SUPERVISED CLASSIFICATION OF IMBALANCED BIDDING FRAUD DATA

A Thesis
Submitted to the Faculty of Graduate Studies and Research
In Partial Fulfillment of the Requirements
For the Degree of

Master of Science
in
Computer Science
University of Regina

By

Farzana Anowar

Regina, Saskatchewan
December, 2018

Copyright 2018: Farzana Anowar
Farzana Anowar, candidate for the degree of Master of Science in Computer Science, has presented a thesis titled, *Supervised Classification of Imbalanced Bidding Fraud Data*, in an oral examination held on November 30, 2018. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:  Dr. Irfan S. Al-Anbagi,  
Faculty of Engineering & Applied Science

Co-Supervisor:  Dr. Samira Sadaoui, Department of Computer Science

Co-Supervisor:  Dr. Malek Mouhoub, Department of Computer Science

Committee Member:  Dr. Habib Louafi, Department of Computer Science

Chair of Defense:  Dr. Laurie Carlson-Berg, Faculty of Education
ABSTRACT

Online auctions have become one of the most convenient ways to commit fraud since a huge amount of money is being invested everyday by thousands of customers. Hence, online auctions are vulnerable to several types of fraud. Shill Bidding (SB), a predominant auction fraud, is the toughest to identify due to its resemblance to the normal bidding behavior. An innocent bidder can easily be cheated and lose money if shill bidders are not identified. Our goal is to devise a SB classification model, which is able to efficiently differentiate between legitimate bidders and shill bidders. For this thesis, we employ a real SB training dataset, which is unlabeled. First, we label the SB dataset by the help of hierarchical clustering with an optimal number of clusters, combining it with a semi-automated labeling approach. Second, we assess and compare several advanced over-sampling (SMOTE), under-sampling (NearMiss and ClusterCentroid) and hybrid sampling (SMOTE-ENN and SMOTE-TomekLink) methods to solve the imbalanced learning problem. We utilize the Randomized Search Cross Validation to tune the hyper-parameters for Support Vector Machine (SVM), Random Forest and default parameters for Artificial Neural Network (ANN) with MLP classifier, finally, to obtain the optimal classifier for our SB dataset. Therefore, we develop eighteen fraud classifiers, including fifteen classifiers using sampling techniques and three classifiers without sampling. We demonstrate that the optimal SB classifier exhibits very satisfactory testing performance for detecting and misclassifying shill bidders. The hybrid sampling method SMOTE-ENN combined with the SVM has turned out to be the most performing classifier. Also, every sampled SB dataset greatly improved the
classification performance of all the fraud classifiers than the imbalanced SB dataset.
ACKNOWLEDGEMENT

All thanks to the Almighty to bless me with all the opportunities I have today.

I would like to express my earnest gratitude towards my supervisors Dr. Samira Sadaoui and Dr. Malek Mouhoub for providing me an opportunity to work under their guidance as a Research Assistant. I would like to address especially Dr. Samira Sadaoui again for her immense support and constant encouragement for my thesis. It would not be possible to complete my MSc thesis successfully without her invaluable guidance. I thank her for the grants from Natural Sciences and Engineering Research Council (NSERC) of Canada which assisted me to achieve my goals.

I deeply acknowledge all the financial assistance in the form of TA-ships and academic support I have received from the Department of Computer Science and the Faculty of Graduate Sciences and Research.

I specially thank Dr. Habib Louafi, my thesis committee member to read my thesis and for giving his invaluable opinion for the betterment of my thesis.

I am thankful to God for blessing me with the most supportive husband and in laws. I thank Md Shamsul Alam for being the most understanding and motivating husband. Last but not the least, who I am now, it has been possible only because of the continuous encouragement and support from my parents; Md Anowar Hossain and Kamrun Nahar Begum, and my younger sisters; Fahmida Anowar and Fatema Tuz Zohra.
# TABLE OF CONTENTS

ABSTRACT ................................................................................................................................. II
ACKNOWLEDGEMENT ................................................................................................................. IV
LIST OF TABLES .......................................................................................................................... VII
LIST OF FIGURES ......................................................................................................................... VIII

INTRODUCTION .......................................................................................................................... 1
  1.1 PROBLEM STATEMENT AND MOTIVATION ................................................................. 1
  1.2 RESEARCH CONTRIBUTION ......................................................................................... 3
  1.3 THESIS ORGANIZATION ............................................................................................... 5

RELATED WORK ......................................................................................................................... 8
  2.1 SHILL BIDDING DETECTION USING MACHINE LEARNING ....................................... 8
  2.2 SHILL BIDDING DETECTION USING OTHER TECHNIQUES ................................... 11

BACKGROUND STUDIES ........................................................................................................... 14
  3.1 HIERARCHICAL CLUSTERING TECHNIQUE ................................................................. 14
  3.2 SAMPLING TECHNIQUES ............................................................................................ 16
  3.3 RANDOMIZED SEARCH CROSS VALIDATION ......................................................... 18
  3.4 CLASSIFICATION MODELS .......................................................................................... 21
    3.4.1 SVM ......................................................................................................................... 22
    3.4.2 Random Forest ........................................................................................................ 23
  3.4.3 MLP Classifier-Based Artificial Neural Network .................................................... 23

A SUPERVISED CLASSIFICATION MODEL OF SHILL BIDDING ........................................... 25
  4.1 SHILL BIDDING STRATEGIES AND CHARACTERISTICS ........................................ 26
  4.2 SUPERVISED CLASSIFICATION OF SHILL BIDDING ............................................. 29
  4.3 ACTUAL SHILL BIDDING TRAINING DATA ............................................................. 30
  4.4 CONCLUSION ................................................................................................................ 32

LABELING SHILL BIDDING DATA VIA CLUSTERING ............................................................ 33
  5.1 COMPARISON OF CLUSTERING TECHNIQUES ......................................................... 33
  5.2 SIMILARITY MEASUREMENT FOR HIERARCHICAL CLUSTERING ............................ 35
  5.3 OPTIMAL NUMBER OF CLUSTERS .............................................................................. 38
  5.4 CLUSTER CONTENT ....................................................................................................... 40
  5.5 LABELING OF SHILL BIDDING DATA ......................................................................... 42
  5.6 CONCLUSION ................................................................................................................ 47

SAMPLING OF SHILL BIDDING DATA .................................................................................... 49
  6.1 UNDER-SAMPLING TECHNIQUE .................................................................................. 50
  6.2 OVER-SAMPLING TECHNIQUE .................................................................................... 51
  6.3 HYBRID-SAMPLING TECHNIQUE ................................................................................. 53
  6.4 CONCLUSION ................................................................................................................ 54

EVALUATION OF SHILL BIDDING CLASSIFIERS ............................................................... 55
  7.1 HYPERPARAMETERS TUNING .................................................................................... 55
LIST OF TABLES

Table 1. Different linkage analysis for HC ................................................................. 16
Table 2. Characteristics of SB patterns ........................................................................ 26
Table 3. Preprocessed auction dataset of ‘iPhone 7’ ......................................................... 31
Table 4. Statistics of two individual auctions .................................................................. 31
Table 5. Comparison between different clustering algorithms ........................................... 34
Table 6. Statistics of SB clusters ..................................................................................... 41
Table 7. Data analysis and labeling .................................................................................. 43
Table 8. SB training dataset ............................................................................................ 47
Table 9. Analysis of generated clusters .......................................................................... 47
Table 10. Sampling of Shill Bidding Dataset .................................................................. 54
Table 11. Hyperparameters used in SVM and Random Forest ......................................... 56
Table 12. Parameters used in ANN with MLP classifier ................................................... 57
Table 13. SVM Performance with sampled SB dataset .................................................... 59
Table 14. Random Forest performance with sampled SB dataset .................................... 60
Table 15. MLP-based ANN performance with sampled SB dataset ................................ 61
Table 16. Comparison between best SB Classifiers using sampling techniques ............... 62
Table 17. Classification with the original imbalanced dataset ......................................... 64
Table 18. Classification results with and without sampling technique ............................. 64
Table 19. Best classification results with and without sampling ..................................... 67
LIST OF FIGURES

Figure 1. Supervised classification of Shill Bidding fraud ........................................30
Figure 2. HC dendrogram with Average linkage .........................................................36
Figure 3. HC dendrogram with Single linkage ...........................................................36
Figure 4. HC dendrogram with Centroid linkage .......................................................37
Figure 5. HC dendrogram with Complete linkage ......................................................37
Figure 6. Optimal number of clusters using Silhouette Index method .......................39
Figure 7. Optimal number of clusters using Gap Statistic method .............................40
Figure 8. SB dataset partitioned into 10 clusters .......................................................41
Figure 9. Performance evaluation using the sampling techniques for Recall ..............63
Figure 10. Performance evaluation using the sampling techniques for AUROC ..........63
Figure 11. Comparision of Recall and AUROC for balanced and imbalanced SB dataset ..........66
Chapter 1

Introduction

Online auction is a service that enables persons to participate in auctions with other potential buyers to purchase any product or service from the seller. Unlike traditional auction system, this online auction poses no difficulty for the buyer or seller as they can contact each other instantly. Additionally, the logistical and physical limitations of usual auctions, e.g.; small target audience, physical proximity, time constraint are removed. For these reasons, online auction is gradually gaining more popularity day by day. This thesis specifically focuses on eBay as an example of online auction.

1.1 Problem Statement and Motivation

In the past twenty years, we have witnessed the rapid increase of online auctions in numerous domains, such as antique, vehicle and housing trading. Since 2002, several commercial auction sites, such as 'Yahoo! Auctions' and eBay, gained a huge popularity [1], [2]. Nevertheless, as reported by the Internet Crime Complaint Center of the Federal bureau of investigation, auction fraud is top ranked cyber-crime [3, 4]. In fact, e-auctions are vulnerable to three types of fraud: pre-auction (e.g., misrepresentation of products), post-auction (like non-delivery of products), and in-auction/bidding (e.g., shill bidding and bid shielding) fraud. We concentrate our research on Shill Bidding (SB) fraud.
because it leaves no obvious evidence unlike the two other types of fraud (such as the misrepresentation and non-delivery of products). SB is the most common in-auction frauds but the most difficult to identify because of its resemblance to the normal bidding behaviors [5]. To maximize the seller's revenue, a shill bidder always tries to increase the price by submitting many bids of a product using multiple fake accounts. In the setting of expensive products, SB will result into substantial financial losses for the auction winners. In [6], it is mentioned that excessive SB can lead to a market failure. Moreover, the presence of shills will demotivate honest users from participating, which may affect the auctioning business. Over the years, numerous sellers and their collaborators have been prosecuted due to SB activities; here are some examples:

- In 2001, three sellers were charged with SB fraud. They participated in over 1,100 art auctions on eBay using more than 40 fake accounts. In total, the colluding sellers gained a profit of $300,000 [7].

- In 2004, owner and employees of the ‘New Windsor Auction Gallery’ pleaded guilty to SB fraud, which had been conducted through numerous auctions on eBay. The owner was required to pay $50,000 in restitution and fines in addition to facing four years in prison [8].

- In 2012, the auction site ‘Trade Me’ was forced to make a payment of $70,000 to each victim of SB fraud committed by a motor vehicle dealer based in Auckland. The fraud, which went undetected for one year, resulted in substantial losses for the victims. After the fraud was detected, Trade Me banned the accused trader and referred the case to the Commerce Commission for a thorough investigation [9].
In 2015, according to the New York Post, a lawsuit was filed in the Manhattan federal court against Wall Street banks, stating that they had influenced the $14 trillion U.S. Treasury bond market and secretly conspired with each other before the auctions to maximize the profit of the federal debt [10].

These cases highlight the immense importance of detecting fraudulent bidding activities, as, such activities will certainly cause significant financial losses for legitimate buyers if left undetected. Several empirical studies demonstrated the presence of SB activities in different commercial auction sites [11, 16]. The common fraud prevention techniques, like user authentication, fraud awareness and feedback ratings, are reliable for small-scale fraud applications, but they do not provide any protection against SB. Consequently, it has become crucial to analyze the behavior of bidders and sellers in order to devise a system that can detect SB and prevent honest buyers from becoming victims. Robust SB detection systems can be obtained by using Machine Learning (ML) techniques.

1.2 Research Contribution

Efficient supervised classifiers have been built to control fraud in various sectors, including credit card, telecommunication, electricity, and insurance industries. Yet, there are limited studies in the literature to classify the bidding transactions of users. This limitation is due to several reasons: determining relevant SB patterns, implementing robust SB metrics, scraping and preprocessing commercial auction data, and evaluating the SB metrics against the extracted data. Auction sites provide a tremendous volume of public data, on which researchers can employ ML to effectively examine auction data to determine the trends and behaviors of online users, as well as the popularity of products
on the market. The ultimate goal of our study is to develop an optimal SB classification model. To this end, we train offline several binary classifiers based on a collection of the most relevant SB patterns that have been observed in many infected auctions. Labeling any training dataset is critical for the classification task. Most of the time, this is done manually, which is a tedious and an error prone operation. To facilitate the labeling of our SB data, we utilize hierarchical clustering (HC) to group participants with similar bidding behavior, then direct and statistical testing methods have been to find out the optimal number of clusters. Subsequently, we apply the labeling strategy introduced in [5] to determine whether the bidders in a given cluster are behaving normally or suspiciously. The resulting labeled SB training dataset is however imbalanced and learning from imbalanced data is another challenge for the classifiers. To overcome this problem, we employ five intelligent sampling methods (SMOTE [26], SMOTE-ENN [44], SMOTE-TomekLink [44, 45], NearMiss [44] and ClusterCentroids [25]) to rebalance our training dataset; otherwise, fraudulent bidders will be misclassified. As a result, we obtain five balanced SB datasets with different sizes. So far, no study has ever used these extensive sampling techniques to compare the classification results for any SB dataset except us. Finally, we apply three classification algorithms (SVM [31], Random Forest [32] and ANN with Multi Layer Perceptron (MLP) classifier [36, 37]) to the five sampled datasets as well as to the original imbalanced dataset. Indeed, we develop eighteen supervised fraud classifiers, including fifteen classifiers using sampling techniques and three classifiers without any data sampling. Based on several performance metrics and Random Search Cross Validation, we perform three main experiments. Again, we are the first to use the Randomized Search CV to obtain the optimal
classification results for SB dataset using different optimal hyperparameters. In the first experiment, we search for the most performing SB classifier by comparing the results of the five sampling methods combined with the three classifiers. In the second experiment, we compare the three classifiers when applied to the imbalanced SB dataset. In the last experiment, we compare the performance between balanced and imbalanced SB data.

For the benefit of users, our SB classification service can be deployed in any auction site. Existing SB classification solutions are mostly offline, but a fraud-detection should be capable of functioning in the real-life scenarios as shown in Chapter 4. In past offline SB studies, a training instance denotes the behavior of a bidder in all the participated auctions. As a consequence, we cannot know which specific auction is infected by fraud. In our work, an instance depicts the conduct of a bidder in a certain auction. Another issue is that SB cannot be prevented from occurring. Therefore, just at the ending of bidding time of a monitored auction, for each participant, we measure the bidder per auction-wise SB patterns. Then, we launch the SB classifier on the generated SB data to detect malicious activities. If the auction is infected by fraud, the administrator will hold off the payment of goods/services until the investigation is completed. After the investigation, if a bidder is proved as a fraud some actions will be taken such as the infected specific auction will be cancelled and the account for the fraud bidder will be blocked forever.

1.3 Thesis Organization

The rest of the thesis is organized as follows:
In Chapter 2, we discuss several related works with respect to fraud detection in two categories: SB detection using ML and SB detection using other techniques.

In chapter 3, we discuss the background studies of HC, Sampling techniques, Randomized Search Cross Validation to tune the optimal parameters and lastly discuss the reasons behind choosing the SVM, Random Forest and MLP based ANN classifier,

In Chapter 4, an overview of the proposed framework is discussed. We explain SB strategies and characteristics and show how the actual SB training dataset has been developed.

In Chapter 5, we review different clustering techniques to find out the best clustering technique and show how we have applied this technique to SB training dataset. Then we discuss Silhouette Index and Gap Statistic methods to obtain the optimal number of clusters for the SB dataset. Then, we demonstrate how the semi-automated labeling approach has been applied to label the optimal clusters of bidders.

In Chapter 6, we focus on the issue of having imbalanced training dataset and discuss about five different sampling techniques including over-sampling (SMOTE), under-sampling (NearMiss, ClusterCentroids) and hybrid sampling (SMOTE-ENN, SMOTE-TomekLink). Then we present the training dataset after applying all these sampling techniques.

In Chapter 7, we discuss different hyperparameters to attain the optimal parameters for SVM and Random Forest classifiers. Also we discuss the different performance metrics in terms of detection and misclassification rate. Then to compare the efficiency, we
develop SVM, Random Forest and ANN with MLP classifier-based fraud classifiers. Then, we compare and discuss the classification results for the original imbalanced training dataset and we show a comparison, using different performance evaluation metrics, between the best classifiers for each of three classifiers with and without using sampling techniques.

In Chapter 8, we conclude the thesis and finally present some promising future works.
Chapter 2

Related Work

In the last decade, multiple useful research associated to auction fraud detection have been carried out because of the huge market demand. Research on SB detection can be categorized into different categories based on their respective approaches, such as statistical [12], concurrent [13], machine learning [14]. In this thesis, we focus on two approaches for recent researches: SB detection using Machine Learning Techniques and SB detection using other techniques such as agent-based system.

2.1 Shill Bidding Detection Using Machine Learning

In [15], the authors proposed a semi-automated approach combining ‘one-class SVM’ and ‘Decision Tree’ to identify shill bidders. Based on the one-class SVM learning, first this approach finds all the outliers that are basically shill bidders. Then, these outliers are sent to the Decision Tree classifier for further analysis because some innocent bidders may have been classified wrongly as outliers. In fact, the authors manually checked all the classified data obtained in each node of the Decision Tree. As a result, they altered the labels of several nodes, the classification of which is erroneous. Finally, this updated decision tree is considered as the optimal model in detecting shill bidders. To carry out the experiments, the authors collected information of 59,949 bidders and 67,244 products.
from a Chinese website named ‘Wowma’. Nevertheless, to evaluate the outliers, the authors took into account only two attributes; feedback rating and bidding history, which do not provide enough information for detecting shill bidders. In addition, they did not present any preprocessing and training steps, such as data normalization, clustering and labeling.

In [16], the authors introduced a SB classifier that is able to adapt to new auction data based on an incremental feedforward backpropagation artificial neural network (ANN). First, the authors retrieved from eBay auctions of the product ‘Used Playstation 3’. Then, they defined a hierarchical clustering technique to partition the bidding data. After that, they manually labeled each cluster as either suspicious or normal based on the values of ten SB attributes, such as ‘Average Outbid Time’, ‘Elapsed Time before First Bid’ and ‘Seller Feedback Rating’. Then, the labeled dataset is used to initialize the fraud classifier. When a bidder is predicted as normal, it is sent to the classifier for an incremental training, and when a bidder is predicted as suspicious, it is first investigated using additional evidences, and then fed to the classifier. The model ANN sometimes suffers from local minima and it is computationally expensive since it has multiple parameters to tune at once [17]. In addition, ANN does not explain the behavior of the network, which ultimately reduces its trust [17]. In [16], bidders are classified according to their participation in all the auctions. In this case, it is not possible to determine which auctions are infected by fraud. Consequently, money loss cannot be stopped.

The authors of [18] implemented a synthetic data generator to build a classification model to detect ‘competitive shilling’ fraud. The synthetic data generator uses an agent-based simulator to generate data based on several fraudulent behaviors. The authors
followed three steps to produce the training data. First, an agent is defined with a certain fraud type, second, the defined agent generates synthetic data, and finally, the generated data is transformed into a series of bids and auctions based on ten user-defined attributes. For the performance evaluation, two synthetic datasets have been produced. For each dataset, the normal and fraudulent bidder ratio was 180:1 approximately. To address the highly imbalanced datasets, the authors used only one technique, the random under-sampling method. Then, a Decision Tree with 10-fold Cross Validation was applied to the two under-sampled datasets. Though authors showed the evaluation of their model with commercial dataset as well. However, they didn’t mention anything about their commercial dataset and the SB features. Synthetic data are not preferred in fraud detection because it does not represent actual auction data.

In [19], the authors developed a general approach to detect in-auction fraud by using the Hidden Markov Model (HMM). In this approach, only two parameters, the number of bids and bid values, have been used for the sake of simplicity. In the registration phase, the authors proposed to restrict the user to create multiple accounts with a legal parameter (credit card information). The general Markov model is organized into two-layered (training and detection) architecture. The authors used K-means clustering to obtain the initial probability set to understand the bidding behavior for the authentication purpose. Then, the detection layer is used to find in-auction fraud, and users are then categorized into three fraud categories (high, medium and low) depending on a behavioral approach that analyzes the bidding habits. Finally, HMM is applied to these categories to detect a bidder as fraud by tracking his bidding behavior. In this study, two parameters are not sufficient to detect in-auction fraud. Moreover, the type of bidding fraud has not been
specified, and no experiments have been conducted for validating the proposed approach.

Very recently, an SVM-based SB detection model has been proposed in [5] which has overcome the issues of ANN and HMM. This classifier allows auction companies to differentiate between normal and shill bidders. The class imbalance is a serious fact in fraud detection as it deteriorates the classification performance and it may also misclassify the minority class (suspicious bidders). The authors have applied three data sampling methods (SMOTE, SpreadSubSample and hybrid of both sampling) to solve the imbalanced learning problem. However, the SB dataset has been built from a small number of auctions by including synthetic data for two missing auction attributes. In their ‘Palm Pilot PDA Dataset’, the authors considered only 149 auctions and 1024 SB instances. In our study, the auction dataset contains 807 auctions of 'iPhone 7', and their corresponding 6321 SB instances.

2.2 Shill Bidding Detection Using Other Techniques

Authors in [12], illustrated a Shill Score (SS) reputation system among the offline approaches to find out the level of SB in online ascending auctions that is finished at a pre-determined time. A series of auctions by a specific seller are considered here for determining bidding patterns and calculating a shill score for each bidder. The Shill Score indicates how a user is likely to be involved in SB. These shill scores can be observed by other bidders and allows them to make an informed decision about whether to participate in that bid or not. For the experiment purpose authors used data from commercial online auctioneer such as eBay. The shortcoming of this particular method is that it does not work in real time auctions.
Then authors of [20] made improvements in later time with a proposal, in which detection of SB where multiple fraudulent users take part in fraudulent activities as a group. A scored system called ‘Collusion Score’ was proposed by the authors which is used to detect a Shill bidding collusion group. The Collusion Score is the combination of every Shill Score rating that labels each and every user with a score depending on the potential of that user’s engagement in collusive bidding activity. However, this method only works at the end of the auction.

In [21], a generic framework is proposed which covers the real-time monitoring of multiple live auctions. Progressing auctions are monitored by this framework to deter shill bidders from succeeding in fraudulent activities by taking actions in real-time. This monitoring process is done in different phases of the auctions based on real-time events. According to the authors, in comparison with the offline process, the real-time monitoring systems investigate a lesser amount of bids, which in turn leads to faster processing. Additionally, it is possible for the system to address a possible shill bidder during a progressing auction and to cut these fraudulent bidders down from the system before legitimate bidders become affected by shill bidding. This proposed real-time monitoring framework is partitioned into three levels (identify SB, react to SB and update the clusters of bidders) of functionality. In the first step, the bidding activities are monitored through the fraud detection mechanisms in run time. Then, reaction to suspicious activities is determined and this is done by adopting real-time steps for the suspicious auctions. In the final step, SS and clusters of bidders are updated after the ending of every auction dynamically. However, this proposed method takes place after the submission of every single bid and each of these bids are assessed against the similar
set of bidding attributes which is time consuming.

Later this study has been improved in [12]. To monitor several live auctions for In-Auction Fraud (IAF), authors introduced a stage-based framework. This framework systematically monitors each live auction in different periods of times, which are dependent on duration of the auctions and types of IAF. This framework identifies IAF initially by observing stage activities for each bidder’s depending on the set of IAF patterns at each time point. After that, proper actions are taken to respond to the shill bidders. A dynamic agent architecture has been employed to delete or create the monitoring agents for this framework and fraud scores for the bidders are updated automatically after an auction is ended. Authors also used the real auction data from the commercial site (eBay) for the validation purpose of their proposed IAF monitoring service. In both papers [21, 12], the authors considered eight bidding behaviors to detect SB in real-time. Though, the authors did not explain a cause behind choosing eight bidding behaviors out of seventeen proposed suspicious bidding patterns.
Chapter 3

Background Studies

In this chapter, we explain the background studies of all the techniques we have used for this research. First, we discuss about clustering techniques, specifically the hierarchical clustering, then, we show sampling techniques, then we explain the Randomized Search Cross Validation, and finally discuss three classifiers: SVM, Random Forest and MLP based ANN.

3.1 Hierarchical Clustering Technique

Clustering is one of the most vital steps as a training step in ML. It is an unsupervised learning and common technique for observing and analyzing data [22]. Clustering technique categorizes data points into several groups in such a way that the data which are clustered in similar group are alike based on some precise properties [23]. Hence, a cluster is a collection of data having ‘similarity’ within a cluster and ‘dissimilarity’ to the data of other clusters [23]. Decent clustering classifies clusters by maximizing the distance between two separate clusters and minimizing the distance of data within a cluster [24].

Hierarchical clustering (HC) generates a hierarchy of given data and the final output of HC is known as ‘dendrogram’, which represents the hierarchical relationship between the
clusters [24]. HC starts by treating each data as one single cluster. Then, it continuously performs the following steps: (1) detect two clusters that are closest to each other, (2) combine the two most similar clusters, and this process lasts until all the clusters are combined together. HC is divided into two types: 1) Agglomerative Hierarchical Clustering that combines all the similar clusters sequentially, 2) Divisive Hierarchical Clustering that splits the clusters successively. Since Agglomerative clustering can create an ordering of the data, which is informative for data visualization, it is widely used compared to the Divisive clustering. We also chose to work with the Agglomerative clustering to create the dendrogram for SB dataset.

For generating the dendrogram, the parameters we used are shown in the following code segment.

```python
dendrogram = sch.dendrogram(sch.linkage(X, method = 'complete'))
```

Where,

- X is SB dataset
- Method is linkage function for the similarity measurement

For calculating similarity measurement, there are four methods named: Single, Complete, Average, and Centroid linkage. They are explained in Table 1.
### Table 1. Different linkage analysis for HC

<table>
<thead>
<tr>
<th>Similarity Measurement</th>
<th>Definition</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Linkage</td>
<td>Distance between the two nearest data in two separate clusters.</td>
<td>$\min_i D(x_i, y_j)$</td>
</tr>
<tr>
<td>Complete Linkage</td>
<td>Distance between two data that are far off from each other in two clusters.</td>
<td>$\max_i D(x_i, y_j)$</td>
</tr>
<tr>
<td>Average Linkage</td>
<td>Distance between all pairs and the average of all these distances in two clusters.</td>
<td>$\frac{1}{k}\sum_{i=1}^{k} \sum_{j=1}^{l} D(x_i, y_j)$</td>
</tr>
<tr>
<td>Centroid Linkage</td>
<td>This is mean a vector location of every cluster and takes into consideration the distance between centroids of clusters.</td>
<td>$D(\bar{x}, \bar{y})$</td>
</tr>
</tbody>
</table>

### 3.2 Sampling Techniques

Imbalanced data problem is surprisingly one of the common problems in ML, particularly in classification, occurring with an inconsistent ratio of data in each class in a dataset. Several classification problems, such as fraud detection, software quality prediction, spam filtering, and text categorization possess only 1% or less of the total of instances of the class of interest. When standard classification algorithms are applied to imbalanced training data, they may have a bias towards the majority class [5]. Classifiers that learn to always predict the majority class may obtain 99% accuracy, but such classifiers are not useful for identifying the minority class because they tend to return poor accuracy. In fraud applications, the situation is more unfortunate as it is the minority (suspicious) class
that is the most important to detect since it carries the highest cost of misclassification. Additionally, a screwed class distribution always lowers the predictive performance as shown in [42].

In this thesis, the SB dataset that we have worked with is not different. That is, we have obtained 5694 bidders as ‘normal’ bidders and only 627 bidders as ‘suspicious’ bidders (see table 9). Hence, the imbalance ratio we obtained is 9:1 which means the SB dataset is moderately imbalanced. To solve this issue, we utilized data sampling techniques on SB dataset. Sampling is a technique used to balance the imbalanced datasets, either by under-sampling the majority class data or over-sampling the minority class data. Sampling allows us to generate a balanced dataset that does not make the classifiers biased towards the majority class. Data sampling techniques can be categorized into three categories: 1) under-sampling, 2) over-sampling and 3) hybrid sampling (combination of under and over-sampling). A brief discussion on these sampling techniques is given in chapter 6.

However, sampling methods have flaws in practice. Under-sampling the majority class can sometimes exclude important instances. ClusterCentroids under-sampling has been used to solve this problem [25]. Hence, we have used ClusterCentroids method for our experiments, along with NearMiss under-sampling method. Similarly, over-sampling the minority can lead to over-fitting the model, since it launches duplicate instances. Therefore, we used the SMOTE over-sampling method that overcomes the over-fitting issue by creating new instances instead of duplicating the existing ones in the minority class [26]. For hybrid sampling, we used SMOTE-ENN and SMOTE-TomekLink to have the advantages of both under and over sampling. Indeed, in recent studies, these two
techniques have been used by other researchers [31], [49].

While implementing these sampling techniques, we imported all the required modules from Sci-kit Learn. For example, the following code shows how SMOTE module can be imported from Sci-kit Learn [50].

```python
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=0)
X_resampled, y_resampled = smote.fit_sample(X, y)
print('Resampled dataset for SMOTE {}'.format(Counter(y_resampled)))
```

Also, while working with the imbalanced dataset, another challenge is to choose the appropriate evaluation metrics when comparing approaches applied to imbalanced classification problems. It is suggested to use metrics beyond accuracy i.e. recall, precision, and AUROC [27]. In our work, we used these metrics for our experiments in addition to three other metrics (F-Measure, False Negative Rate, and False Positive Rate) to evaluate the performances. Then, we used Log-loss Function to evaluate the developed classification models.

### 3.3 Randomized Search Cross Validation

Another major challenge is to tune the best hyperparameters for the classification model. Hyperparameters are settings which can be tuned to manage the behavior of a ML algorithm. We have used ‘Randomized Search Cross Validation (CV)’, one of the mostly used methods to build the optimal fraud classifier for each of the five sampled training datasets by selecting the best kernel function. The reasons we have chosen Randomized Search CV over Grid Search CV are; 1) Grid Search CV is computationally expensive,
especially if we are searching over a large hyperparameter space and dealing with multiple hyperparameters [28]. 2) With Grid Search, we specify the parameters to try for different combinations and Sci-kit Learn then tries each possible combination which is not efficient all the time especially for the fraud detection domain. Randomized search, on the other hand, takes some distributions to sample from and a maximum number of iterations to try [28]. This allows us to focus the search on areas where the parameters should perform better using stratified k-fold CV.

We split the input dataset in an 80/20 percent ratio between training set and validation set respectively for Randomized Search CV. Randomized Search CV fits the classifier model and tunes the hyperparameters on the training dataset. This 20% validation set guides in the selection of hyperparameters. Technically, the validation set is used to ‘train’ the hyperparameters prior to optimization. Then, it provides the classification results based on the validation dataset using the tuned hyperparameters.

In the following code, we show how to split dataset into training and validation sub-datasets.

```python
# Split the dataset in two parts
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

There are two hyperparameters to tune for SVM classifier model which are: Cost (C) and Gamma (ɤ) [29]. For C and ɤ parameters, this is important to provide a distribution which is continuous for having the complete benefit of randomization. Hence, it will increase the number of iterations which eventually lead to an advanced search. We have used number of iterations (n_iter) = 10, which is the default iteration number in scikit-learn.
Below, we show the parameter range we have used for the SVM classifier.

```python
from sklearn.svm import SVC
import scipy
params = {'C': scipy.stats.uniform(100, 500), 'gamma': scipy.stats.uniform(0.1, 10),
          'kernel': ['rbf'], 'class_weight': ['balanced', None]}
```

We have set the range of C for SVM classification to [100, 500] and of γ to [0.1, 10]. After applying the Randomized Search CV, we obtained C=266.09 γ=1.59 as the tuned optimal hyperparameters with the highest performance metrics for SMOTE-ENN with 5-fold CV.

For Random Forest, we have chosen three hyperparameters for tuning to obtain the best performance. They are: n_estimators, max_features and max_depth.

In the following code, we show the parameter range we have used for the Random Forest classifier.

```python
rfc = RandomForestClassifier()
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 500)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt', 'log2']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(start = 10, stop = 110)]
```

The range we have chosen for n_estimators, max_features and max_depth is [100, 500], [auto, sqrt, log2] and [10, 110], respectively. Randomized Search CV provided n_estimators= 467, max_features= ‘auto’, and max_depth= 48 as the optimal hyperparameter for Random Forest Classifier with SMOTE-ENN 5-fold CV.

Though for MLP based ANN, we did not use Randomized Search CV. To obtain a model
using the ANN is a hard task and scikit-learn makes it easier by providing estimator objects. We import Multi-Layer Perceptron Classifier (estimator) from the Scikit-learn Neural Network library. Then, we create an instance of the model by defining the size of the hidden layer. For this parameter, we pass in a tuple consists of the number of neurons in Sci-kit Learn. For the sake of simplicity, we selected three layers having the same number of neurons with a default number of iterations of 200 in this case. ANN with MLP classifier has a regularization term that can further add the loss function to prevent the model from over-fitting. We show the different parameters we have used for the ANN with MLP classifier in the following code segment.

```python
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(30,30,30))
mlp.fit(X_train,y_train)
```

All the parameters described in this section for three classifiers are explained briefly in chapter 7.

### 3.4 Classification Models

We have selected the SVM, Random Forest, and ANN to develop the eighteen SB classifiers. We have used Randomized Search CV to obtain the best performance metric using the tuned hyperparameter for SVM and Random Forest, and Multi-Layer Perceptron (MLP) classifier for the ANN for obtaining the best performance metric.
3.4.1 SVM

SVM is a discriminative classifier that performs classification for two non-overlapping classes by producing a hyperplane that maximizes the margin between each of two classes [30]. Vectors that define the hyperplane are known as ‘support vectors’ [30]. This classifier helps to construct a hyperplane that performs as a ‘decision boundary’ by separating data samples into two classes [30]. SVM tries to obtain the hyperplane that generates maximum distance between two classes. In other words, SVM assists to build a margin by maximizing the distance of the closest data of two separate classes and bigger the margin, better the generalization ability of this classifier is [30]. Also, SVM provides ‘kernel trick’ to make classification efficient when data points are non-linearly separable [30]. Using a kernel function, we can apply dot product between two data points so that every point can be mapped into high dimensional space.

The SVM classifier has been chosen for this research for following important reasons:

- Through empirical studies, it has been shown that combining data sampling with SVM leads to higher classification performance [31]
- Auction data are difficult to linearly separate. This issue can be easily solved with the SVM kernel [16].
- SVM is very efficient with medium sized training datasets like the one we are using in our experiments.
3.4.2 Random Forest

Random forest can be used to assemble predictive models for classification as an ensemble learning algorithm. A forest is composed of several trees. Random forest generates decision trees from randomly selected data and obtains prediction from each tree and picks the best result by means of ‘voting’ [32]. Being an ensembled algorithm, Random Forest tends to produce more precise result [33]. Even though several decision trees are likely to noise, final classification outcome tends to be accurate for this classifier.

The Random Forest classifier has been chosen for following reasons:

- Random Forest is handy and easy to use since its default hyperparameters often provide satisfactory prediction result [34].
- It doesn’t require higher number of hyperparameters, which makes it easy to understand and implement.
- In ML, one of the biggest issues is ‘over-fitting’. However, Random Forest usually does not suffer from this issue. This classifier does not over-fit the model because there are enough trees in the forest [35].

3.4.3 MLP Classifier-Based Artificial Neural Network

MLP is a feedforward artificial neural network and supervised learning algorithm [36]. An MLP comprises three layers of nodes at least. First layer of node is the input layer, intermediate layer is denoted to as a hidden layer and the final layer of nodes is output layer. Every node is a neuron which employs a ‘nonlinear activation function’, excluding
the input nodes. A supervised learning method named ‘back propagation learning’ is used for training [37]. The property of having non-linear activation function and multiple layers makes MLP different from a linear/single perceptron. MLPs are often known as ‘Vanilla Neural Networks (VNN)’, particularly when MLP have only one hidden layer [48].

The advantages of using MLP are [36, 37, 48]:

- It has the capability to distinguish non-linearly separable data.
- MLP can learn model in real-time which is beneficial to us for our online shill bidding phase to detect the suspicious bidder.
Chapter 4

A Supervised Classification Model of Shill Bidding

In this thesis, we propose a fraud detection model that allows online auctioneers to distinguish between normal and suspicious bidders. In our case, these fraudulent bidders are shill bidders. A shill bidder is a malicious user, the seller or his accomplice, who places many bids via phony accounts with the motive of inflating the price of the products being auctioned. SB may cause a significant financial loss for purchasers in case of expensive products as well as products with an unknown value in the market. Our model takes effect just after the bidding period and right before the payment is done in an auction. We can suspend an auction when it is found to be contaminated by SB to prevent financial losses for the auction winners. In fact, the learned classification model is to be deployed after the bidding period for detecting suspicious bidders before processing any payment. Nevertheless, additional investigation is necessary to verify the suspicion. If suspected bidders are verified as actual shills then actions are taken, such as cancelling the infected auction and suspending the shill bidder's account.
4.1 Shill Bidding Strategies and Characteristics

SB is the most dominant in-auction fraud and also the most difficult to detect because it is similar to normal bidding behavior. As exposed in Table 2, we use eight SB strategies [5], [11] to effectively distinguish between fraudulent and legitimate bidding behavior. When the participation is low in the early bidding stage (up to 25% of the duration), shills (a seller and/or accomplices) place bids to encourage other users to join the auction. Most of the fraud happens during the middle bidding stage (25% to 90% of the duration) because it is quite risky to bid in the last stage since a shill might win. Each pattern is unique as it represents one aspect of the bidding behavior that occurs in a certain stage. A pattern can be computed from the characteristics of an auction, or a bid transaction, or a bidder. The uniqueness of SB patterns (as the training features) will help increase the predictive performance of the fraud classifiers. The algorithms to measure the SB patterns have been defined in detail in [6], [11].

Table 2. Characteristics of SB patterns

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Motive</th>
<th>Level</th>
<th>Weight</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auction Starting Price</td>
<td>Seller unusually sets a cheaper starting price than other concurrent auctions</td>
<td>Draws attraction of the people in the auction</td>
<td>Auction</td>
<td>Low</td>
<td>Higher the value, the more suspicious</td>
</tr>
<tr>
<td>Early Bidding</td>
<td>Bidder bids in early hours of an auction</td>
<td>To allure honest bidders to take part in the auction</td>
<td>Bid</td>
<td>Low</td>
<td>Higher the value, the more suspicious</td>
</tr>
<tr>
<td>Bidding Ratio</td>
<td>Bidder participates much more as</td>
<td>To inflate the price and draw higher bids</td>
<td>Bid</td>
<td>Medium</td>
<td>Higher the value, the more suspicious</td>
</tr>
</tbody>
</table>
It is essential to assign weights to the SB patterns to be able to label bidders as fraudulent or non fraudulent. Through scenarios, we show how to weight the eight SB patterns:

- At the early bidding stage, a seller may submit a low starting price for attracting more bidders. The legitimate bidder might participate to get the product with a low price. On the other hand, a dishonest bidder might also bid just to increase the price. From these two scenarios, we can expect these two patterns, ‘Early Bidding’ and ‘Auction Starting Price’, maybe somehow similar to normal bidding. Moreover, there can be bidders who may bid aggressively only to win a
product in an auction without shilling, and usually an auction with shill bidders has more bids than regular auctions. Hence, sometimes ‘Auction Bids’ can be similar for an auction with or without shilling. So, we assign to these three patterns a low weight.

- Successive Outbidding’ reflects a very high probability of a malicious activity since the user outbids himself several times. So, we assign a high weight to this pattern.

- A suspicious and legitimate bidder may have same values sometimes for four patterns ‘Bidding Ratio’, ‘Winning Ratio’ and ‘Last Bidding’. A legitimate bidder may not be interested to buy a product anymore after bidding a certain amount of money. On the contrary, a fraudulent bidder bids with the sole motive to increase the price and when the price is high enough or the bidding is at its last stage, the fraudulent bidder will not bid anymore. Both will be inactive as they don’t want to win. This inactiveness refers to an indication of fraudulent activities. So, we assign to these three patterns a medium weight. A legitimate bidder may participate in few auctions for a specific seller for his desired product only, and a fraudulent bidder tends to participate only for those sellers with whom he has secretly colluded. In this scenario, both bidders may have similar buyer tendency. Hence, we assign a medium weight to ‘Buyer Tendency’ as well.
4.2 Supervised Classification of Shill Bidding

As depicted in Figure 1, we show how to develop an efficient offline SB classifier, and once it is online, how we operate in real-life scenarios. First of all, we need to fetch a large number of completed auctions of a particular product from a commercial auction site. We need to select a popular product whose auctions may have encouraged SB activities. Nevertheless, as shown in [6], the original auction dataset contains noisy data and inappropriate data formatting. Therefore, properly preprocessing the auction dataset is a critical step before measuring the SB patterns.

Next, we evaluate the SB patterns against the extracted auction data, and as a result we generate the training dataset. Now, we need to label the SB data using certain ML technique, such as unsupervised learning (like data clustering), semi-supervised classification or active learning. As any labeled SB dataset is imbalanced, therefore we solve this imbalanced learning problem via data sampling or cost-sensitive learning. Otherwise, the classification may be biased to the majority class, which means in our case the fraudulent bidders will be incorrectly classified as normal. Finally, we look for the optimal SB classification model by developing several efficient fraud classifiers based on multiple performance metrics.

Our model takes effect just after completing the bidding period and before processing the payment, as illustrated in Figure 1. First, the SB patterns will be measured for all the participants of the current auction once all the bid transactions are available. Subsequently, the classification model will be applied to detect suspicious bidders, which are then sent for further investigation.
Figure 1. Supervised classification of Shill Bidding fraud

4.3 Actual Shill Bidding Training Data

In [6], the authors crawled from eBay a good number of auctions of a very hot product ‘iPhone 7’ for the period of March to June 2017. They selected ‘iPhone 7’ for factors that might had increased the chance of SB activities: 1) it attracted a high number of bidders and bids; 2) it has a good price range with the average of $578.64 (US currency). Certainly, more the product price is high, more likelihood of being fraud [38]; 3) it has different bidding durations: 1, 3, 5, 7 and 10 days. All these durations were considered because in long duration, a shill may easily mimic normal behavior, and in short duration, fraudulent sellers may receive positive feedback ratings [38].

The scraped auction dataset has gone through a rigorous preprocessing operation by
removing irrelevant and duplicated attributes as well as records with missing values (like ID of bidders and sellers) and records with inconsistent values, merging several attributes into a single one, converting several attributes into proper formats, and generating IDs for the auctions [6]. Table 3 provides a summary of the preprocessed auction dataset of ‘iPhone 7’. The dataset consists of different types of iPhone 7 (iPhone 7 with 32 GB, iPhone 7 plus with 128GB and iPhone 7 plus with 256 GB). We also give a short summary for two auctions in Table 4.

Table 3. Preprocessed auction dataset of ‘iPhone 7’

<table>
<thead>
<tr>
<th>No. of Completed Auctions</th>
<th>807</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Records</td>
<td>15145</td>
</tr>
<tr>
<td>No. of Bidder IDs</td>
<td>1054</td>
</tr>
<tr>
<td>No. of Seller IDs</td>
<td>647</td>
</tr>
<tr>
<td>Avg. Winning Price</td>
<td>US $ 578.64</td>
</tr>
<tr>
<td>Bidding Durations</td>
<td>1, 3, 5, 7 and 10 days</td>
</tr>
</tbody>
</table>

Table 4. Statistics of two individual auctions

<table>
<thead>
<tr>
<th>Auction ID</th>
<th>Product Information</th>
<th>Number of Bidders</th>
<th>Number of Bids</th>
<th>Initial Bid (US $)</th>
<th>Winning Bid (US $)</th>
<th>Average Bid (US $)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Apple iPhone 7 Plus with 256GB Black (Unlocked)</td>
<td>17</td>
<td>37</td>
<td>0.99</td>
<td>710</td>
<td>289.04</td>
</tr>
<tr>
<td>119</td>
<td>Apple iPhone 7 - 128GB - Black (Unlocked)</td>
<td>14</td>
<td>76</td>
<td>0.99</td>
<td>50</td>
<td>19.5</td>
</tr>
</tbody>
</table>

Next, the authors implemented the metrics (normalized to the range of [0, 1]) for the 8 SB patterns. The mathematical explanation of all the patterns has been given in [11]. Then,
the authors measured each metric against each bidder in each of the 807 auctions. The generated SB training dataset has a total of 6321 instances. In the SB dataset, an instance shows the conduct of a participant in a certain auction. It is a vector of ten elements: Bidder ID, Auction ID and eight SB classification patterns [5]. A high value of a metric indicates a high level of doubt concerning a monitored bidder.

4.4 Conclusion

In this chapter we discussed the following points.

We explained the strategies and characteristics of eight unique SB patterns and assigned weights to each of eight patterns. Then we categorized them into three categories; high (successive outbidding), low (auction bid, last bidding and auction starting price) and medium (bidding ratio, winning ratio, buyer tendency and last bidding) according to the weights.

Then we proposed a supervised classification model of SB. Our proposed model works in two phases: offline and online classification. We will explain all the necessary steps for the offline classification in the next chapters. For the online classification, our model takes into effect right after completion of the bidding period and before processing the payment. The SB patterns will be then calculated for all the participants of the current auction and the classification model will be applied to detect suspicious bidders.

Finally, we detailed the actual SB training dataset, which is extracted from the commercial auction site eBay, and contains 6321 instances.
Chapter 5

Labeling Shill Bidding Data via Clustering

In the auction fraud detection domain, it is preferred to use a labeled dataset always; however, it is quite difficult to find labeled data. Hence, the most crucial part to construct the training dataset is to label them. In the case of SB, it is exceptionally unusual to have labelled fraudulent dataset either from the online auction sites or different open sources. To overcome this issue, we come up with a semi-automated approach that uses clustering techniques for labeling the auction data.

5.1 Comparison of Clustering Techniques

Labeling any training datasets is a critical phase for classification. Most of the time, labeling is done manually, which is a very tedious task. To make this task efficient, we first cluster the SB instances/bidders into meaningful groups. A robust clustering technique produces clusters where the distance is maximized between the inter-clusters and minimized in the intra-cluster [22]. Numerous clustering techniques have been proposed, such as K-means, HC, SOM, EM, DBSCAN, BIRCH and CURE. Each clustering algorithm has its own advantages and disadvantages, and which one to choose depends on some factors, e.g., size and dimensionality of training dataset. In Table 5, we
compare these techniques based on the studies conducted in [22-24].

Here,

- \( n \) = total number of data points,
- \( k \) = number of clusters,
- \( d \) = dimensionality,
- \( m \) = prototype of SOMs.

Table 5. Comparison between different clustering algorithms

<table>
<thead>
<tr>
<th>Clustering Method</th>
<th>K-means</th>
<th>HC</th>
<th>EM</th>
<th>BIRCH</th>
<th>CURE</th>
<th>DBSCAN</th>
<th>SOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset Handling Capacity</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Time Complexity</td>
<td>( O(knd) )</td>
<td>( O(n^3) )</td>
<td>( O(knd) )</td>
<td>( O(n) )</td>
<td>( O(n^3\log n) )</td>
<td>( O(n^2) )</td>
<td>( O(n^3m) )</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Handling Noisy Data</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Among the algorithms above, we select HC for our SB dataset for the following reasons:

- In terms of cluster quality, HC is very efficient for datasets of medium size and dimensionality, like our SB dataset. The other advanced clustering methods, such as CURE, DBSCAN and BIRCH, are efficient for large-scale datasets.
• HC handles noisy data effectively.

• HC does not necessitate the user to redefine the number of clusters, and this is a significant advantage over flat clustering (i.e. K-means and SOM). An inappropriate choice of k may lead to poor quality of clusters.

• Even though HC has a high time complexity, it is not an issue as the data clustering is done offline.

We work with hierarchical agglomerative clustering that combines clusters into a single one at each step. We follow three main phases to generate the SB clusters. Firstly, we create the dendrogram by choosing the most appropriate distance/similarity function. A dendrogram is a tree structure illustrating the cluster arrangement of a given dataset. Secondly, we search for the optimal number of clusters using direct and statistical testing methods. Finally, we generate the content of all the SB clusters based on the best distance function and the optimal number of clusters. We carry out all the experiments with Scikit Learn environment using Python and R languages.

5.2 Similarity Measurement for Hierarchical Clustering

There are four functions to measure the similarity of data points: Average Linkage, Single Linkage, Centroid Linkage and Complete Linkage. The first three functions did not work well (see Figure 2, 3 and 4) because our SB dataset are very condensed (in the range of [0, 1]). Single linkage performs poorly in generating the dendrogram. On the other hand, as shown in the dendrogram of Figure 5, Complete Linkage properly partitioned our SB dataset and provided much better results than the three other functions.
Figure 2. HC dendrogram with Average linkage

Figure 3. HC dendrogram with Single linkage
Figure 4. HC dendrogram with Centroid linkage

Figure 5. HC dendrogram with Complete linkage
5.3 Optimal Number of Clusters

From Figure 5, we see that we have several choices for selecting the number of clusters (k). We determine the optimal clustering number using the ‘Silhouette Index/ Average Silhouette’ and ‘Gap Statistic’ methods to remove this uncertainty as explained in [39]; the first technique calculates the average silhouette of the data instances for multiple values for k [40]. Optimal number of clusters maximizes average silhouette from a variety of probable values [40]. We may note that in Sci-kit Learn toolkit, the default range of k is 1 to 10. We separately try three ranges 1 to 10, 1 to 15, and 1 to 30, and for each of which, we obtained 10 as the optimal number of clusters. In the next code, we show the function we have used for the Silhouette Index method where ‘data’ refers to the SB dataset.

```r
require(cluster)
fviz_nbclust(data, hcut, method = "silhouette",
             hc_method = "complete")
```

As shown in Figure 6, the optimal number of clusters obtained with complete linkage for Silhouette Index method is ten for SB dataset.
With the Gap Statistics, the total of intra-cluster variation is compared to several $k$ values with probable values for the ‘null reference distribution’ of samples [41]. The estimation of the optimal number of clusters maximizes the gap statistic, which represents clustering arrangement that is distantly away from ‘random uniform distribution’ of data instances. Next code segment shows the function we used for the Gap Statistic method, and Figure 7 shows the ten optimal clusters we attained using Gap Statistic method.

```r
set.seed(123)
fviz_nbclust(data, hcut, nstart = 25, method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")
```
In figure 3 and 4, to show the optimal number of clusters using Silhouette Index anf Gap statistic methods. We considered the range 1 to 15 as cluster numbers. Both graphs show the even number of clusters in the x-axis from 1 to 15 ranges, and in each case, 10th cluster has obtained the highest silhouette width and gap statistic value, respectively, for Silhouette Index and Gap Statistic methods.

5.4 Cluster Content

Since we obtained ten optimal clusters, now we fit the HC technique to the SB dataset as shown in the next code segment. Then, we plot our SB dataset and generate ten clusters as presented in Figure 8. We show which SB instance belongs to which cluster. Moreover, from Table 6, we can observe that cluster 1 has most of the instances, which is 30.28%, and cluster 5 has the least amount, which is 0.46%.
Figure 8. SB dataset partitioned into 10 clusters

Table 6. Statistics of SB clusters

<table>
<thead>
<tr>
<th>ID</th>
<th>Number of Instances</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 0</td>
<td>274</td>
<td>4.33%</td>
</tr>
<tr>
<td>Cluster 1</td>
<td>1914</td>
<td>30.28%</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>210</td>
<td>3.32%</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>882</td>
<td>13.95%</td>
</tr>
<tr>
<td>Cluster 4</td>
<td>362</td>
<td>5.72%</td>
</tr>
<tr>
<td>Cluster 5</td>
<td>29</td>
<td>0.46%</td>
</tr>
<tr>
<td>Cluster 6</td>
<td>114</td>
<td>1.80%</td>
</tr>
<tr>
<td>Cluster 7</td>
<td>1516</td>
<td>23.98%</td>
</tr>
<tr>
<td>Cluster 8</td>
<td>528</td>
<td>8.35%</td>
</tr>
</tbody>
</table>
5.5 Labeling of Shill Bidding Data

We discuss how we label our SB training dataset with the help of data clustering. In the previous section, we used HC with Complete Linkage as the preferred partitioning technique, and we obtained ten optimal clusters. Before labeling the clusters, we first categorize each SB pattern of each cluster into two groups named ‘Low Category’ and ‘High Category’ [5]. In fact, in each cluster, we compute the average value of each fraud pattern for all the instances in that cluster. A pattern with the average value from 0 to 0.5 is marked as ‘Low’, and from 0.51 to 1 as ‘High’ [5]. Since we will use binary classifications, we chose to have only ‘Low’ and ‘High’ values.

Next, to label a cluster, we manually analyze the category and weight of the SB patterns to make a decision. In Table 7, we demonstrate the labeling of ten clusters. For example, if we consider cluster 2, we obtained five fraud patterns that fall into the ‘High Category’, including the high weighted pattern ‘Successive Outbidding’. An instance with several high values of the SB patterns is probably fraud, especially when highly weighted patterns have very high values. Only three patterns belong to the ‘Low Category’ with very low values. For all these reasons, we label this cluster and its bidders as ‘Suspicious’. Now, considering cluster 3, we obtained only one pattern in the ‘High Category’, which has a low weight. All the other seven patterns belong to the ‘Low Category’, and they all have very low values. All these values strongly indicate that bidders in this cluster are most likely normal. Hence,
we label this cluster as ‘Normal’. We follow the same strategy to label the rest of the eight clusters based on the weights and categories of the fraud patterns.

Table 7. Data analysis and labeling

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Category</th>
<th>Weight</th>
<th>Size</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 0</td>
<td><strong>Low Average Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bidder Tendency (0.26608)</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bidding Ratio (0.429219)</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Auction Bids (0.082903)</td>
<td>Low</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Last Bidding (0.264267)</td>
<td>Medium</td>
<td>4.33%</td>
<td>Suspicious</td>
</tr>
<tr>
<td></td>
<td>Auction Starting Price (0.048829)</td>
<td>Low</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Early Bidding (0.17192)</td>
<td>Low</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>High Average Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Winning Ratio (0.832261)</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Successive Outbidding (0.817518)</td>
<td>High</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster 1</td>
<td><strong>Low Average Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bidder Tendency (0.112438)</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bidder Ratio (0.133076)</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Successive Outbidding (0.003396)</td>
<td>High</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Auction Bids (0.058353)</td>
<td>Low</td>
<td>30.28%</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Last Bidding (0.124179)</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Auction Starting Price (0.000269)</td>
<td>Low</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Early Bidding (0.110384)</td>
<td>Low</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Winning Ratio (0.445501)</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster 2</td>
<td><strong>Low Average Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bidder Tendency (0.387139)</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bidding Ratio (0.344252)</td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Auction Bids (0.375692)</td>
<td>Low</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>High Average Value</strong></td>
<td></td>
<td>3.32%</td>
<td>Suspicious</td>
</tr>
</tbody>
</table>

43
<table>
<thead>
<tr>
<th>Cluster</th>
<th>Average Value</th>
<th>Bidder Tendency</th>
<th>Bidding Ratio</th>
<th>Successive Outbidding</th>
<th>Auction Bids</th>
<th>Last Bidding</th>
<th>Early Bidding</th>
<th>Winning Ratio</th>
<th>Auction Starting Price</th>
<th>Winning Ratio</th>
<th>Early Bidding</th>
<th>Last Bidding</th>
<th>Successive Outbidding</th>
<th>Auction Bids</th>
<th>Last Bidding</th>
<th>Early Bidding</th>
<th>Winning Ratio</th>
<th>Auction Starting Price</th>
<th>Winning Ratio</th>
<th>Early Bidding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 3</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>13.95%</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>High Average Value</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>13.95%</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster 4</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>5.72%</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>High Average Value</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>5.72%</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster 5</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>5.72%</td>
<td>Normal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster 6</td>
<td><strong>Low Average Value</strong></td>
<td><strong>High Average Value</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------</td>
<td>-----------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bidder Tendency (0.211988)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bidding Ratio (0.276177)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last Bidding (0.382686)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auction Bids (0.374955)</td>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Early Bidding (0.255193)</td>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>High Average Value</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auction Starting Price (0.972148)</td>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Winning Ratio (0.86743)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Successive Outbidding (0.881579)</td>
<td>High</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cluster 7</strong></td>
<td><strong>Low Average Value</strong></td>
<td><strong>High Average Value</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bidder Tendency (0.110825)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bidding Ratio (0.042185)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auction Bids (0.464765)</td>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Successive Outbidding (0.004947)</td>
<td>High</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Winning Ratio (0.00033)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>High Average Value</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last Bidding (0.776734)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auction Starting Price (0.975419)</td>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Early Bidding (0.756939)</td>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cluster 8</strong></td>
<td><strong>Low Average Value</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bidder Tendency (0.100074)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bidding Ratio (0.057518)</td>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auction Bids (0.169577)</td>
<td>Low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Now, we have labeled each instance (bidder) in the SB training dataset. Apart from the eight columns for eight SB patterns, we have added one more column named ‘label’ that categorizes the bidder as either a normal bidder or suspicious bidder. This is our final training dataset. A portion of our final training dataset has been shown in Table 8 and in the label column, ‘N’ represents a normal bidder and ‘S’ represents a suspicious bidder.
An analysis of the training dataset is presented in Table 9.

Table 9. Analysis of generated clusters

<table>
<thead>
<tr>
<th>Bidders</th>
<th>Clusters</th>
<th>Dataset Size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Suspicious</td>
<td>Normal</td>
</tr>
<tr>
<td>5694</td>
<td>627</td>
<td>6</td>
</tr>
</tbody>
</table>

From Table 9, we see that among the ten clusters, we obtained four clusters with approximately 9% of all instances, which show an indication of being ‘Suspicious’. The rest of the clusters, representing 91% of the SB dataset, are labeled as ‘Normal’.

5.6 Conclusion

In this section, we summarize the conclusions we found from this chapter.

We selected HC algorithm with complete linkage to have the clusters of bidders from different clustering methods. Then, ten optimal clusters have been obtained by using the ‘Silhouette Index/ Average Silhouette’ and ‘Gap Statistic’ method.
Since, it is almost impossible to find the labeled auction data, a semi-automated labeling approach has been applied to label the ten clusters from the SB dataset. We categorized each SB pattern of each cluster into ‘Low Category’ and ‘High Category’ before labeling the clusters. A pattern with the average value from 0 to 0.5 is marked as ‘Low’, and from 0.51 to 1 as ‘High’. In this way, after labeling, we attained six clusters as ‘Normal’ and four clusters as ‘Suspicious’ based on the weight of eight SB patterns.

From Table 9, we attained two imbalanced classes in our dataset. In the next chapter we discuss about different sampling techniques to solve this imbalanced class problem.
Chapter 6

Sampling of Shill Bidding Data

Imbalanced data usually denotes to the issue where the classes are not divided equally in classification. Learning from imbalanced data deteriorates the predictive performance as shown in [42]. Additionally, the minority class (usually the class of interest) in the imbalanced dataset will be misclassified because baseline classifiers favor the majority class. In fraud detection domain, having screwed class distribution is a serious concern because the fraudulent class tends to be misclassified as ‘normal’. Moreover, classifiers that learn to always predict the majority class may obtain 99% accuracy, but such classifiers are not helpful in identifying the fraud class. Also, classifiers tend to obtain poor accuracy when identifying the fraud class, while the fraud class that carries the highest cost of misclassification.

SB data are imbalanced in nature, and the ratio of majority to minority instances can attain 100:1; sometimes more extreme as 10000:1 [43]. This is due to the inaccessibility to fraud data as they are jammed by the organizers of an auction. Additionally, using artificial data is not recommended because they do not represent the real bidding behavior [16], [18]. Several techniques exist to handle imbalanced learning problems, which are categorized into two groups [43]: data-level approach (data sampling) and algorithm-level approach (cost sensitive learning). The first approach re-adjusts the class
ratio with the purpose to obtain a more balanced class distribution. On the other hand, the second approach assigns different weights to the classes, such as a higher cost to the class of interest. We focus on data sampling methods that increases data to the minority class (over-sampling), decreases data from the majority class (under-sampling), and does both (hybrid sampling).

6.1 Under-Sampling Technique

To make the data set balanced; this sampling method decreases the number of observations from majority class to have less effect on Machine Learning algorithms. This method works best when the dataset is large. Decreasing the number of training data helps to obtain better run time and removes storage problem. One of the major disadvantages of this method is while reducing the number of instances; it may eliminate some important instances from the dataset.

In under-sampling methods, NearMiss technique is commonly used. It keeps only the instances from the majority class, where the average distance to k-nearest neighbor is minimum in minority class [44]. Instead of resampling the Minority class, using a distance, this will make the majority class equal to minority class. NearMiss under-sampling method has three variants [44]. First, In NearMiss-1, it will continue to have data from majority class whose average distance to k-nearest data in the minority class is minimum. Second, NearMiss-2 continues to keep data from the majority class whose average distance to the k-farthest data is minimum in minority class. And the final variant, NearMiss-3 chooses k nearest neighbors in majority class for each data point in the minority class and the under-sampling ratio is directly controlled by k in this case.
One major problem of using under-sampling is that significant information may be lost from the majority class which can cause overly general rules which means samples can be misclassified after classification. This can not afforded in fraud detection especially for fraud samples. Hence, to overcome this problem, ClusterCentroids method has been introduced in [25]. ClusterCentroids under-samples the majority class by replacing majority samples from clusters with the cluster of centroids using K-means algorithm by considering the ratio of majority class samples to minority class samples. This technique performs under-sampling by generating centroids based on k-means clustering methods. The data will be grouped based on the similarity to preserve information. A K-means algorithm is fitted to the data and the number of clusters (k) is obtained by the level of under-sampling. Then, the majority samples from the clusters are entirely substituted by the sets of cluster centroids from K-Means.

6.2 Over-Sampling Technique

This sampling technique adds the number of data to the minority class to have more effect on Machine Learning algorithms. It duplicates the observations from minority class to balance the data. It is also recognized as ‘up-sampling’. One of the advantages of using oversampling is that it does not lead to any information or data loss. The disadvantage of using this method is that, as it adds duplicated observations in the dataset, it may end up adding multiple observations of several types which will eventually lead to over-fitting.

Sophisticated over-sampling methods have been introduced, such as, the famous Synthetic Minority Oversampling Technique (SMOTE). The idea is to solve the over-fitting rendered by simply oversampling by replication and assist the classifier to improve
its generalization on testing data considering k-nearest neighbors of each of the minority instances [26]. The value of k depends on the number of synthetic samples that will be produced.

SMOTE is a statistical method that increases the number of data samples in a dataset. This method works by producing new data instances from the existing minority class that we provide as input. SMOTE implementation does not make any change in the number of data in majority class.

The new data samples are not just duplications of the existing minority class data instances; instead, the algorithm considers data from ‘feature space’ and its closest neighbors for target class, and generates new data which join the target class with the features of its nearest neighbors [26].

SMOTE takes into consideration the whole dataset as input, however, it expands only the percentage of the minority class.

To extend this idea, a new SMOTE based approach named Border-SMOTE has been introduced which applies SMOTE only to the minority instances deemed to be on the decision boundary. Then, the Cluster-based Oversampling technique proposed, which clusters the instances of each class individually, and randomly oversamples the instances within individual clusters to achieve equality both between and within classes. For our experiments, we have focused only on the SMOTE using Sci-kit Learn.
6.3 Hybrid-Sampling Technique

Several extensions of SMOTE have been proposed to achieve a higher efficiency, and the most performing ones are SMOTE-ENN and SMOTE-TomekLink. Both methods combine under and over-sampling, which makes them hybrid methods. They have advantages and flaws of both methods as described above, which is still a trade-off. In addition to over-sampling the minority class via SMOTE, distance-based methods, TomekLink and ENN under-sample the majority class by deleting instances that form Tomeklinks, i.e., borderline and noisy samples that reduce the predictive performance [44].

A Tomeklink can be described as follows [45]: let two data a and b belong to two different classes and m(a, b) is distance between a and b. The pair (a, b) is a Tomeklink if there is no data c, such that m(a, c) < m(a, b) or m(b, c) < m(b, a). If two data sample forms a Tomeklink then one of these two data samples is either noise or borderline. This way, instead of only removing data from the majority class that form Tomeklinks, data from both of the classes are removed. On the other hand, ENN removes data whose class is different from a majority of it’s k-nearest neighbor and keeps removing the data until the remaining dataset is minimal [44]. ENN eliminates more instances than TomekLink. Therefore, it is considered that it offers more data cleaning in depth.

We apply the five sampling methods in Table 10, and consequently we produce five different SB training datasets with different sizes.
Table 10. Sampling of Shill Bidding Dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>Normal Instances</th>
<th>Suspicious Instances</th>
<th>New Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMOTE</td>
<td>Over-sampling</td>
<td>5694</td>
<td>5694</td>
<td>11388</td>
</tr>
<tr>
<td>SMOTE-ENN</td>
<td>Hybrid-sampling</td>
<td>5595</td>
<td>4890</td>
<td>10485</td>
</tr>
<tr>
<td>SMOTE-TomekLink</td>
<td>Hybrid-sampling</td>
<td>5681</td>
<td>5681</td>
<td>11362</td>
</tr>
<tr>
<td>NearMiss</td>
<td>Under-sampling</td>
<td>627</td>
<td>627</td>
<td>1254</td>
</tr>
<tr>
<td>ClusterCentroids</td>
<td>Under-sampling</td>
<td>627</td>
<td>627</td>
<td>1254</td>
</tr>
</tbody>
</table>

6.4 Conclusion

We chose SMOTE, SMOTE-ENN and SMOTE TomekLink, NearMiss and ClusterCentroids, respectively as over-sampling, hybrid sampling and under-sampling sampling for SB dataset. In this chapter we briefly explained the above-mentioned over-sampling, under-sampling and hybrid sampling techniques.

Finally, in Table 10, we showed the original number of instances before applying the sampling techniques and the number of instances we obtained after applying the sampling techniques for ‘Normal’ and ‘Suspicious’ bidders.

In the next chapter we utilize these five sampled and balanced datasets for our experiments using the selected classification models.
Chapter 7

Evaluation of Shill Bidding Classifiers

We select SVM, Random Forest and MLP-Based ANN to develop the fraud classifiers by learning from the five labeled and sampled SB datasets. Also, to obtain the optimal classifiers we have tuned the hyperparameters. Based on the obtained results, we discuss and evaluate the classification performance results for our learning model.

7.1 Hyperparameters Tuning

Tuning the hyperparameters of classification algorithms is usually considered as an optimization problem [46] and its objective function controls the predictive performance of model, which is induced by learning algorithm. The optimal parameter configuration maximizes the performance for the training dataset. Several probabilistic and deterministic approaches have been developed to optimize the parameters. Among them, we use the Randomized Search CV in order to tune efficiently the parameters for SVM, Random Forest, and leave MLP classifier for ANN with default parameters. The reason we have chosen Randomized Search CV over Grid Search CV is that the latter is computationally expensive, especially if we are searching over a large space of
parameters and dealing with multiple parameters. After specifying the parameter values, Grid Search tries every possible combination, which is not time-efficient. Randomized Search CV, on the other hand, takes some distributions to sample from and the maximum number of iterations to try. This allows us to focus the search on areas where the parameters perform the best. Randomized Search CV uses the ‘Stratified k-Fold’, in which we fix the number of folds to 5 and 10. Randomized Search CV has two major advantages than the exhaustive search [47]: 1) budget, i.e. the number of sampled candidates or sampling iterations can be selected independently from the possible values and number of parameters; 2) parameters can be added without affecting the time efficiency.

We use the non-linear kernel RBF, which has been proved efficient in many auction fraud detection problems as auction data are non-linearly separable [5]. Besides, when we used Randomized Search CV, it provided us with RBF as the best kernel. Tables 11 and 12, respectively present the SVM, Random Forest hyperparameters and some of ANN parameters among 21 parameters.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Hyperparameters</th>
<th>Description and Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>Misclassification Cost (C)</td>
<td>C adjusts misclassification of training data instances against the simplicity of decision surface. If C is low, it provides a smooth decision surface, whereas a high C (such as 1000 or more) classifies all data instances properly by providing the model freedom to choose more data instances as support vectors. Hence, it is better to limit C with low value for favoring the models which are faster and consumes less memory.</td>
</tr>
</tbody>
</table>
As a result, we assign the range \([100, 500]\) to \(C\).

**Gamma of RBF**

If the free kernel parameter \((\gamma)\) has a large value then the area of influence for the support vectors cannot include any other data points except themselves, and no amount of \(C\) regularization can prevent it from over-fitting. So, we limit the range of \(\gamma\) to \([0.1, 10]\).

**Random Forest**

- **Number of Estimators**
  - \((n\_estimators)\)
  - This represents the number of trees to build before taking the maximum voting or average of predictions. So, we choose a high range for this number to make the predictions stronger and more stable. The range we choose is \([1, 500]\).

- **Maximum Features**
  - \((max\_features)\)
  - This is maximum number of features for considering the best split for a node in Random Forest. Increasing this parameter generally improves the performance. We have a high number of options to choose from at each node. However, this is not necessarily true as it decreases the diversity of the individual tree. The default value is ‘Auto’ in sci-kit learn for ‘max_features’. Randomized Search CV also provides ‘Auto’ as the best value. This value simply takes all the features that make sense in every tree.

- **Maximum Depth**
  - \((max\_depth)\)
  - This is maximum number of levels in each decision tree. It reduces complexity of the learned model and lowers over-fitting. The range we give for max_depth is \([10, 110]\).

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Parameter</th>
<th>Description and Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hidden Layer Sizes (hidden_layer_sizes)</td>
<td>It passes a tuple containing the number of neurons of MLP for each layer. For the sake of simplicity, we chose three layers of neurons with the same number (30).</td>
</tr>
</tbody>
</table>
### ANN with MLP

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation</td>
<td>For hidden layer, we use rectified linear unit function as activation function.</td>
</tr>
<tr>
<td>Solver</td>
<td>As a weight optimization solver, we use ‘adam’, a stochastic gradient-based optimizer.</td>
</tr>
<tr>
<td>Learning Rate Initialization</td>
<td>It regulates step size while updating the weights. It is used only with the ‘adam’ solver. We set learning_rate_init to 0.001.</td>
</tr>
<tr>
<td>Learning Rate (learning_rate)</td>
<td>Learning rate schedules the weight updates. In our case, we use learning_rate equals to ‘constant’ given by ‘learning_rate_init’.</td>
</tr>
<tr>
<td>Maximum number of Iterations (max_iter)</td>
<td>We set max_iter to 200. For stochastic solvers (like ‘adam’), it decides epochs’ number to regulate how many times each data instance is going to be used.</td>
</tr>
</tbody>
</table>

### 7.2 Performance Metrics

In the fraud detection area, we put more emphasize on the suspicious class as it has a much higher misclassification cost. First, we focus on the detection rate of suspicious bidders by using two metrics: Recall and Precision. We emphasize more on Recall since it denotes how sensitive the classifier is in detecting shill instances. These two metrics are used to calculate another important metric known as F-score, which measures the effectiveness of detecting the fraud class. We employ f1_score = binary, which basically puts more weight on the positive instances, i.e., suspicious ones in our case. Additionally, we use AUROC, which represents the accurateness of a model distinguishing data observations from the two classes. Second, we evaluate the misclassification rate of the
fraud class based on FNR (False Negative Rate) and FPR (False Positive Rate). FNR denotes the percentage of suspicious bidders incorrectly identified as normal, and FPR the percentage of normal bidders incorrectly classified as suspicious. Third, we use the Log-Loss function to evaluate the developed classification models for the SB dataset. By penalising the false classifications, Log-Loss measures the accuracy of classifiers. A perfect classification model has a Log-Loss of 0. If Log-Loss value increases for a classification model that means the predicted labels are diverging from actual labels. Therefore, minimising Log-Loss function corresponds to maximising the accuracy of classifiers.

### 7.3 Classification with Balanced Shill Bidding Datasets

In Tables 13, 14 and 15, we provide the classification results obtained with SVM, Random Forest, and ANN. We developed in total 15 classification models (with 5 and 10-fold for each model) for the three classifiers, using the five sampling techniques.

Table 13. SVM Performance with sampled SB dataset

<table>
<thead>
<tr>
<th>Sampling Method</th>
<th>CV</th>
<th>C</th>
<th>y</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMOTE</td>
<td>5</td>
<td>269.97</td>
<td>3.08</td>
<td>0.95918</td>
<td>0.97295</td>
<td>0.96577</td>
<td>0.96545</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>443.27</td>
<td>1.14</td>
<td>0.96579</td>
<td>0.97366</td>
<td>0.96464</td>
<td>0.96745</td>
</tr>
<tr>
<td>SMOTE-ENN</td>
<td>5</td>
<td>266.09</td>
<td>1.59</td>
<td>0.98429</td>
<td>0.98154</td>
<td>0.98272</td>
<td>0.98277</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>344.65</td>
<td>3.08</td>
<td>0.98215</td>
<td>0.92960</td>
<td>0.95811</td>
<td>0.95980</td>
</tr>
<tr>
<td>SMOTE-TomekLink</td>
<td>5</td>
<td>437.50</td>
<td>3.36</td>
<td>0.97731</td>
<td>0.88006</td>
<td>0.93578</td>
<td>0.93957</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>437.98</td>
<td>1.94</td>
<td>0.98557</td>
<td>0.96404</td>
<td>0.97955</td>
<td>0.97276</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>n_estimators</td>
<td>max_features</td>
<td>max_depth</td>
<td>Precision</td>
<td>Recall</td>
<td>F-measure</td>
</tr>
<tr>
<td>--------------------</td>
<td>----</td>
<td>--------------</td>
<td>--------------</td>
<td>-----------</td>
<td>-----------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>NearMiss</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>221.27</td>
<td>1.44</td>
<td>0.79856</td>
<td>0.86046</td>
<td>0.82835</td>
<td>0.81547</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>396.98</td>
<td>0.64</td>
<td>0.85106</td>
<td>0.88235</td>
<td>0.86642</td>
<td>0.84987</td>
</tr>
<tr>
<td><strong>Cluster Centroids</strong></td>
<td>5</td>
<td>232.49</td>
<td>0.24</td>
<td>0.800000</td>
<td>0.89922</td>
<td>0.84671</td>
<td>0.83075</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>175.60</td>
<td>1.01</td>
<td>0.74418</td>
<td>0.80000</td>
<td>0.77108</td>
<td>0.77404</td>
</tr>
</tbody>
</table>

Table 14. Random Forest performance with sampled SB dataset
Table 15. MLP-based ANN performance with sampled SB dataset

<table>
<thead>
<tr>
<th>Sampling Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMOTE</td>
<td>0.96548</td>
<td>0.95450</td>
<td>0.95996</td>
<td>0.96007</td>
</tr>
<tr>
<td>SMOTE-ENN</td>
<td>0.97467</td>
<td>0.97552</td>
<td>0.97509</td>
<td>0.97254</td>
</tr>
<tr>
<td>SMOTE-TomekLink</td>
<td>0.96016</td>
<td>0.97169</td>
<td>0.96016</td>
<td>0.97169</td>
</tr>
<tr>
<td>NearMiss</td>
<td>0.85826</td>
<td>0.82575</td>
<td>0.84169</td>
<td>0.83724</td>
</tr>
<tr>
<td>ClusterCentroids</td>
<td>0.84172</td>
<td>0.88636</td>
<td>0.86346</td>
<td>0.85074</td>
</tr>
</tbody>
</table>

From the above tables, we see that we have the best overall performance with SMOTE-ENN using 5-fold CV for SVM. But, SVM has better Precision with SMOTE-TomekLink using 10-fold CV than SMOTE-ENN. When compared to SMOTE-TomekLink, SVM with SMOTE-ENN with 5-fold improved its Recall, F-measure and AUROC values by 1.75%, 0.317%, and 1.001%, respectively. For RF, we obtained the best results with SMOTE-ENN, using 5-fold CV on all the evaluation metrics. We note that SMOTE-ENN using 10-fold CV, is the second-best method (the values for each metric are worse than 5-fold CV in every case). For MLP-based ANN, we see that SMOTE-ENN is the clear-cut winner in every situation. Just after SMOTE-ENN, SMOTE-TomekLink performed well for Recall, F-measure and AUROC, but SMOTE has better Precision.

On a final note, for every fraud classifier, SMOTE-ENN outperforms the other four sampling techniques. This is not surprising since it has also already been shown in [31] that SMOTE-ENN performs outstandingly well on 11 real-world churn datasets.

To determine the optimal classifier, we utilize Recall, AUROC, FNR and FPR. In Table
SVM returns the best Recall value (0.98154) outperforming the other two classifiers. Also, it has the least FNR value of 0.01846, which means only 1.846% of suspicious bidders have been labeled as Normal wrongly. It has FPR of 0.01571, which means only 1.571% of normal bidders have been labeled as suspicious erroneously. However, Random Forest provides the best FPR value (0.01351), but since we are more interested in labeling the suspicious class, we give higher priority to FNR than FPR. In conclusion, for our SB detection problem, the SVM classifier combined with SMOTE-ENN (using 5-fold CV) provides the optimal performance.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Detection Rate</th>
<th>Misclassification Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>AUROC</td>
</tr>
<tr>
<td>SVM</td>
<td>0.98154</td>
<td>0.98277</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.97198</td>
<td>0.97803</td>
</tr>
<tr>
<td>ANN with MLP Classifier</td>
<td>0.97552</td>
<td>0.97254</td>
</tr>
</tbody>
</table>

In the Figure 9 and 10, we show the comparison for Recall and AUROC among six evaluation metrics for SB dataset using the above mentioned five sampling methods and, we obtain best result for SVM with SMOTE-ENN for both Recall and AUROC.
Figure 9. Performance evaluation using the sampling techniques for Recall

Figure 10. Performance evaluation using the sampling techniques for AUROC

### 7.4 Classification with Imbalanced Shill Bidding

#### Dataset

Table 17 exposes the evaluation of SVM, Random Forest, and ANN when using the original imbalanced SB dataset (given in Table 9) that Precisions are satisfactory, except for SVM using 10-fold CV. On the other hand, Recall, F-measure and AUROC are low
for all the three classifiers. Although SVM with 5-fold CV achieved a high Precision (0.90), the other performance metrics, especially Recall (0.06338) and F-measure (0.11842), deteriorated very badly. For the non-sampled SB dataset, the best overall performance is obtained with MLP-based ANN classifier.

Table 17. Classification with the original imbalanced dataset

<table>
<thead>
<tr>
<th>Classifier</th>
<th>CV</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>5</td>
<td>0.90000</td>
<td>0.06338</td>
<td>0.11842</td>
<td>0.53134</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.68269</td>
<td>0.50000</td>
<td>0.57723</td>
<td>0.73853</td>
</tr>
<tr>
<td>RF</td>
<td>5</td>
<td>0.96629</td>
<td>0.60563</td>
<td>0.74458</td>
<td>0.80177</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.97802</td>
<td>0.62676</td>
<td>0.76394</td>
<td>0.81268</td>
</tr>
<tr>
<td>MLP based ANN</td>
<td></td>
<td>0.84042</td>
<td>0.67521</td>
<td>0.74881</td>
<td>0.83107</td>
</tr>
</tbody>
</table>

7.5 Comparison on Balanced and Imbalanced Shill Bidding Datasets

Table 18 exposes the best performance results for each of the three classifiers with and without using the sampling techniques from Table 13, 14, 15 and 17.

Table 18. Classification results with and without sampling technique

<table>
<thead>
<tr>
<th>Best Classifier</th>
<th>Sampling</th>
<th>CV</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>Yes</td>
<td>5 (SMOTE-ENN)</td>
<td>0.98429</td>
<td>0.98154</td>
<td>0.98272</td>
<td>0.98277</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>10</td>
<td>0.68269</td>
<td>0.50000</td>
<td>0.57723</td>
<td>0.73853</td>
</tr>
<tr>
<td>RF</td>
<td>Yes</td>
<td>5 (SMOTE-ENN)</td>
<td>0.98649</td>
<td>0.97198</td>
<td>0.97918</td>
<td>0.97803</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>10</td>
<td>0.97802</td>
<td>0.62676</td>
<td>0.76394</td>
<td>0.81268</td>
</tr>
</tbody>
</table>
After applying the sampling techniques, we obtain the best performance for SVM with SMOTE-ENN and 5-fold CV. On the other hand, for SVM without data sampling, all the metrics have deteriorated by 30.160%, 48.154%, 40.549%, and 24.424%, respectively, for Precision, Recall, F-measure and AUROC. Regarding Random Forest, the best performance has been achieved with SMOTE-ENN and 5-fold CV. However, for Random Forest without data sampling techniques, all the performance metrics have declined by 0.847%, 34.519%, 21.524% and 16.535% respectively for Precision, Recall, F-measure, and AUROC. Finally, for MLP-based ANN, again SMOTE-ENN achieved the best results. On the contrary, without using sampling methods, in every case Precision, Recall, F-measure, and AUROC have worsened by 13.425%, 30.031%, 22.628%, and 14.147% respectively.

In the following figure, we show the comparison of classification results (Recall and AUROC) between balanced SB dataset and original imbalanced SB dataset. We see that in each case, for SVM, Random Forest, and MLP based ANN; we obtained better result with sampled SB dataset.
In Table 19, we compare the best classifiers with and without sampling the SB dataset. The ANN classifier provides a Recall of 0.67521 using the imbalanced SB dataset, which means almost 32.479% of suspicious bidders have been labeled as ‘normal’ wrongly. On the other hand, using sampling with SVM, we obtained a Recall value of 0.98154, which means only 1.846% of suspicious bidders have been misclassified. Also, Precision, F-measure and AUROC have been increased by 14.387%, 23.391%, and 15.170% respectively, for SVM classifier after sampling, when compared to best classifier without sampling. Moreover, the SVM classifier with the sampled SB dataset obtains a very low value (0.07903) for the Log-Loss function for which reflects that our predicted labels have diverged in very less amount from actual labels. On the other hand, ANN with MLP classifier obtains quite high value (1.39247) for the Log-Loss function which means the predicted labels for this classifier have diverged in good amount from the actual labels for the unsampled SB dataset. To conclude, Table 18 and 19 demonstrate that sampling techniques greatly improve the predictive performance of classifiers.

Figure 11. Comparision of Recall and AUROC for balanced and imbalanced SB dataset
Table 19. Best classification results with and without sampling

<table>
<thead>
<tr>
<th>Best Classifier</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
<th>AUROC</th>
<th>Log-Loss Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM (with sampling)</td>
<td>0.98429</td>
<td>0.98154</td>
<td>0.98272</td>
<td>0.98277</td>
<td>0.07903</td>
</tr>
<tr>
<td>MLP based ANN with MLP (without sampling)</td>
<td>0.84042</td>
<td>0.67521</td>
<td>0.74881</td>
<td>0.83107</td>
<td>1.39247</td>
</tr>
</tbody>
</table>

### 7.6 Conclusion

In this section we summerize this chapter and experimental results we obtained.

We chose Randomized Search CV for tuning the hyperparameters for SVM and Random Forest and left MLP based ANN with their default parameters. We decided to prioritize Recall out of six performance evaluation metrics to choose the optimal classification model since it shows the sensitivity of the classifier in detecting shills.

We presented a comparison between SVM, Random Forest and MLP based ANN for the five sampled SB dataset. We found out that SVM classifier outperforms other classifiers with SMOTE-ENN using 5-fold CV.

Then, we showed a comparison between these three classifiers using the original imbalanced SB dataset. MLP based ANN did perform well in terms of Recall than other classifiers.

Finally, we presented a comparison between the best classification results for balanced
and imbalanced SB dataset. For each classification model, balanced SB dataset did extremely well and provided better performance results.
Chapter 8

Conclusion and Future Work

This chapter demonstrates all assumptions that we have drawn from this thesis along with the experimental results, and finally we talk about future work as extensions of this research.

8.1 Conclusion

With the increasing need of e-commerce applications, auctions have become an ever-growing industry [19]. Online auctions provide users with great convenience, but they also come with a risk. One such risk is SB, which is a type of fraud, wherein the price of products (goods and services) is inflated in an unethical manner. To increase the user trust, it is necessary to make this environment secure by detecting SB before processing the payment of products (goods and service) in order to minimize financial losses for legitimate buyers. Nevertheless, the complexity of SB behavior makes it difficult to detect. It has become immensely important to develop monitoring systems for analysing and tracking SB activities. There are limited studies on shill bidding because of the lack of training data. For this purpose, we developed an efficient shill bidding classification model based on real data from eBay. Firstly, we applied a HC technique to partition training data into clusters of bidders with similar behavior. Subsequently, we utilized a
semi-automated approach to label clusters and their bidders as Normal or Suspicious. Then, to solve the imbalanced learning problem, we applied advanced over, under and hybrid sampling methods to our SB dataset.

In this study, we conducted several experiments to detect the shill bidders, specific findings are given below:

- The hybrid sampling method SMOTE-ENN is the most performing among all the sampling methods (under, over and hybrid).

- SVM returned the highest accuracy in distinguishing suspicious bidders depending on the measured fraud patterns with Recall of 98.154% compared to Recall of 97.198% with Random Forest and Recall of 97.509% with MLP-based ANN.

- For the imbalanced SB dataset, the best classification results in terms of Recall and AUROC have been attained with ANN; while SVM provided the worst Recall value of 0.06338.

- After sampling, all the three classifiers greatly improved their performance. We compared the best classifications results for each classifier with and without sampling. SVM improved by 30.160%, 48.154%, 40.549%, and 24.424%. Random Forest improved by 0.847%, 34.519%, 21.524% and 16.535%. ANN with MLP classifier improved by 13.425%, 30.031%, 22.628% and 14.147%, respectively for Precision, Recall, F-measure and AUROC for the sampled SB dataset.
Furthermore, we conducted a comparison between the best performance results with and without data sampling. Precision, Recall, F-measure and AUROC have been improved by 14.387%, 30.633% 23.391% and 15.170% respectively for SVM classifier when compared to ANN classifier without using any sampling techniques. Also, the Log-Loss function for the SVM classifier provided very low value (0.07903) comparing to the Log-Loss function (1.39247) for the ANN classifier.

In summary, we conclude that data sampling greatly improved the predictive performance, and SVM classifier combined with SMOTE-ENN provides the most satisfactory detection and misclassifications rates. Indeed, as previous empirical studies have shown, classification accuracy can be maximized by combining data sampling with SVM [31].

8.2 Future Work

This present work can lead to a few research possibilities, including:

- In real-life scenarios, hundreds of auctions occur simultaneously. To scale up with this huge traffic, our fraud classifier can be deployed on multiple autonomous agents. More precisely, for each new auction, an agent can be created dynamically to classify the bidders, and then destroyed once the auction is completed or cancelled as shown in [11].

- Develop a shill bidding classifier that is able to evolve constantly with new bidding data and trends. To this end, we can develop an adaptive fraud classifier
based on incremental and decremental learning with SVM. SVM can be selected as it returned the best classification performance in our study.

- We plan to apply Ensemble Learning techniques in future to compare the results with SVM, Random Forest and MLP based ANN for SB dataset.
References


[7] John Dobrzynski. "3 men are charged with fraud in 1,100 art auctions on


