University of Toronto
ECE496 Testing Document

Credit Card Fraud Detection

Team Number: 2022524

Supervisor: Zeb Tate

Shadman Kaif, Abdurrafay Khan, Krutarth Patel, Shanthosh Sivayogaligam

Executive Summary

The final model and any working prototypes will be tested and validated to check if it satisfies the proposed project goals. The team has outlined the set of requirements, descriptions, a plan, the methods and a success criteria to carry out effective tests. Following the implementation plan, there was one main function and three main objectives that had outlined metrics to indicate a successful model: detecting fraudulent credit card transactions using ensemble learning, minimizing the amount of false positives, being accurate, and meeting a desired processing speed respectively. For all tests, the team will require a computer, connection to the internet, access to Google Colab and a specific IBM dataset. These main objectives will further require built-in Python libraries to measure false positive ratios, accuracy and as well as processing time to test. Considering the project is completely software based, the majority of the requirements fall within similar software testing tools.

The team has created descriptions to further describe how each test will be conducted and specifically which software tools will be applied. This regards to the variety of Python libraries used such as Tensorflow, Pandas, Numpy, Sklearn, Keras and math. A testing plan and the testing methods for the functionality and each objective has been created to outline exactly in what procedure and manner each test will need to be conducted. A successful final solution or working prototype will have to meet the following criteria: classify fraudulent and legitimate transactions, meet a 20-30% false positive rate, 95% model accuracy and process 65,000 transactions/s. The team is also in-line with the implementation plan and no changes were made to the project that will alter testing.

Table of Contents

Requirements for Testing

The table below outlines the design specification, the type of design specification (functional, objective, or constraint), and requirements for each testing item.

No.	Design Specification	Type	Requirement for Testing
1	Detect fraudulent credit card transactions using ensemble learning	Functional	Require access to a computer with connection to the internet, Google Colab software and the IBM dataset (applies as a requirement for testing of all design specifications). Use built in Python libraries within Colab to run the test data through the final model (applies as a requirement for testing of all design specifications). Given testing data, predict fraudulent and legitimate transactions.
2	Minimize the amount of false positives	Objective	Take final predictions from the trained model and use built-in fit function to create false positive ratio. Compare to success criteria and adjust variables accordingly until desired outcome.
3	Be accurate	Objective	Use built-in fit function to determine accuracy of model on testing data.
4	Processing speed	Objective	Run testing data set over final model. Use built-in time library to record initial and final times for when the test set is passed in and when the outputs are received. The difference of the times will provide processing speed.

Table 1. Requirements for testing

Description of Testing

The following list contains descriptions for each of the tests that will be conducted on our design.

1. Detect fraudulent credit card transactions using ensemble learning
    a. Imported Python libraries into the Google Colab and connect to Google Colab's GPU
    b. Loaded the testing dataset (part of the IBM dataset)
    c. Processed the testing data to convert categorical data (time, fraud, decimal, amount) into numerical data interpretable by the model
    d. Loaded AdaBoost model (containing the designed LSTM model)
    e. Ran the testing data over the AdaBoost model to get final predictions
2. Minimize the amount of false positives
    a. Used the final predictions obtained through Item (1) to calculate the False Positive Ratio (%) (false positive transactions detected within the testing dataset)
3. Be accurate
    a. Ran the evaluate function to measure and record model accuracy
4. Processing Speed
    a. Measured time before and after running the testing data over the Adaboost model
    b. Used the two measured times to calculate the processing speed of the training data by our model

Test Plan

The setup needed to effectively test and experiment requires the successful operation of the softwares used. The setup consists of the usage of an internet connection to be able to use Google Colab. The IBM dataset, Python libraries, and Google Drive will be needed to be used in Google Colab to conduct the tests. Figure 1 shows the setup required to conduct the testing.

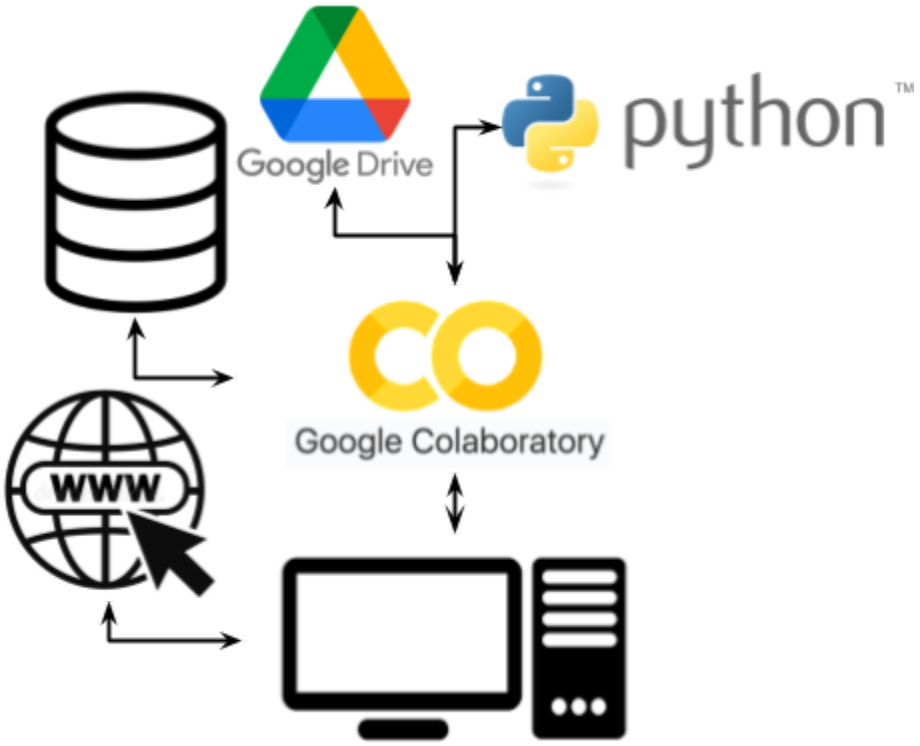

Figure 1. Configuration of the test plan setup

Test Method

For each of the four design specifications, there is a systematic test plan required to replicate the testing process. The description for the step-by-step testing procedure for each test is written below.

1. Detect fraudulent credit card transactions using ensemble learning

   The following procedure applies for the creation of the AdaBoost ensemble learning model:

   a. Import necessary Python libraries on Google Colab for the end-to-end workflow: Tensorflow, Pandas, Numpy, Sklearn, Keras and math.

   b. Load the test data csv file containing a preprocessed subset of 20% of the IBM Credit Card Dataset instantiated at the time of training.

   c. Connect to Google Colab's GPU.

   d. Perform data transformations to convert the categorical data to numerical data.

   i. Use the Sklearn's DataMapper function around the dataset's column entries.

   ii. Create 4 encoder functions - time encoder, fraud encoder, decimal encoder, and amount encoder to convert the date, time, amount, and fraud columns.

   iii. Use Sklearn's built-in OneHotEncoder function to convert the zip, merchant name,

merchant state, use chip, errors and merchant city columns.

e. Load the AdaBoost model that is saved in H5 format.

f. Use the predict function to generate final predictions over the test data.

```
tp: 243251.0000 - fp: 8567.0000 - fn: 17883.0000 - tn: 849851.0000
```

Figure 2: Metrics for One Epoch for LSTM Model where "fp" is False Positives

2. Minimize the amount of false positives

The following procedure applies for the testing of false positives for the AdaBoost ensemble learning model from the first design specification:

a. Use the predict function to display the number of false positives over the entire test data.

3. Be accurate

The following procedure applies for the testing of accuracy for the AdaBoost ensemble learning model from the first design specification:

a. Set verbose=1 in the predict function to display the accuracy for the entire test data (see Figure 2).

```
model.evaluate(test_generate, verbose=1)
```

Figure 3: Verbose=1 for Model on Predict Function

4. Processing Speed

The following procedure applies for the speed test on GPU for the AdaBoost ensemble learning model from the first design specification:

a. Before running the predict function, import the Python time library in Google Colab.

b. Find the current time by adding the following line of code before the predict function:

i. *start_time = time.time()*

c. After the predict function add the following lines of code:

i. *end_time = time.time()*

ii. *final_time = end_time - start_time*

The final_time is the processing speed for the Adaboost ensemble model using the GPU (see Figure 3).

```
import time

start_time = time.time()
model.evaluate(test_generate, verbose=1)
end_time = time.time()

final_time = end_time - start_time
```

Figure 4: Processing Time Calculation

Success Criteria

Functional Requirement ID	PASS/FAIL Criteria
1	Classify testing data into either fraudulent or legitimate transactions. At least 1 transaction in each classification.
2	The false positives amount should be between 20-30% of the testing data [2]
3	Achieves an accuracy of 95% on the testing dataset. [3]
4	Be able to process at least 65,000 transactions/second [4]

Table 2. Pass/Fail Criteria

References

[1] “Protection of Personal Information,” *Bank of Canada*. [Online]. Available: https://www.bankofcanada.ca/privacy/protection-of-personal-information/.

[2] F. Wallny, “False Positives in Credit Card Fraud Detection: Measurement and Mitigation,” 2022. [Online]. Available: https://scholarspace.manoa.hawaii.edu/. E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, “A neural network ensemble with feature engineering for improved credit card fraud detection,” IEEE Access, vol. 10, pp. 16400–16407, 2022.

[3] R. Bin Sulaiman, V. Schetinin, and P. Sant, “Review of Machine Learning Approach on credit card fraud detection,” *Human-Centric Intelligent Systems*, vol. 2, no. 1-2, pp. 55–68, 2022.

[4] “Visanet: Global Electronic Payment Network,” *Visa*. [Online]. Available: https://www.visa.ca/en_CA/about-visa/visanet.html.