Team028 - IntelliFraud: Bank Account Fraud Detection

Emir Talundzic
etalundzic3@gatech.edu

Nukta Bhatia
nbhatia3@gatech.edu

Piyush Shivrain
pshivrain3@gatech.edu

Ashish Puri
apuri61@gatech.edu

Jagannath Banerjee
jbanerjee7@gatech.edu

Peter Kovari
pkovari3@gatech.edu

1 INTRODUCTION

IntelliFraud is a tool designed to identify fraudulent bank accounts originating from online applications in a consumer bank. The motivation behind its development lies in the significant costs incurred by banks due to fraudulent accounts. Tracing back individuals involved in fraudulent activities is challenging, time-consuming, and expensive [24]. According to the Federal Trade Commission (FTC), institutions incurred a cost of $8.8 billion that was passed to the consumers, marking a 44% surge from 2021[25]. Additionally, the presence of class imbalance in fraud accounts emphasizes the critical need for an effective and efficient fraud detection system like IntelliFraud.

2 PROBLEM DEFINITION

1) Analyze bank application data to identify trends and patterns, enabling bank analysts to visualize these trends based on specific feature selections.
2) Allow the analysts to compare different machine learning models, providing insights into their performance metrics aiding in informed model selection based on changing data.
3) Enable analysts to input application parameters and promptly determine the application's fraud potential, ensuring swift actions to prevent financial losses and secure banking operations.

3 LITERATURE SURVEY

Currently, financial fraud detection is accomplished through a combination of traditional and advanced techniques which includes transaction monitoring, Rule-Based systems, behavioral analysis, anomaly detection [10], [11], [19], [20], Machine Learning (ML) and Artificial Intelligence (AI) [1], [2], [3],[4], [18]. One of the major obstacles in developing a robust tool for detecting fraud comes from the inherent complexity of the data. Often times a large imbalance exists between the ratio of fraudulent transactions and genuine transactions, leading to an increase in the false positive rate and a poor customer experience due to misclassification [5], [6], [17].

In our research, we identified a crucial gap: The researchers have not used voting or stacking classifiers that takes into account multiple classifier prediction power [21] to get accurate predictions. Additionally, there is an absence of model prediction explainability. We have tried to bridge these gaps through our project.

4 PROPOSED METHOD

(a) Intuition

Our strategy for tackling fraudulent bank applications capitalizes on the underutilized potential of ensemble methods like voting and stacking classifiers for fraud detection. While existing models mainly rely on standard machine learning techniques [**1**], [**2**], [**3**], [**4**], [**10**], [**11**], [**18**], [**19**], [**20**], [**23**], our research highlights the untapped power of voting and stacking classifiers. It combines the strengths of multiple models, introducing a novel dimension to the realm of fraud detection. Through our intuitive user interface, we are enabling analysts to dive deeper into the data through EDA, evaluate multiple models, real time prediction to experiment and verify fraud. This adaptability forms the backbone of our dynamic fraud detection strategy.

Additionally, through the use of graph network visualization in Figure 1, we provide the bank analysts with a comprehensive view of fraudulent transaction flows across various subsystems in the current data.

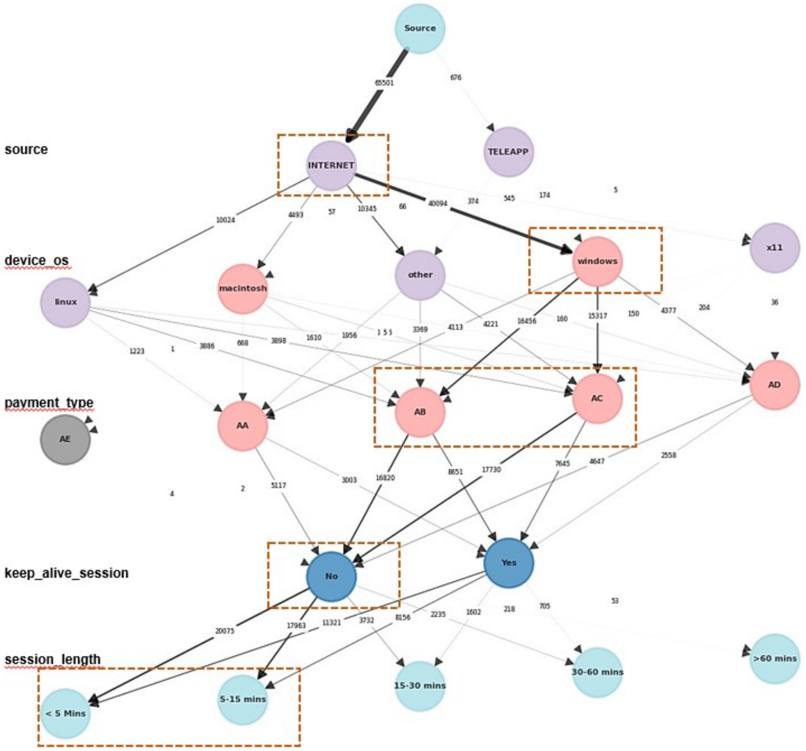


Figure 1: Fraudulent Transaction Flow - Network Visualization

(b) Description

DATA

We employed the Bank Account Fraud Dataset Suite (NeurIPS 2022) [**22**] [Click Dropbox to open the link.], utilizing all of the data variants available in the dataset. The

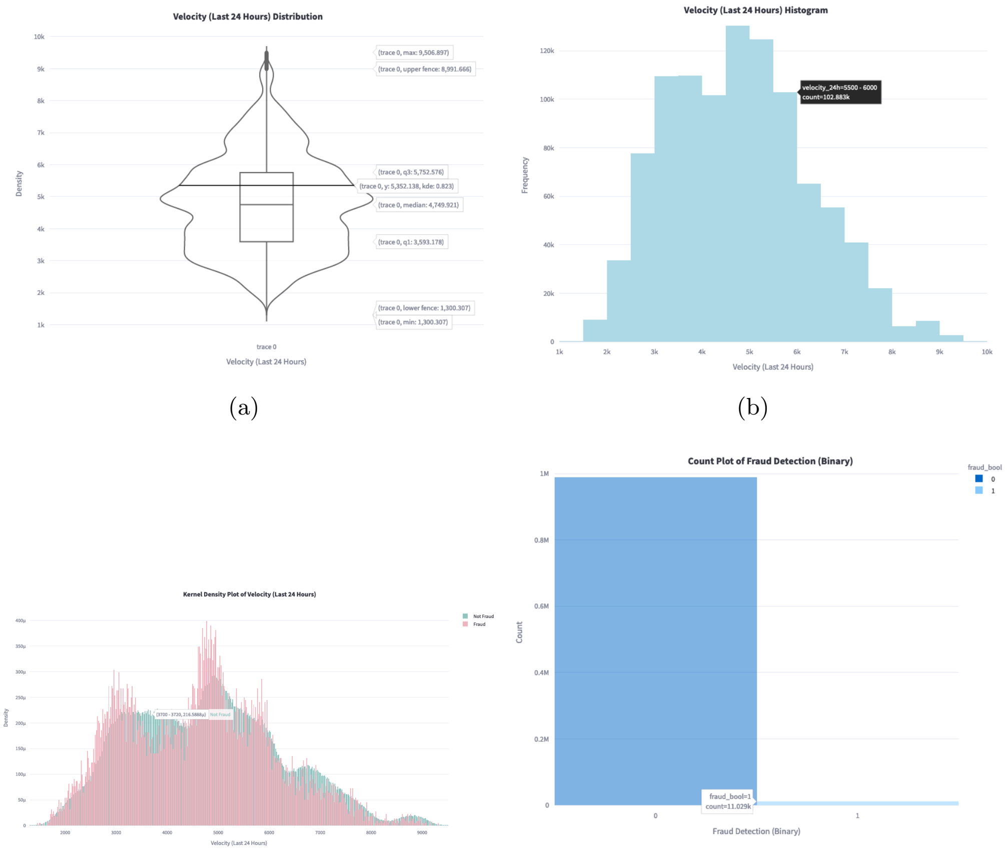
Bank Account Fraud Dataset is crafted for fraud detection from real-world data. This dataset underwent privacy techniques to ensure applicant identity protection, encompassing noise reduction and feature encoding. The description of different variants of data are provided in Table 1. The collection of these data variants comprises approximately 6 million rows, totaling around 1.4GB in disk size. The imbalance within the dataset is evident, as depicted in Figure 2(d).

Base	Sampled to represent Original Dataset
Variant I	Has higher group size disparity than base.
Variant II	Has higher prevalence disparity than base.
Variant III	Has better separability for one of the groups.
Variant IV	Has higher prevalence disparity in train.
Variant V	Has better separability in train.

Table 1: Dataset Variants

EDA

As shown in Figure 2, we present the user with interactive violin plots, histograms, Kernel Density Estimation(KDE) plots (if applicable) or an interactive counts plot allowing the user to quickly gain an understanding of the underlying data.



(a)

(b)

(c)

(d)

Figure 2: (a) Violin Plot (b) Histogram Plot (c) KDE Plot (d) Count Plot

MODELLING

Our Modeling page provides users with the ability to quickly select a model, visualize model metrics, and assess feature importance. When constructing the model, we utilized all the datasets with 33 independent variables, which is highly imbalanced, with less than 1 percent of the data labeled as fraud Figure 2(d). Top 11 features Figure 3 (Description available in Table 2) were selected through variance threshold test, correlation matrix and random forest feature importance method.

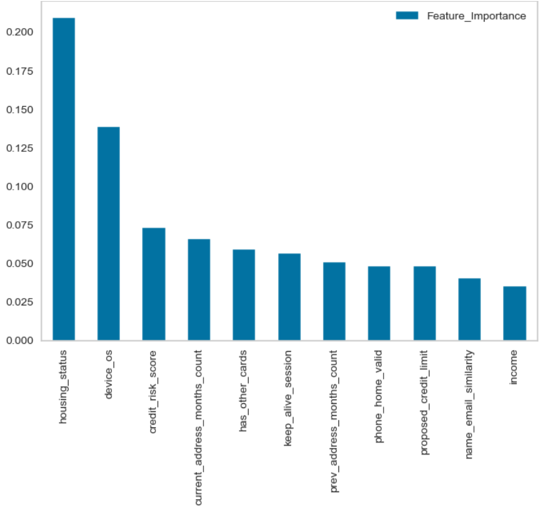


Figure 3: Top 11 Features Selected

Feature	Description
housing_status	Current residential status for applicant.7 possible annonymized values.
device_os	Operative system of device that made request. Possible values Windows, macOS, Linux, X11, or other.
credit_risk_score	Internal score of application risk. Ranges between [-191, 389].
current_address_months_count	Months in currently registered address of the applicant. Ranges between [-1, 429] months. -1 is a missing value.
has_other_cards	If applicant has other cards from the same banking company.
keep_alive_session	User option on session logout.
prev_address_months_count	Number of months in previous registered address of the applicant, i.e. the applicant's previous residence, if applicable. Ranges between [-1, 380] months.-1 is a missing value.
phone_home_valid	Validity of provided home phone.
proposed_credit_limit	Applicant's proposed credit limit. Ranges between [200, 2000].
name_email_similarity	Metric of similarity between email and applicant's name. Higher values represent higher similarity. Ranges between [0, 1].
income	Annual income of the applicant (in decile form). Ranges between [0.1, 0.9].

Table 2: Top 11 Features Data Description

Subsequently, we conducted under-sampling of fraud versus non-fraud instances in various ratios such as 1:1, 1:2, and 1:3. For each set, 80% of the data was split into training data and 20% into our test data while implementing 5-fold cross-validation. The classifiers used included LightGBM, XGBoost, AdaBoost, a VotingClassifier, and a StackingClassifier. The performance was evaluated based on the Recall metric and ROC Score for fraud detection.

Furthermore, we fine-tuned LightGBM, XGBoost, and AdaBoost using Grid Search

Hyperparameter tuning. The VotingClassifier made predictions through majority voting, while the StackingClassifier combined multiple classification models using a meta-classifier.

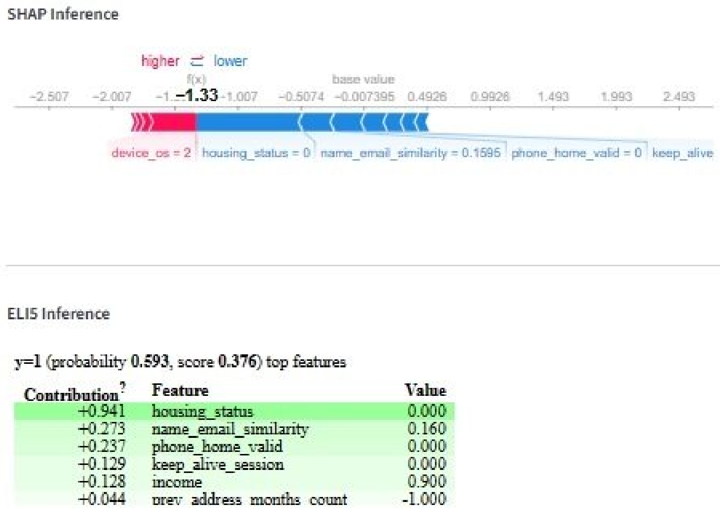
The ROC of the Voting and Stacking Classifiers was higher than that of individual classifiers for the 1:1 sampled data, while the recall for fraud detection remained unchanged across all models. Cross-validation scores for all classifiers were consistent with the test set. A detailed overview of all models can be seen in Table 3.

Sample Class	Classifier	5-Fold CV Score	Recall (Fraud)	ROC Score
1:1	XGB	0.783	0.785	0.784
1:1	AdaBoost	0.785	0.776	0.785
1:1	LGBM	0.782	0.783	0.781
1:1	Voting	0.787	0.784	0.785
1:1	Stacking	0.787	0.784	0.785
1:2	XGB	0.797	0.649	0.763
1:2	AdaBoost	0.797	0.622	0.758
1:2	LGBM	0.794	0.613	0.753
1:2	Voting	0.799	0.636	0.761
1:2	Stacking	0.799	0.637	0.761
1:3	XGB	0.820	0.542	0.730
1:3	AdaBoost	0.819	0.515	0.724
1:3	LGBM	0.816	0.496	0.717
1:3	Voting	0.821	0.527	0.728
1:3	Stacking	0.821	0.532	0.730

Table 3: Classifier Metrics for Different Sample Classes

INFERENCE USER INTERFACE

The Inference Page offers a user-friendly interface with a model selection dropdown and user-friendly sliders and text boxes for inputting feature values generated through Feature Engineering. Users can use these features to evaluate an application for potential fraud. The page presents prediction results, confidence levels, and provides a clear explanation of model predictions using Shapley values and Eli5 (Figure 4).



SHAP Inference

ELI5 Inference

y=1 (probability 0.593, score 0.376) top features

Contribution?	Feature	Value
+0.941	housing_status	0.000
+0.273	name_email_similarity	0.160
+0.237	phone_home_valid	0.000
+0.129	keep_alive_session	0.000
+0.128	income	0.900
+0.044	prev_address_months_count	-1.000

Figure 4: SHAP and Eli5 for 1:1 Sampling

5 EXPERIMENTS/EVALUATION

(a) Description of Testbed

The testbed incorporates three components, including the dataset suite, user interface, and the underlying algorithms like XGBoost, Light GBM, AdaBoost, Voting and Stacking classifier Model. Following are some of the questions that our experiments are designed to answer using IntelliFraud Tool
1) What are the key features influencing fraud detection? [Figure 3]
2) How do different machine learning algorithms perform in terms of accuracy, precision, and recall? Which algorithm yields the most reliable fraud predictions? [Table 3]
3) Can users effectively choose models based on observed performance, enhancing overall system accuracy? [Modelling UI]

(b) Experiments & Observations

In the course of our experiments, a comprehensive examination of the data revealed noteworthy observations. Leveraging graph networks, we have conducted visualization and analysis of fraudulent transaction flows, aiming to identify anomalous paths or cycles that could signify potential fraudulent activities. The visual representation in Figure 1 illustrates a compelling insight: a predominant portion of fraudulent transactions originates from the Internet, characterized by activity through the Windows Operating System and payment types AB & AC. Notably, these fraudulent actors consistently terminate their sessions in less than 5 minutes, further emphasizing the utility of graph networks in uncovering and understanding sophisticated fraud patterns. Additionally, a correlation analysis highlighted a strong interdependence between velocity_4w (average number of applications per hour in the last 4 weeks) and the month when the application was made. It's worth noting that the stacking classifier exhibited significant computational intensity, running with O(n)2 complexity. Despite achieving slightly superior performance compared to LightGBM, XGBoost, and AdaBoost, both Voting and Stacking classifiers presented challenges in terms of interpretability, emphasizing the trade-off between performance and transparency in model selection.

6 CONCLUSIONS & DISCUSSIONS

Intellifraud, is a Fraud Detection System leveraging graph network analysis and machine learning classifers (voting & stacking), to identify and prevent fraudulent activities. The system offers users an intuitive interface with detailed statistical analyses. With features like interactive dashboards and comprehensive metric evaluations, inference explanations, it ensures efficient and accurate identification of fraudulent patterns in various transactions. All team members have contributed a similar amount of effort.

Issues & Limitations

We discovered that the stacking classifier was computationally intensive, running with O(n)2 complexity.Even though Voting & Stacking classifiers performed slightly better than LightGBM, XGBoost and AdaBoost on this dataset, they are difficult to interpret.

Future Extension

Future work will involve conducting further exploration of the stacking classifier investigating alternative meta-learners and experimenting with diverse combinations of base models to optimize the overall model performance. Additionally, we aim to enhance model interpretability by integrating Local Interpretable Model-Agnostic Explanations (LIME) with SHAP features, enabling users to gain nuanced insights into individual predictions. Further enhacements can be made by integration of real-time data from financial institutions, establishing a continuous feedback loop for ongoing model refinement in dynamic financial environments.

References

[1] Paruchuri, H. (2017). Credit Card Fraud Detection using Machine Learning: A Systematic Literature Review. ABC Journal of Advanced Research, 6(2), 113-120

[2] Lucas, Y., & Jurgovsky, J. (2020). Credit card fraud detection using machine learning: A survey. arXiv preprint arXiv:2010.06479.

[3] Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407.

[4] Kou, Y., Lu, C. T., Sirwongwattana, S., Huang, Y. P. (2004, March). Survey of fraud detection techniques. In IEEE International Conference on Networking, Sensing and Control, 2004 (Vol. 2, pp. 749-754). IEEE.

[5] Zojaji, Z., Atani, R. E., & Monadjemi, A. H. (2016). A survey of credit card fraud detection techniques: Data and technique oriented perspective. arXiv preprint arXiv:1611.06439.

[6] Ahmed, M., Mahmood, A. N., & Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. Future Generation Computer Systems, 55, 278-288.

[7] Agrawal, S., & Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. Procedia Computer Science, 60, 708-713.

[8] Bin Sulaiman, R., Schetinin, V., & Sant, P. (2022). Review of machine learning approach on credit card fraud detection. Human-Centric Intelligent Systems, 2(1-2), 55-68.

[9] Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. arXiv preprint arXiv:1009.6119.

[10] Guha, S., Mishra, N., Roy, G., & Schrijvers, O. (2016, June). Robust random cut forest based anomaly detection on streams. In International conference on machine learning (pp. 2712-2721). PMLR.

[11] Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. Artificial intelligence review, 22, 85-126.

[12] Olowookere, T. A., & Adewale, O. S. (2020). A framework for detecting credit card fraud with cost-sensitive meta-learning ensemble approach. Scientific African, 8, e00464.

[13] West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: a comprehensive review. Computers & security, 57, 47-66.

[14] Thaifur, A. Y. B. R., Maidin, M. A., Sidin, A. I., & Razak, A. (2021). How to detect healthcare fraud?"A systematic review". Gaceta sanitaria, 35, S441-S449.

[15] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga and N. Kuruwitaarachchi, "Real-time Credit Card Fraud Detection Using Machine Learning," 2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence), Noida, India, 2019, pp. 488-493, doi: 10.1109/CONFLUENCE.2019.8776942.

[16] Jesus, S., Pombal, J., Alves, D., Cruz, A., Saleiro, P., Ribeiro, R., ... & Bizarro, P. (2022). Turning the tables: Biased, imbalanced, dynamic tabular datasets for ml evaluation. Advances in Neural Information Processing Systems, 35, 33563-33575.

[17] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2017). Credit card fraud detection: a realistic modeling and a novel learning strategy. *IEEE transactions on neural networks and learning systems*, 29(8), 3784-3797.

[18] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, ”Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms,” in IEEE Access, vol. 10, pp. 39700-39715, 2022, doi: 10.1109/ACCESS.2022.3166891.

[19] W. Fang, X. Li, P. Zhou, J. Yan, D. Jiang and T. Zhou, ”Deep Learning Anti-Fraud Model for Internet Loan: Where We Are Going,” in IEEE Access, vol. 9, pp. 9777-9784, 2021, doi: 10.1109/ACCESS.2021.3051079.

[20] Y. Yang, Y. Yu and T. Li, ”Deep Learning Techniques for Financial Fraud Detection,” 2022 14th International Conference on Computer Research and Development (ICCRD), Shenzhen, China, 2022, pp. 16-22, doi: 10.1109/ICCRD54409.2022.9730314.

[21] Carneiro, T., da Silva, R. C., & Pardo, T. A. S. (2017). A comparative study of techniques for financial fraud detection using a boosting algorithm. *IEEE Latin America Transactions*, 15(3), 540-546.

[22] Bank Fraud Data https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022

[23] Jesus, S., Pombal, J., Alves, D., Cruz, A., Saleiro, P., Ribeiro, R. P., Gama, J., & Bizarro, P. (2022). Turning the Tables: Biased, Imbalanced, Dynamic Tabular Datasets for ML Evaluation

[24] April 13, S. G., & 2022. (n.d.). New Fraud on the Block Causes Bank Losses to Rise. Www.bankinfosecurity.com. https://www.bankinfosecurity.com/new-fraud-on-block-causes-bank-losses-to-mount-a-18867

[25] O’Brien, S. (2023, March 1). Fraud cost consumers $8.8 billion last year, Federal Trade Commission says. That’s up 44% from 2021. CNBC. https://www.cnbc.com/2023/03/01/ftc-fraud-cost-consumers-8point8-billion-in-2022.html