National College of Ireland

When Simpler Models Outperform Hybrid Ensemble: Re-evaluating Hybrid Ensembles for Financial Fraud Detection

MSc Research Project
Data Analytics

Sanidhya Bajaj
Student ID: x23251867

School of Computing
National College of Ireland

Supervisor: Prof. Shubham Subhnil

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sanidhya Bajaj

Student ID: X23251867

Programme: MSc in Data Analytics **Year:** 2024 – 2025

Module: Research Practicum

Supervisor: Prof. Shubham Subhnil

Submission Due Date: 11th Aug 2025

Project Title: When Simpler Models Outperform Hybrid Ensemble: Re-evaluating Hybrid Ensembles for Financial Fraud Detection

Word Count: 802 **Page Count**.....7.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sanidhya Bajaj

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sanidhya Bajaj
Student ID: X23251867

1 Introduction

The following guide entails how the environment is configured to execute the code that has been developed to predict financial fraud using machine learning models and their hybrids. Solution is created using Python and requirements of the program have been defined to ensure that they can be easily reproduced.

2 System specification

The development and testing of the Fraud Detection Machine Learning Model Comparison project make use of the following hardware and software configurations:

Hardware specifications:

- Processor: AMD Ryzen 5 comparable or i5 (8 th generation or higher)
- OS: Ubuntu 18.04+, macOS 10.14+, or Windows 10/11
- RAM: 8 GB minimum (16 GB recommended to work well).
- Storage: 5 gb of free space

The library is to be installed, and the datasets are to be downloaded using the internet connection.

Software requirements:

- At least version 3.8 of Python
- Development Environment: either Google Colab or Jupyter Notebook
- Package Manager: Anaconda

3 Dataset

3.1 Dataset involved in Source:

- Detection of fraud using transactions data set.
- Kaggle is the repository for the community data set.
- The data you can use is at kaggle.com/datasets/samayashar: the fraud-detection-transactions-dataset.

3.2 Dataset Loding:

After creating a Kaggle account it is possible to download the wildfire dataset [1] manually by visiting its data webpage. Once downloaded, extract the contents in the ZIP file in the

working directory you have in your computer. Be sure that the extracted CSV file is accessible to your active working directory, then launch your favourite coding environment, e.g. VS Code, Jupyter Notebook, or Google Colab. As soon as one enters a command such as import pandas as pd and then enters df = pd. read csv("wildfire_dataset.csv"), the dataset can be imported using the panda's library, but one must ensure that the file name and path corresponds to the point where he/she extracts the file.

With the help of file upload feature of Google Colab, upload the CSV file extracted locally on your computer first (from google.colab import files, followed by files.upload()). Thereafter, ensure that the file appears on file explorer of the session. Put it in the working directory or use uploaded path in case of need. Once this file is prepared, take it into a DataFrame, and the name of the file must be the same, that is, df = pd.read_csv ("wildfire_dataset.csv").

3.3 Description:

- Two-class classification (fraud or non-fraud) of tabular transaction data.
- Size: It is loaded in this project, and it consists of 50,000 rows with 21 columns.
- Label Fraud_Label = the target variable (0 = legitimate, 1 = fraud).
- Timestamp: The transactions are added with the timestamp of the whole year 2023 by using the provided Timestamp parameter.
- Data source: Synthetic generated data set to resemble real fraud patterns and behaviour of users and transactions.

4 Execution of code

4.1 Import necessary libraries:

- Pandas: Data processing, cleaning and manipulation of Tabular data
- NumPy: Array operations and numbers of calculations
- Matplotlib: Representational of static data (charts, graphs)
- Seaborn: Better display of statistics data
- Scikit-learn: Preprocessing, evaluation metrics and learning algorithms
- Installation command:

  pip install pandas numpy matplotlib seaborn scikit-learn

4.2 Aspects of Machine Learning

The modules of scikit-learn that are used are listed below:

a) Data Preprocessing

- train_test_split - Splits the dataset into training and testing.
- OneHotEncoder can encode categorical variables.
- StandardScaler normalizes the numerical features.

b) Classification models

- Decision Tree (DecisionTreeClassifier)
- Random Forest (RandomForestClassifier)
- Gradient Boosting (GradientBoostingClassifier)
- AdaBoost (AdaBoostClassifier)
- Bagging (BaggingClassifier)
- Support Vector Machine (SVC)
- Naïve Bayes (GaussianNB)
- K-Nearest Neighbors (KNeighborsClassifier)
- Extra Trees (ExtraTreesClassifier)
- Logistic Regression (LogisticRegression)

c) Hybrid Methods

- Voting Classifier: (Hard and soft)
- Stacking Classifier

d) Performance Measure

- Accuracy
- Precision
- Recall
- F1_score
- AUC_ROC

e) Pseudocode

BEGIN
LOAD dataset using pandas

PERFORM Exploratory Data Analysis (EDA):
- Generate summary statistics
- Visualize feature distributions
- Identify correlations and patterns
- Detect class imbalance

HANDLE missing values:
- Impute or remove as appropriate

PREVENT data leakage:
- Apply transformations (scaling, encoding) only on training data
- Avoid using target-related information in feature engineering

ENCODE categorical variables using OneHotEncoder
SCALE numerical features using StandardScaler (fit on training data, transform both sets)

SPLIT dataset into training and testing sets

INITIALIZE multiple classifiers:
Decision Tree, Random Forest, Gradient Boosting,
AdaBoost, Bagging, SVM, Naïve Bayes, KNN etc.

FOR each classifier:
TRAIN on training set
PREDICT on testing set
EVALUATE using accuracy, precision, recall, F1, ROC-AUC

CRITERIA FOR SELECTION:
- Evaluate all base classifiers on validation set
- Rank models by multiple metrics (Accuracy, F1-score, ROC-AUC)
- Identify "strong learners":
Models with consistently high performance across metrics
- Identify "weak learners":
Models with moderate performance but complementary error patterns

COMBINE selected models using ensemble methods: (Strong + Strong, Weak + Strong, Weak+ Weak)

- Voting Classifier (Hard and Soft)
- Stacking Classifier

OUTPUT:
- Comparative performance metrics
- Visualizations of results

END

References

[1] **Ashar, S. (Samayashar)**, 2025. *Fraud Detection Transactions Dataset*. Kaggle [dataset]. Available at: https://www.kaggle.com/datasets/samayashar/fraud-detection-transactions-dataset