context for some of the provided features. Moreover, the novel Self-Supervised Learning (S-SL) paradigm [2] is selected for this study, particularly implementing the SubTab architecture [3] for the tabular data[1] domain.

In summary, the primary contributions of this work are as follows:

- Self-Supervised Tabular Learning models beating the classification performance provided by state-of-the-art baselines including XGBoost [4], for the binary credit card fraud detection task, using SubTab and the IEEE-CIS dataset.
- Non-linear evaluation head using Multi-layer Perceptron (MLP) is added to the original SubTab architecture to capture more complicated boundaries in the customers' data behavior.
- Pioneering experimentation in the S-SL and tabular learning domains, closing the existing performance gap between Machine Learning (ML) and Deep Learning (DL) models, without requiring the labeling of large amounts of input data.

The rest of this document is organized as follows: section 2 shows the related literature about the use of the IEEE-CIS; section 3 presents our solution proposal for the problem of Card-Not-Present (CNP) fraud; section 4 introduces the obtained classification and time performance results, as well as varied insights across interpretability and the direct comparison between S-SL and supervised learning; lastly, section 5 shows our final observations and future work that may be carried out.

[1] Tabular data refers to the distribution of transactions in rows and columns.

Latinx in AI @ NeurIPS-24 (LXAI 2024).

2 Related work

In this section, we present the literature about the use of the IEEE-CIS dataset to create smart credit card fraud detection systems based on ML and DL models. See Table 1 for a concise breakdown of the works discussed.

Table 1: Comparison between this work and the related state of the art about using the IEEE-CIS dataset for credit card fraud detection.

	Self-Supervised Learning	**Explainable AI**	**Time performance evaluation**
Najadat et al. [5], 2020	✗	✗	✗
Alkhatib et al. [6], 2021	✗	✗	✗
Nguyen et al. [7], 2022	✗	✗	✗
Bakhtiari et al. [8], 2023	✗	✔	✗
Jiang et al. [9], 2023	✗	✗	✗
This work	✔	✔	✔

Based on Table 1, our work is the first one using the IEEE-CIS and S-SL to tackle the CNP fraud problem, at its fundamental binary nature, to distinguish between legitimate and fraudulent transactions per customer. This S-SL solution translates into a cheaper and faster training process, in addition, it provides feature importance values for the explainability of the tested AI models, and not only presents classification performance but also time performance evaluation to validate its potential deployment into a real production network.

In the next section, details about the methodology followed for our solution proposal are presented.

3 Methodology

Here, we show information about the feature engineering process conducted over the IEEE-CIS dataset and the introduction to tabular learning using SubTab, both critical differentiators in our proposed solution.

3.1 IEEE-CIS dataset

To model user behavior, we decided to group the transactions per customer since not all people spend money in the same way. Moreover, we merged the product (e.g., transaction amount, location, date) and device type information (e.g., operating system, device version, browser/application to commit the transaction) into a single dataframe, and performed one-hot encoding over the categorical features. From the over 400 available variables, we mostly selected those that accurately describe how they were calculated and what they represent, aiding model interpretation and its transfer to other scenarios beyond this dataset. In addition, we created aggregate metrics (i.e., calculations of the average, maximum, and minimum values), achieving strong detection rates.

Finally, our feature engineering process derived 46 features, with 8 categorical and 38 numerical attributes, distributed across five different categories, namely customer identity, transaction amount, spatial and temporal information, and obfuscated meaning. To avoid data bias, features were scaled by applying standardization and min-max normalization.

In the end, 27,657 grouped transactions were used for training, and 7858 single transactions for testing. Grouped transactions per customer with more than 1 occurrence represented over 35% of the final training dataset, indicating recurrent purchase behavior. Additionally, 8.24% and 9.53% of the training and testing rows were fraud, respectively.

Therefore, due to the marked class imbalance, the Area Under the Receiver Operating Characteristic (AUROC) curve, and the macro-average version of the precision and F1 score metrics, were selected as a suitable way to balance the importance given to each class [10].

3.2 Tabular learning

SubTab works by dividing an initial set of features into small groups, where an overlap can exist. Then, these groups with fixed locations are passed into a common encoder to produce hidden representations (one per each subset of features), and finally fed into a shared decoder to calculate the reconstruction loss, and optionally to a projection network to get projections of the hidden representations to calculate contrastive and distance losses. At testing time, the former tuned encoder is now used and receives the testing subsets, where their latent representations are aggregated either by using minimum, maximum, average, or any other method, to get a joint representation [3].

See Fig. 1 for an overview diagram of our solution proposal. In the next section, the results obtained and their discussion are presented.

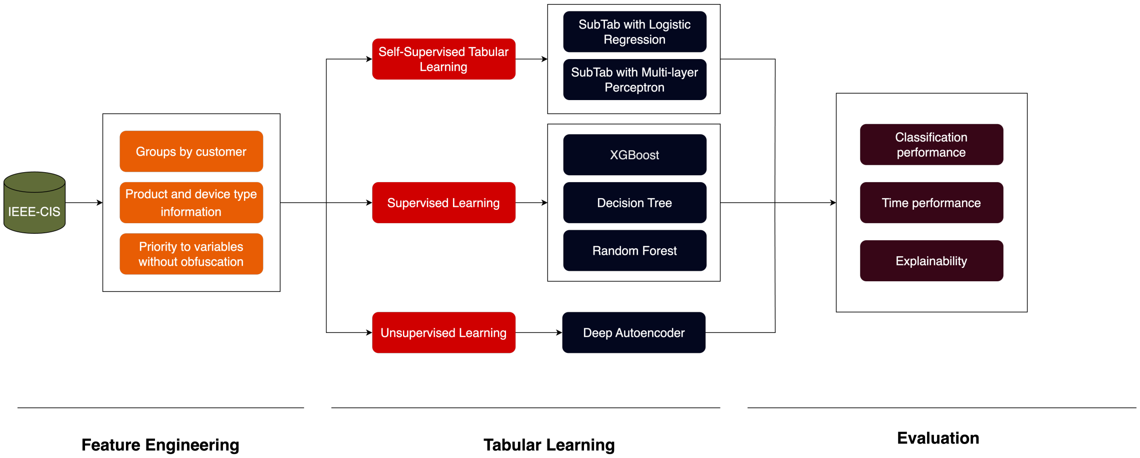

IEEE-CIS
Groups by customer
Product and device type information
Priority to variables without obfuscation
Self-Supervised Tabular Learning
Supervised Learning
Unsupervised Learning
SubTab with Logistic Regression
SubTab with Multi-layer Perceptron
XGBoost
Decision Tree
Random Forest
Deep Autoencoder
Classification performance
Time performance
Explainability
Feature Engineering
Tabular Learning
Evaluation

Figure 1: The priority in feature engineering was to capture customer behavior by grouping transactions by identity information like payment card and device data and also mostly to select those features where the meaning is not hidden.

4 Experimental results and discussion

In this section, we provide the classification and time performance results, as well as the feature importance values of different classifiers. All the metrics, except for explainability, are averaged through 9 iterations to report stability.

4.1 Classification performance

From Table 2, when benchmarking the linear evaluation of SubTab with the proposed non-linear evaluation using MLP, this latter is more stable and beats the AUROC performance by 1.33%, detecting 67.44% of the fraud cases. Nevertheless, in terms of macro F1 score, there is an average gap of 2.48%. In this case, if a bank wants to maximize the number of fraud cases detected, the proposed SubTab non-linear evaluation is the strongest option. Indeed, it is expected that this system does not work alone, where a bank's call center can handle non-legitimate cases with a double verification directly contacting the potential victims of CNP fraud, reducing the number of false positives.

Table 2: Classification results for the binary credit card fraud detection task. The proposed SubTab model with non-linear evaluation using MLP is the best architecture (bold value).

Model	AUROC	Macro Precision	Macro F1 score
Decision Tree	61.71% ± 0.31%	66.12% ± 1.19%	63.38% ± 0.48%
Random Forest	65.15% ± 0.85%	77.00% ± 0.79%	68.89% ± 0.90%
XGBoost	65.30% ± 0.51%	73.16% ± 3.76%	68.01% ± 0.40%
SubTab with Logistic Regression	66.11% ± 1.70%	79.05% ± 1.46%	70.07% ± 1.46%
SubTab with MLP	**67.44% ± 0.83%** ↑1.33%	68.33% ± 2.86%	67.59% ± 1.24%

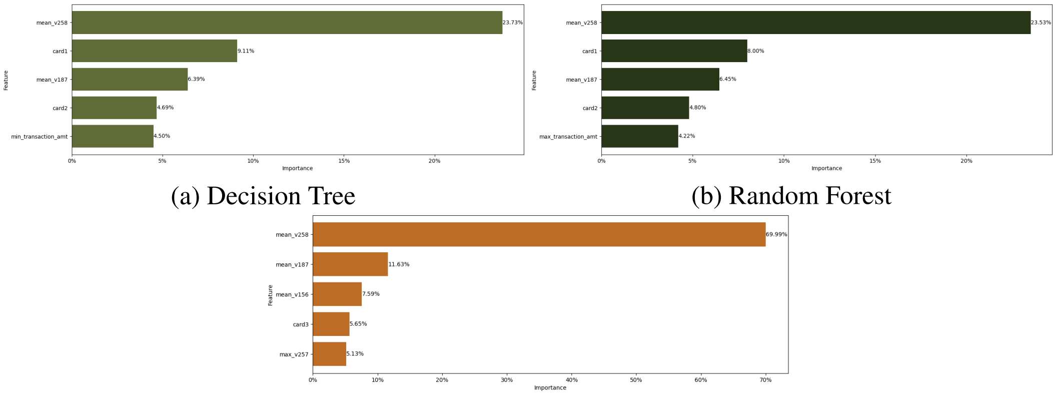
4.2 Time performance and feature importance

Another critical aspect of our CNP fraud system is how many payment card transactions it can handle per second and the time it takes to analyze each transaction. This is relevant because it allows us to define the scale and demand at which it can suitably work. From Table 3, Logistic Regression requires half the time of MLP to classify one instance, with an average speed of 0.32 ms / $transaction$, classifying twice the number of samples per second, with an average value of 8863 $transactions$ / $second$, indicating a strong speed to be validated on a production network.

Table 3: Time performance results across the SubTab classifiers for the binary credit card fraud detection task.

Model	transactions / second	ms / transaction
Logistic Regression	8863 ± 2285	0.32 ± 0.23
MLP	4016 ± 841	0.64 ± 0.55

Lastly, we leveraged the in-built explainability provided by the Decision Tree, Random Forest, and XGBoost models tested, with Fig. 2 showing the top 5 variables used per each architecture. Even though the mean of the obfuscated variable *v258* (i.e., *mean_v258*) is the most important one, in some cases by almost 70% as in Fig. 2(c), the aggregate metrics here proposed, and the priority given to select those features with a clear explanation of what they mean or how they were calculated, helped to highlight the customer identity variables {*card1*, *card2*, *card3*}, and the minimum and maximum aggregations of the transaction payment amount (i.e., {*min_transaction_amt*, *max_transaction_amt*}), in all the cases for over 4% of the importance given in the predictions, and in one of these cases to 9.11% importance.



(a) Decision Tree

(b) Random Forest

(c) XGBoost

Figure 2: Feature importance plots from the tree-based models. The top 5 variables used per architecture are shown.

4.3 Comparison with previous works

The works presented in section 2 obtain at least 85.56% AUROC in the IEEE-CIS, as in the case of [9]. However, these works do not mention the importance of prioritizing a feature selection process where the variables used have a clear meaning or calculation process that can later be replicated in other scenarios. Even though we at most detect 67.44% fraud, the feature engineering process to construct Hyphatia tries not to depend on obfuscated variables relying on aspects like spatial and temporal information, such as the average number of days between transactions per customer.

5 Conclusion and future work

Self-Supervised Tabular Learning using SubTab has demonstrated to be a strong model for the binary task of CNP fraud detection using the IEEE-CIS. As future work, since S-SL is strong for generalization purposes, we plan to transfer Hyphatia to other datasets with the aid of fine-tuning.

References

[1] Addison Howard, Bernadette Bouchon-Meunier, IEEE CIS, inversion, John Lei, Lynn@Vesta, Marcus2010, and Prof. Hussein Abbass. Ieee-cis fraud detection. https://kaggle.com/competitions/ieee-fraud-detection, 2019. [Accessed on 12-08-2024].

[2] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2023.

[3] Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. Subtab: Subsetting features of tabular data for self-supervised representation learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18853–18865. Curran Associates, Inc., 2021.

[4] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[5] Hassan Najadat, Ola Altiti, Ayah Abu Aqouleh, and Mutaz Younes. Credit card fraud detection based on machine and deep learning. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, pages 204–208, 2020.

[6] Khalid I. Alkhatib, Ahmad I. Al-Aiad, Mothanna H. Almahmoud, and Omar N. Elayan. Credit card fraud detection based on deep neural network approach. In *2021 12th International Conference on Information and Communication Systems (ICICS)*, pages 153–156, 2021.

[7] Nghia Nguyen, Truc Duong, Tram Chau, Van-Ho Nguyen, Trang Trinh, Duy Tran, and Thanh Ho. A proposed model for card fraud detection based on catboost and deep neural network. *IEEE Access*, 10:96852–96861, 2022.

[8] Saeid Bakhtiari, Zahra Nasiri, and Javad Vahidi. Credit card fraud detection using ensemble data mining methods. *Multimedia Tools and Applications*, 82(19):29057–29075, 2023.

[9] Shanshan Jiang, Ruiting Dong, Jie Wang, and Min Xia. Credit card fraud detection based on unsupervised attentional anomaly detection network. *Systems*, 11(6), 2023.

[10] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. https://arxiv.org/abs/2008.05756, 2020.