# Boosting Algorithms for Credit Card Fraud Detection Across Varied Datasets

TRANSPORT AND TELECOMMUNICATION INSTITUTE

Author: Justs Vīdušs
Study Programme: Master of Computer Science: Data Analytics and Artificial Intelligence
Supervisor: Spiridovska Nadežda, Dr. sc. ing., asoc. profesors

## Introduction

The rise in digital banking has amplified the complexity and prevalence of fraudulent activities, placing financial institutions, especially smaller Boones, at significant risk. This research investigates the potential of big data analytics and advanced boosting algorithms to enhance fraud detection systems, crucial for safeguarding financial assets and maintaining consumer trust.

By comparing the efficacy of algorithms like XGBoost, AdaBoost, and Gradient osting Machine against traditional methods such as Random Forest, KNN, and SVM, this study aims to identify cost-effective, high-performing solutions suitable for small and medium-sized banks. The findings will provide actionable insights into the adoption of advanced fraud detection technologies, highlighting their benefits in accuracy, speed, and resource efficiency.

- Research Question:

How can boosting algorithms like XGBoost, AdaBoost, and Gradient Boosting Machine enhance the effectiveness and efficiency of fraud detection systems compared to traditional methods such as Random Forest, KNN, and SVM in the banking sector?

- Objectives:

1. Review existing research
2. Compare boosting and traditional algorithms on three different datasets
3. Investigate impact of dataset size on model performance
4. Evaluate the computational requirements for each algorithm
5. Analyze resuts, make conclusions and recommendations

## Methodology

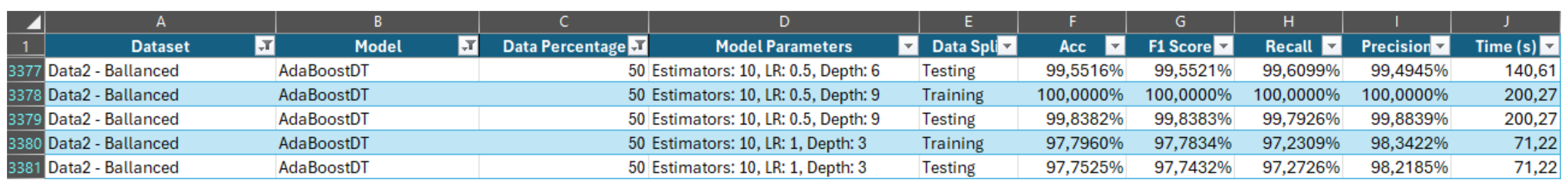
- Study Design: This study employs a quantitative research design, leveraging data analytics and machine learning techniques to evaluate the performance of various fraud detection algorithms.

- Materials and Procedures:
  - **Tools**: Python Jupyter Notebooks for algorithm testing, data cleaning, and preparation.
  - **Datasets**: CSV files obtained from Kaggle and other sources, preprocessed and stored in the same format.
  - **Software**: Python for model training, data processing, and result aggregation.
  - **Storage**: All project artifacts, including code, datasets, and results, were saved and made publicly accessible via GitHub.

- Data Collection and Analysis:
  - **Data Collection**: Datasets were downloaded from Kaggle and other sources. Preprocessed datasets were stored in CSV files.
  - **Model Training**: Various algorithms, including boosting algorithms (XGBoost, AdaBoost, Gradient Boosting Machine) and traditional methods (Random Forest, KNN, SVM), were trained and evaluated.
  - **Result Aggregation**: Training results were automatically saved in Excel files using Python. Pivot tables were created to aggregate results and visualize the performance of each model variant.
  - **Performance Metrics**: Key metrics such as Accuracy, F1 Score, Recall, Precision, and Time Taken were recorded for each iteration.
  - **Documentation**: All results and analyses were documented and uploaded to GitHub for transparency and public access.

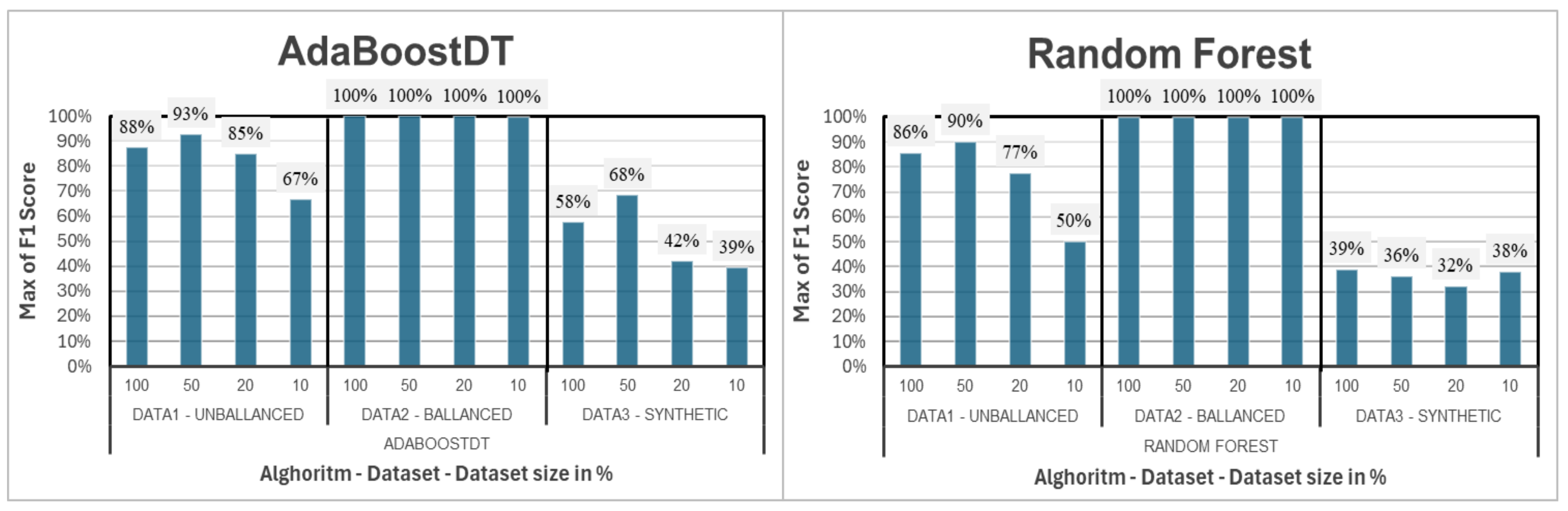
## Results and Discussion

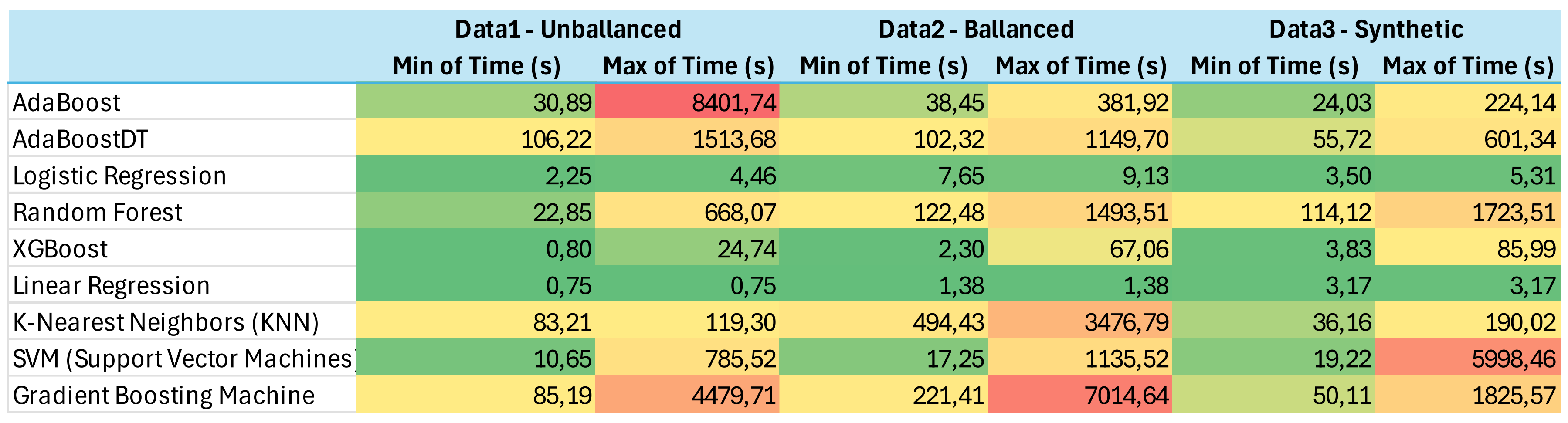
- Numerical results: all 3582 test results containing variety of tested algorithms on different parameters, performance and training times available in Github repository. Example:



| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Dataset | Model | Data Percentage | Model Parameters | Data Spli | Acc | F1 Score | Recall | Precision | Time (s) |
| 3377 | Data2 - Ballanced | AdaBoostDT | 50 | Estimators: 10, LR: 0.5, Depth: 6 | Testing | 99,5516% | 99,5521% | 99,6099% | 99,4945% | 140,61 |
| 3378 | Data2 - Ballanced | AdaBoostDT | 50 | Estimators: 10, LR: 0.5, Depth: 9 | Training | 100,0000% | 100,0000% | 100,0000% | 100,0000% | 200,27 |
| 3379 | Data2 - Ballanced | AdaBoostDT | 50 | Estimators: 10, LR: 0.5, Depth: 9 | Testing | 99,8382% | 99,8383% | 99,7926% | 99,8839% | 200,27 |
| 3380 | Data2 - Ballanced | AdaBoostDT | 50 | Estimators: 10, LR: 1, Depth: 3 | Training | 97,7960% | 97,7834% | 97,2309% | 98,3422% | 71,22 |
| 3381 | Data2 - Ballanced | AdaBoostDT | 50 | Estimators: 10, LR: 1, Depth: 3 | Testing | 97,7525% | 97,7432% | 97,2726% | 98,2185% | 71,22 |

- Graphs and Images:





| | Data1 - Unballanced | | Data2 - Ballanced | | Data3 - Synthetic | |
|---|---|---|---|---|---|---|
| | Min of Time (s) | Max of Time (s) | Min of Time (s) | Max of Time (s) | Min of Time (s) | Max of Time (s) |
| AdaBoost | 30,89 | 8401,74 | 38,45 | 381,92 | 24,03 | 224,14 |
| AdaBoostDT | 106,22 | 1513,68 | 102,32 | 1149,70 | 55,72 | 601,34 |
| Logistic Regression | 2,25 | 4,46 | 7,65 | 9,13 | 3,50 | 5,31 |
| Random Forest | 22,85 | 668,07 | 122,48 | 1493,51 | 114,12 | 1723,51 |
| XGBoost | 0,80 | 24,74 | 2,30 | 67,06 | 3,83 | 85,99 |
| Linear Regression | 0,75 | 0,75 | 1,38 | 1,38 | 3,17 | 3,17 |
| K-Nearest Neighbors (KNN) | 83,21 | 119,30 | 494,43 | 3476,79 | 36,16 | 190,02 |
| SVM (Support Vector Machines | 10,65 | 785,52 | 17,25 | 1135,52 | 19,22 | 5998,46 |
| Gradient Boosting Machine | 85,19 | 4479,71 | 221,41 | 7014,64 | 50,11 | 1825,57 |

- Key Findings:
- **Unballanced dataset**: AdaBoost achieved the highest recall rate (83.7%) indicating its superior performance in detecting fraud in skewed data.
- **Balanced Dataset**: KNN, XGBoost, and Random Forest achieved 100% recall, demonstrating their effectiveness where class distribution is even.
- **Synthetic Datasets**: Gradient Boosting Machine led with a recall of 72.2%, showing robustness against anomalies.Computational
- **Efficiency:** Boosting algorithms required more processing time due to their complexity, affecting their practical feasibility.

The findings suggest that financial institutions should tailor their fraud detection approach based on the specific characteristics of their datasets. Smaller banks with limited resources might opt for simpler algorithms like Logistic Regression, while larger institutions with higher importance in accuracy could benefit from the more complex boosting algorithms.

## Conclusions / Outcomes

### ✓ Boosting Algorithms Excel in Unbalanced Datasets

Boosting algorithms, particularly AdaBoost, significantly outperform traditional methods in detecting fraud within unbalanced datasets. AdaBoost achieved the highest recall rate of 83.7%, making it highly effective for identifying fraudulent transactions amidst skewed data distributions.

### ✓ Traditional Algorithms Perform Well in Balanced Datasets

In balanced datasets, traditional algorithms like KNN, XGBoost, and Random Forest achieved exeptional recall rates of 99.99%, demonstrating their robustness in environments where class distribution is even. This highlights their suitability for scenarios where data balance can be maintained.n

### ✓ Computational Efficiency and Practical Feasibility

The study found that while boosting algorithms provide higher accuracy, they come at the cost of increased computational resources and time. Traditional methods, though less computationally intensive, may be preferred by smaller institutions with limited resources.

## Key References

- Github Repository: https://github.com/Justs-Viduss/Master-Thesis
- Publication: Step into the Future Journal, Volume 19, Issue 1, p. 34

## Contacts

LinkedIn