FRAUDULENT PATTERN RECOGNITION THROUGH ANOMALY DETECTION USING UNSUPERVISED LEARNING

BITBURST ASSESSMENT REPORT

TABLE OF CONTENTS

TABLE OF FIGURES

INTRODUCTION

In this era, one of the easiest ways to generate passive income is through creating automated and repetitive bots which are assigned to do a certain task in high speeds and as much as it could, whether it being a task that would grant you revenue in return, like in our case here, or just being for collecting points and boosting your profile. Both these cases are considered fraud and they would not mean to be for bots to partake in but rather humans who are willingly investing their time in a certain task which they would get something in return, on a note that also humans could be fraudulent and that could be by spamming any key or button in order to reach the finish faster. To distinguish between these fraudulent patterns and normal patterns, we would require a large dataset which is heavily dominated with normal patterns otherwise a lot of fraudulent actions would pass through.

Dataset

The dataset that was handed over was divided into two, one being the main dataset, which was used in this assessment, the other one being a large dataset of all the users. Going back to the main dataset, which was given in a csv format, was called “events.csv”, it contained over 150k datapoints all of which were within a day indicating whether a user opened, screened out or completed the survey. Along side that was the revenue per completion of the survey, and the timestamp of every event.

Jason-to-csv script

This task involved the transformation of the given users dataset, which is “users.json”, into a csv file “users.csv”. One of the obstacles faced was that the JSON file was not correctly coded from the server as the nested data was all stringified, so I had to run the JSON file through a loop and then concatenate it.

The code is found in “json-to-csv.py”

DATA PROCESSING

The first main step of knowing how to process the data is to understand the data and what it represents and how to handle it. This could also include extracting additional features and removing irrelevant features, now this would all depend on the type of work you are willing to do and for what are you going to use that data.

Pre-Processing

As the task was handed, the main goal of this assessment was to figure out any fraudulent usage patterns which were, given as example, high conversion rate and high revenues per complete. The three main datapoints which could be used here are:

- Revenue
- Type (open, close, etc)
- Created_at (Time stamp of each event)

The rest of the data found there could be disregarded since it would not be of any help to detect patterns, for example countries cannot determine whether a user is fraudulent or not since a VPN could be used, another would be which platform a user could use is also irrelevant, so is app_id, uid and network_id. Now the original data is, of course, not to be tampered with so we create different csv files for different processes.

We can keep the uid and the survey_id, we will get back to these in the future since they would be crucial to determine the fraudulent user and which survey they partook given by the survey_id.

Taking in the first datapoint, which is the revenue, we are going to normalize the values and resize them to a margin of 0-1. We would do the same with the time stamp, but the difference would be that we would disregard the date and only take in the time from 0-24 hours including the minutes and seconds.

DURATION CALCULATION

Now to make use of the datapoints, Type and Created_at, there was a script which I built that finds out the duration it took for a user to complete a survey. It would find where the user completed their survey, then it would store the user id and the survey id, then it would iterate backwards through the dataset where it would compare the user id against the survey id and would look for the entry “open” under the column “Type” while taking into consideration the user id and survey id. As the event where all these conditions match, the duration would be calculated based where the timestamp of the completed survey would be subtracted by the timestamp of the opened survey, thus generating the duration of the survey.

We then discard the columns “Type” and “Created at” since they would no longer help us in detecting patterns.

The code is found in “Duration-csv.py

MODEL SET-UP

It is agreeable that the variety of the Artificial Neural Networks is ungraspable, due to the heavy researching and improved computational power, it is only a journey to take on with experimenting and discovering different Neural Network patters.

Our Model

The most popular Neural Network that is used for pattern recognition would be the Autoencoders. The reason that this specific model was picked is due to the many successful research about it. Here is the shape of the autoencoder as seen in the figure below.

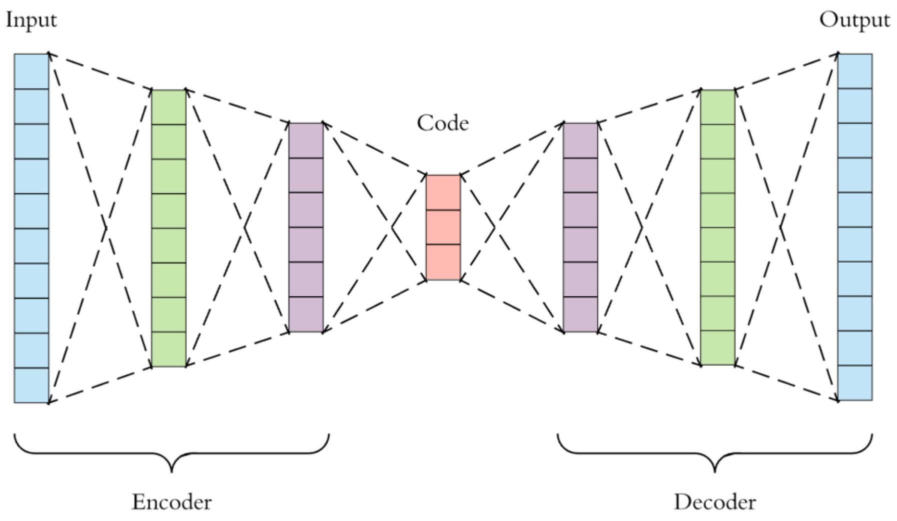

Figure 1: Autoencoder shape

The basic principle of this Neural Network is to break down the given input, in our case it is revenue and duration, and then learn to build it back to the original data. This kind of neural network is capable of other applications such as denoising of images, so when thought about it, anomaly points are datapoints which are out of the ordinary, in other words abnormal. This is the same case when it comes to noise in images.

Training Our Model

We would start by splitting the data into 80%-20%, using the 20% for testing which would be post training stage (testing stage), and 80% of the data would be for training the model.

Since the dataset we are using is heavily dominated by normal patterns, the autoencoder will pick that pattern up and learn how to build a normal pattern, with the very few exceptions of the patterns being abnormal but that will not affect the model.

The code is found in “Train.py”

Testing Our Model

Using the 20% that we saved for trying out whether our model works properly and would detect the anomalies correctly, we would run that dataset through out model. Normal patterned data would receive low reconstruction error since our model is trained to reconstruct normal patterned events. While fraudulent patterns would receive high reconstruction error because our model is not used to receiving such patterns thus it will not be able to correctly reconstruct it. In the figure below, it is clear which datapoints have high reconstruction error and which do not.

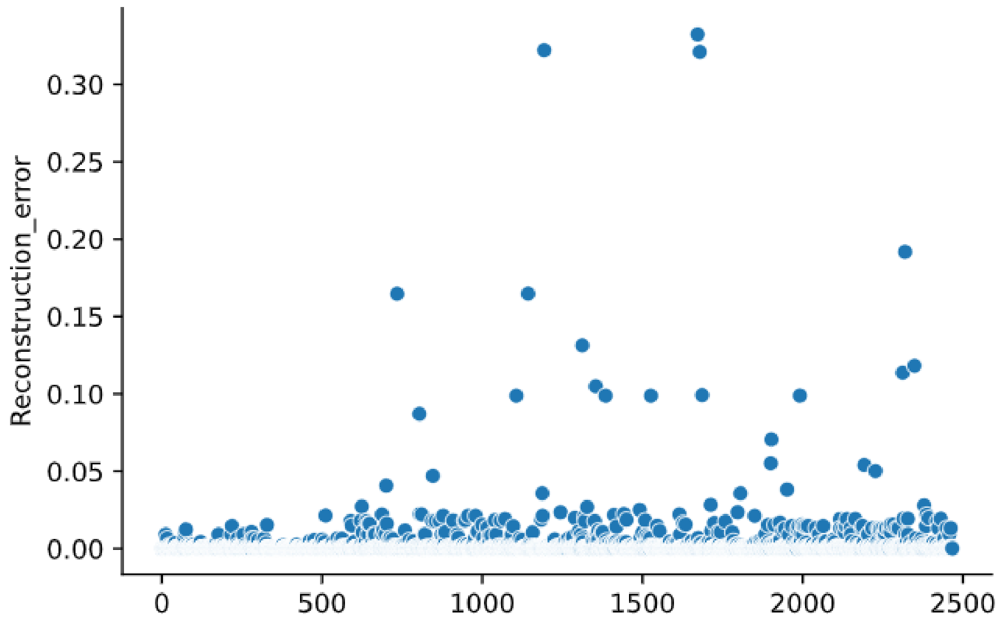

Figure 2: Test data reconstruction error graph

Now we would require a given threshold to determine the top limit of a reconstruction error. This threshold has been calculated using the following equation:

$$Threshold = Median + 1.5 \times IQR$$

The IQR being the interquartile range, both the median and the interquartile range is used from the mean average error of the training dataset since we are going to be basing all our test, including this one, of the training dataset.

This would set the threshold to be a value of ***0.03667…***

We can see in the figure below where the threshold draws a line to split the datapoints between anomaly points, over the threshold, and normal points, which are going to be under the threshold.

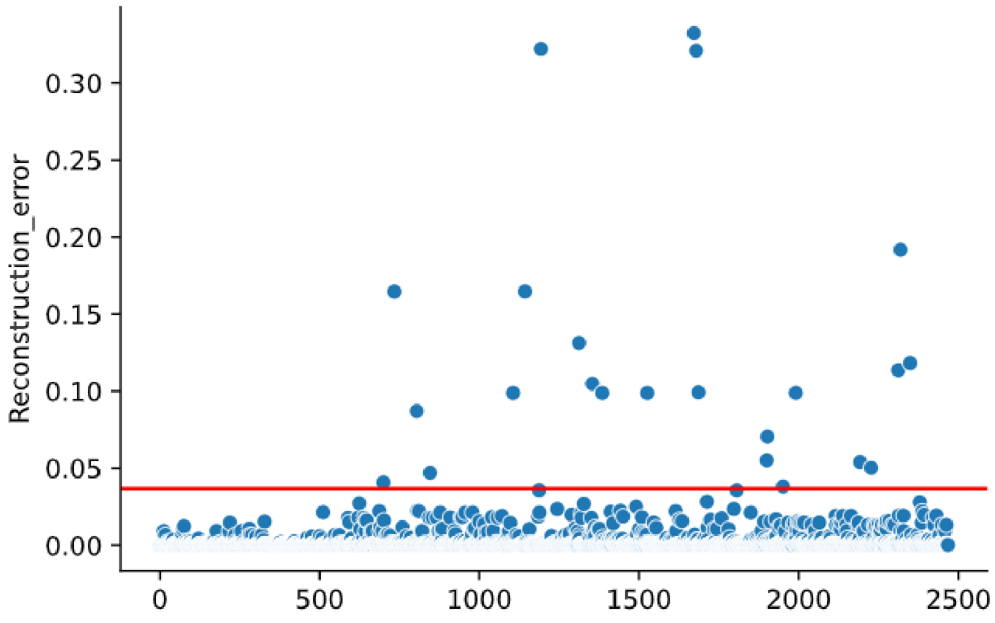

Figure 3: Test data reconstruction error graph with threshold

Now that we have distinguished the anomaly points from the normal ones, we would store their indices in a Dataframe. We would go back to our test data and extract the uid and the survey id of these fraudulent activities based on the indices of the anomaly points and store them in a csv file “Anomaly_data.csv”.

The code is found in “Test.py”

CONCLUSION

It would be impossible to measure the accuracy of the model since the data is unlabeled, but we got a reasonable result, being 23 anomaly points in 2468 entries. These fraudulent activities could be put under question whether to be confirmed as fraud or not fraud and have their account suspended if they are confirmed to be fraudulent by an audit.

Future enhancements

There could be a feature to be added so that a user would time out if they would take longer than the expected time for a survey as they might be considered fraudulent due to the abnormal pattern of having a very long duration.