Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

Detection of Auction Fraud in Commercial Sites

Farzana Anowar[1] and Samira Sadaoui[2]

[1] University of Regina, Department of Computer Science, Regina, Canada, fad469@uregina.ca
[2] University of Regina, Department of Computer Science, Regina, Canada, sadaouis@uregina.ca

Received 13 October 2018; received in revised form 12 December 2018; accepted 7 January 2019

Abstract

Online auctions have become one of the most convenient ways to commit fraud due to a large amount of money being traded every day. Shill bidding is the predominant form of auction fraud, and it is also the most difficult to detect because it so closely resembles normal bidding behavior. Furthermore, shill bidding does not leave behind any apparent evidence, and it is relatively easy to use to cheat innocent buyers. Our goal is to develop a classification model that is capable of efficiently differentiating between legitimate bidders and shill bidders. For our study, we employ an actual training dataset, but the data are unlabeled. First, we properly label the shill bidding samples by combining a robust hierarchical clustering technique and a semi-automated labeling approach. Since shill bidding datasets are imbalanced, we assess advanced over-sampling, under-sampling and hybrid-sampling methods and compare their performances based on several classification algorithms. The optimal shill bidding classifier displays high detection and low misclassification rates of fraudulent activities.

Keywords: Auction fraud, Fraud detection, Shill bidding, Data clustering, Data labeling, Data sampling, supervised classification

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

1 Introduction

This section presents the problem statement and scope as well as our research contributions about the detection of bidding fraud in commercial auctions.

1.1 Problem and Motivation

Over the last twenty years, the use of online auctions has rapidly increased in numerous domains, such as antiques, vehicles, and real estate. Since 2002, several commercial auction companies, such as Trade Me and eBay, have become immensely popular [7], [22]. Given this surge in popularity, it is perhaps unsurprising that the FBI's Internet Crime Complaint Center reports that auction fraud has become one of the top forms of cyber-crime (Site 1). E-auctions are vulnerable to three types of fraud: pre-auction (like misrepresentation of products), post-auction (like non-delivery of products), and in-auction (like Shill Bidding and Bid Shielding) [2], [3]. The present research focuses on Shill Bidding (SB) because, unlike the other two types of fraud, it does not leave any obvious evidence. SB is one of the most common auction frauds and also the most difficult to detect because it closely resembles normal bidding behavior [14]. To maximize the seller's revenue, a shill bidder drives up the price of products by submitting many bids via multiple fake accounts. In the case of expensive products, SB will result in substantial financial losses for the auction winners. Furthermore, excessive SB could lead to the failure of the market [2]. Indeed, the presence of shills will discourage honest users from participating in online auctions, and this may in turn negatively affect the auctioning business.

We examined the two popular auction sites, eBay and Trade Me, and could not find any SB detection service. Online auctions are one of the most profitable e-commerce applications because they generate high revenues. As an example, in 2017, eBay claimed that the net revenue reached 9.7 billion US dollars (Site 2). Regardless of their popularity, e-auctions remain very vulnerable to SB fraud as demonstrated in several empirical studies and lawsuits against shill bidders. Note that over the years, auction users complained about SB in several forums and blogs (Site 3).

EBay focuses mostly on user authentication and feedback ratings. The authentication is carried out through Master and Visa cards by using software named FADE. EBay states *If you think that another member is trying to manipulate feedback and Shill Bidding, you don't need to report it to us. eBay has a number of systems in place to detect and monitor bidding patterns and practices, and feedback usage. If we identify any malicious behavior, we'll take steps to prevent it*. There are only two policies posted on eBay site: a seller cannot bid in his own auction using another account, and an employee cannot bid for his company (Site 4). Yet, a user can easily create fake accounts and use dynamic IP addresses. In this case, a shill bidder cannot be tracked. That is why analyzing the bidding behaviour is the only solution to detect SB. eBay did not show what the systems are in place to monitor SB fraud. Most auction sites encourage users to provide feedback ratings of other users. eBay uses these ratings to help detect SB. Nevertheless, ratings can also be easily manipulated to improve the reputation of users. Since ratings cannot be fully trusted, they may bias the classification outcome. Trade Me has two teams, *Trust and Safety* and Customer Support, which continuously monitors and removes inappropriate listings. Trade Me allows users to complain about SB and any other suspicious activities. Still, it is very challenging for users to manually examine the bidding behaviour as an auction involves many bidders and bids, and long bidding duration. Also, in long duration, a shill bidder can easily mimic normal bidding behaviour to avoid being detected [10]. Trade Me mentioned SB but did not show how to detect this fraud. The site gave examples of scam emails and suspicious behaviour but not related to SB. It also provided some manual tips, such as always check the seller's feedback history, never send money overseas, and keep all the payment details and emails exchanged with the seller (Site 5).

Over the years, several sellers and their collaborators have been prosecuted for SB. The following list provides some examples of such cases:

1. In 2012, the auction site, Trade Me, was forced to take payments of $70,000 to each victim of SB fraud that had been committed by an auto 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 (Site 6).

2. In 2014, according to the report of New York Post, a California court accused a real estate auctioning company (Site 7), nicknamed the *eBay of real estate*, for increasing the price of its offerings. Indeed, the *Village at Redlands Group* blamed this company or either using shills or helping the loan holder to obtain the property with 2 million dollars more than the secured winning bid (Site 8).

3. In 2015, a former executive of *Mastro Auctions* was sent to the federal prison for 20 months for using fake bids to increase the price of his company's products and earn more revenue for himself and his company (Site 9).

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

4. As stated in a 2017 article in 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 federal debt (Site 10).

These cases highlight the immense importance of detecting fraudulent bidding activities, as, if left undetected, such activities will almost certainly cause significant financial losses for legitimate buyers. To this end, several empirical studies have sought to demonstrate the presence of SB activities in different commercial auction sites [13], [21]. Although common fraud prevention techniques, such as user authentication, fraud awareness, and feedback ratings, may be reliable for small-scale applications, they do not provide any protection against SB. Consequently, it has become crucial to analyze the behavior of bidders and sellers in order to detect SB and to prevent honest buyers from becoming victims. Fortunately, robust SB detection systems can be obtained through the use of Machine Learning Techniques (MLTs).

1.2 Contributions

Efficient supervised classifiers have been built to control fraud in various sectors, including the credit card, telecommunication, electricity, and insurance industries. However, there are few studies in the literature that classify the bidding transactions of users due to a lack of SB training datasets. More precisely, this lack is attributable to several challenges associated with producing SB data: 1) determining SB strategies that are relevant; 2) implementing robust metrics for the SB patterns; 3) scraping auctions from commercial sites; 4) preprocessing auction data; and 5) evaluating the SB metrics against the extracted data. Auction sites provide a tremendous volume of public data (anonymous), and researchers can employ MLTs 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 research is to develop an optimal SB classification model. For this purpose, we trained several binary classifiers using a collection of the most relevant SB patterns commonly observed in infected auctions. The authors in [2] produced a high-quality SB (unlabeled) training data set by crawling a large number of auctions for a popular product on eBay. We used this new SB dataset to develop our fraud classifiers. Labeling any training dataset is critical for the task of classification. Most of the time, this is done manually, which is a tedious and an error prone operation, especially for large and complex datasets. To facilitate the labeling of our SB dataset, we first utilized hierarchical clustering to group participants with similar bidding behavior, as well as direct and statistical testing methods to find out about the optimal number of clusters. After grouping the bidders, we applied the labeling strategy introduced in [14] to determine whether the bidders in a given cluster were behaving normally or suspiciously. However, the labeled SB training dataset is imbalanced, which poses a challenge because the classifiers have difficulty learning from such datasets. Indeed, an imbalanced dataset deteriorates the predictive performance of the classifiers. Additionally, the minority class, which has the highest misclassification cost, will be misclassified because classifiers favor the majority class. To overcome this problem, we employed different types of intelligent sampling methods: Synthetic Minority Over-sampling Technique (SMOTE), SMOTE-Edited Nearest Neighbor (ENN), SMOTE-TomekLink, NearMiss and ClusterCentroids. Hence, we generated five balanced SB datasets with different sizes. Next, we applied three well-known classification algorithms, Support Vector Machine (SVM), Artificial Neural Network (ANN) with Multi-Layer Perceptron (MLP) and Random Forest, to the sampled datasets as well as to the original imbalanced dataset. Based on several performance metrics and the Random Search Cross Validation, we performed three major experiments. In the first experiment, we searched for the highest performing SB classifier by comparing the results of the five sampling methods combined with the three classifiers. In the second experiment, we evaluated the accuracy of the three classifiers when applied to the imbalanced SB dataset. Finally, in the last experiment, we compared the predictive performance classifiers when applied to balanced and imbalanced SB datasets. In total, we developed eighteen fraud classifiers, including fifteen that use sampling techniques and three that do not.

For the benefit of potential users, our SB classification service can be deployed on any auction site, which is advantageous because existing SB classification solutions are mostly offline. Moreover, in past SB studies, a training instance denotes the behavior of a bidder in all auctions that they participated in. This makes it impossible to know which auctions are infected by fraud. In our work, an instance captures the conduct of a bidder in a certain auction. Another issue is that SB fraud cannot be prevented. Consequently, we measure the bidder-per-auction SB patterns for each bidder just at the end of the bidding period for each monitored auction, which are then analyzed by the fraud classifier in order to detect malicious activities. If the auction is found to be infected by fraud, the administrator will hold any payment for the auctioned product (goods or services) until the investigation is completed.

2 Related Works

Researches on SB detection can be categorized in different ways depending on their underlying approaches, such as statistical, concurrent and machine learning. We may mention that very few studies employed MLTs to distinguish between shill bidders and normal bidders.

In [25], the authors proposed a semi-automated approach combining *one-class SVM'* and *Decision Tree* to detect shill bidders. Based on the one-class SVM learning, first this approach finds all the outliers whose are basically shill bidders. After that, the outliers are sent to the Decision Tree classifier for further analysis because some innocent bidders may

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

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 whose classification is erroneous. Finally, this updated decision tree is considered as the optimal model to detect shill bidders. To carry out the experiments, the authors collected information on 59,949 bidders and 67,244 products from a Chinese website named *Wowma* and detected 77.5% of shill bidders by the proposed method. 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 [13], 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 will be first investigated using additional evidences, and then fed to the classifier. The model ANN sometimes suffers from local minima, and ANN is computationally expensive since it has multiple parameters to tune at once [9]. In addition, ANN does not explain the behavior of the network, which ultimately reduces its trust [26]. In [13], 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.

[24] implemented a synthetic data generator to build a classification model to detect *competitive shilling* fraud. The synthetic data generator uses an agent-based simulation to generate data based on several fraudulent behaviors. The authors followed three steps to produce the training data. Firstly, an agent is defined with a certain fraud type, secondly, the defined agent generates synthetic data, and finally, the generated data are 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. In this paper, the authors did not provide the SB features. Also, using synthetic data is not preferred in fraud detection because they do not represent actual auction data.

In [16], 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 a user to create multiple accounts by using 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 only do not offer enough information 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.

Recently, an SVM-based SB detection model has been developed in [14] that overcome the issues of ANN and HMM. Since the class imbalance is a serious fact in fraud detection problems, the authors applied three data sampling methods (SMOTE, SpreadSubSample, and hybrid of both). Our ultimate goal is to greatly increase the predictive performance results of [14] by using a more advanced and robust preprocessing and classification process. The differences between [14] and the present study are in terms of the training datasets, clustering, sampling and classification approaches, and classification results. In [14], the authors used an available dataset containing 149 PDA auctions. However, an important attribute was missing (seller ID). To be able to build the SB dataset, [14] produced artificial values for this attribute. Yet, to perform a valid empirical assessment, it is better to use actual bidding data because they represent the real behaviour of users. That is why in this research, we employ auctions of iPhone 7 crawled from eBay in 2017. Note that in addition to the two base classifiers SVM and ANN with MLP, our paper utilizes an ensemble classification method called Random Forest.

3 A Supervised Classification Model of SB

This section first describes the behavior of the selected SB patterns and then provides an overview of the SB detection framework for both offline and online fraud classification.

3.1 SB 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 1, we use eight SB strategies [14], [21] 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),

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

shills (a seller and/or his 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 SB pattern is unique as it represents one aspect of the bidding behavior that occurs in a certain bidding stage. A pattern can be computed from the characteristics of an auction, or a bid transaction, or a bidder. The uniqueness of the 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 [2], [21].

Table 1: Characteristics of SB behaviour

Name	Description	Motive	Level	Weight
Auction Starting Price	Seller unusually sets a cheaper starting price than other concurrent auctions	Draws attraction of the people in the auction	Auction	Low
Early Bidding	Bidder bids in the early hours of the auction	To allure honest bidders to take part in the auction	Bid	Low
Bidding Ratio	Bidder participates much more as opposed to normal bidders	To inflate the price and draw higher bids from honest bidders	Bid	Medium
Successive Outbidding	Bidder outbids himself with several repeated small bids	To increase the auction price gradually	Bid	High
Last Bidding	Bidder does not bid at the final phase of the auction	To stop the bidder from winning an auction	Bid	Medium
Buyer Winning Ratio	Bidder bids aggressively but hardy wins in auctions	The target is to increase the price not to win the auction	Bidder	Medium
Buyer Tendency	Bidder takes part in auctions for some sellers exclusively	Secretly colludes with the fraudulent seller	Bidder	Medium
Auction Bids	Total number of bids is much higher comparing to an auction without having shills	To make a product appear more popular	Auction	Low

It is significantly important to give weights to the SB patterns to be able to label bidders into fraud or not fraud. Through scenarios, we show how to weight the eight SB patterns:

- At the early bidding stage, a seller may submit a low amount as the starting price to attract bidders. A legitimate bidder may participate to get a product with a low price. On the other hand, a colluding bidder might compete just to increase the price. From these two scenarios, we can expect that the two patterns, *Early Bidding* and *Auction Starting Price*, maybe somehow similar to normal bidding. Moreover, there may be bidders who bid aggressively only to win a product in an auction without shilling, and usually an auction with shills has more bids than regular auctions. Hence, sometimes *Auction Bids* can be similar for an auction with or without shilling. In conclusion, 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 sometimes same values for the four patterns *Bidding 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 other hand, a fraudulent user 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. So, we assign to these two patterns a medium weight. Moreover, a legitimate bidder may participate in few auctions of a specific seller for his desired product, and a fraudulent bidder tends to participate only for those sellers with whom he has secretly colluded. In this

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

scenario, both bidders may have similar buyer tendency. Hence, we assign a medium weight to *Buyer Tendency* as well.

3.2 Supervised Classification of SB

As depicted in Figure 1, we show how to develop offline an efficient SB classifier, and once online how it operates 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 and also a product whose auctions may have encouraged SB activities. Nevertheless, as shown in [2], the original auction dataset contains noisy data and inappropriate data formatting. Therefore, preprocessing the auction dataset is critical for properly measuring the SB patterns.

Next, we evaluate the fraud patterns against the extracted auction data, and as a result we generate the training dataset. Now, we need to label the SB data using a certain MLT, such as unsupervised learning (like data clustering), semi-supervised learning or active learning. As any labeled SB dataset is imbalanced, consequently we solve this *Imbalanced Learning Problem* via *Data Sampling* or Cost Sensitive *Learning*. Otherwise, a classifier will be influenced by 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 and then comparing several fraud classifiers based on multiple performance metrics.

As illustrated in Figure 1, our model takes effect just after completing the bidding task and before processing the payment of the auctioned product. Once all the bid transactions are available, they are first preprocessed. Based on the concept of "autonomous agents" [21], we can fully automate the preprocessing tasks, such as detecting outliers, removing or imputing missing values, and converting data into proper formats. Moreover, the SB metrics (based on the attributes of the auction dataset) have been implemented with MSSQL language, and these metrics can be automatically calculated from any auction dataset [2]. Next, for each of the participants of the current auction, the SB patterns are measured based on the bidding data. Subsequently, the classification model is launched to detect suspicious bidders. The latter are then sent for a further verification based on other fraud evidence of users.

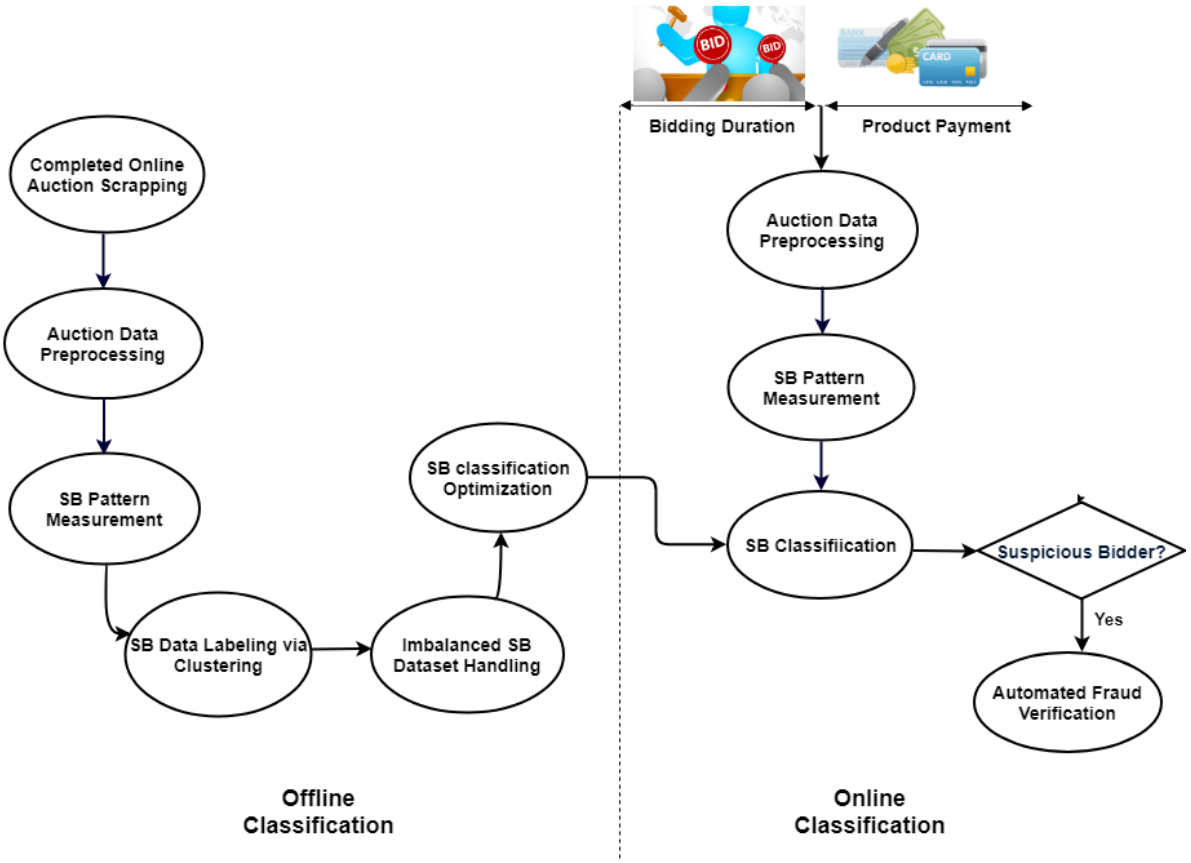

Figure 1: Supervised classification of SB fraud

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

3.3 Actual SB Training Data

In [2], the authors crawled from eBay a good number of auctions of the very hot product *iPhone 7* for the period of March to June 2017. They selected *iPhone 7* for factors that might have 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 [10]; 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 [10].

In [2], 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. Table 2 provides statistics about the preprocessed auction dataset of *iPhone 7*. The dataset consists of different types of iPhone 7 such as iPhone 7 with 32 GB, iPhone 7 plus with 128 GB and iPhone 7 plus with 256 GB. In Table 3, we also give a short summary of two auctions that we selected randomly.

Table 2: Preprocessed auction dataset of iPhone 7 [2]

No. of Completed Auctions	807
No. of Records	15145
No. of Bidder IDs	1054
No. of Seller IDs	647
Avg. Winning Price	US $ 578.64
Bidding Duration	1, 3, 5, 7 and 10 days

Table 3: Statistics of two individual auctions

Auction ID	Name of the Product	Number of Bidders	Number of Bids	Initial Bid (US $)	Winning Bid (US $)	Average Bid (US $)
6	iPhone 7 Plus Black-256GB-(Unlocked)	17	37	0.99	710	289.04
119	iPhone 7 - Black-128GB-(Unlocked)	14	76	0.99	50	19.5

Next, the authors in [2] implemented the metrics (scaled to the range of [0, 1]) for the eight SB patterns. The mathematical explanation of all the patterns has been given in [21]. 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. A high value of the metric indicates a high level of doubt about the monitored bidder.

4 Clustering of SB Data

Among well-known data clustering algorithms, this section first shows that Hierarchical Clustering (HC) is the most appropriate for our training dataset. Then, when applying HC, we need to determine the best similarity function and the optimal number of clusters as well. Lastly, we provide the statistics of the generated clusters.

4.1 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 easier, 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 [11]. A cluster is a group of data points that have the maximum similarity between them and dissimilarity to the data of other clusters. Numerous clustering techniques have been proposed, like K-means, HC, SOM, EM, DBSCAN, BIRCH and CURE. Each clustering algorithm has its own pros and cons, and which one to choose depends on facts like dimensionality and size of the training dataset. In Table 4, we compare these techniques

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

based on the studies conducted in [11], [18], [20]. Here, n number of data, k number of clusters, d dimensionality, and m prototype of SOM.

Table 4: Comparison between different clustering algorithms

Clustering Method	K-means	HC	EM	BIRCH	CURE	DBSCAN	SOM
Dataset handling capacity	High	Medium	High	High	High	High	Medium
Time complexity	O(knd)	$O(n^2)$	O(knd)	O(n)	$O(n^2 \log n)$	$O(n^2)$	$O(n^2 m)$
Accuracy	Low	High	Low	High	High	Low	High
Handling noisy data	No	Yes	No	Yes	Yes	No	No

Among the algorithms above, we select HC for our SB dataset owing to 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 require the user to predefine the cluster number, 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 an offline operation.

We work with hierarchical agglomerative clustering that combines clusters into a single one at each step [5]. 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 set of data points. 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 Sci-kit Learn environment using Python and R.

4.2 Similarity Measurement for HC

There are four functions to measure the similarity of data points: *Single Linkage, Average Linkage, Centroid Linkage* and *Complete Linkage*. The first three functions did not work well because our SB data are very condensed (in the range of [0, 1]). Single Linkage performs worst for SB dataset. As shown in the dendrogram of Figure 2, Complete Linkage properly partitioned our SB data and provided much better results than the three other functions.

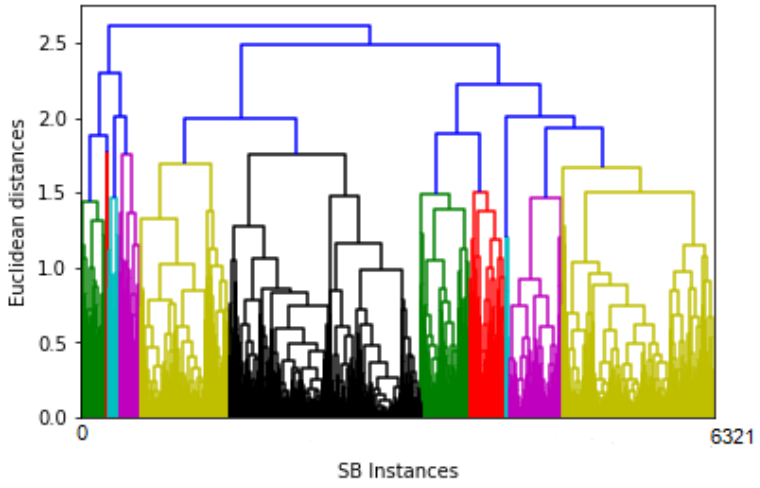

Figure 2: HC dendrogram with Complete Linkage

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

4.3 Optimal Number of Clusters

From Figure 2, 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. The first technique calculates the average silhouette value of the instances for multiple k values [23]. Optimal number of clusters maximizes the average silhouette score [23]. We may note that in Scikit-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 these ranges, we got 10 as the optimal number of clusters and Figure 3 depicts the optimal number of clusters using Silhouette Index with Complete Linkage.

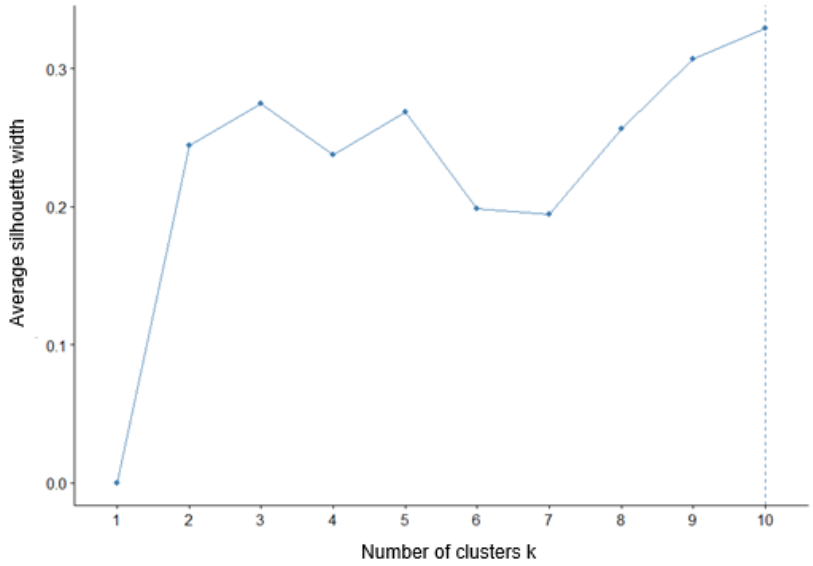

Figure 3: Optimal clusters obtained from Silhouette Index with Complete Linkage

With the Gap Statistics, the total of intra-cluster variation is compared to several k values with probable values under the *null reference distribution* of instances [27]. The estimation of optimal cluster numbers maximizes the gap statistic, which represents clustering arrangement that is far-off from *random uniform distribution* of data instances. Figure 4 shows optimal cluster numbers obtained from Gap Statistic method.

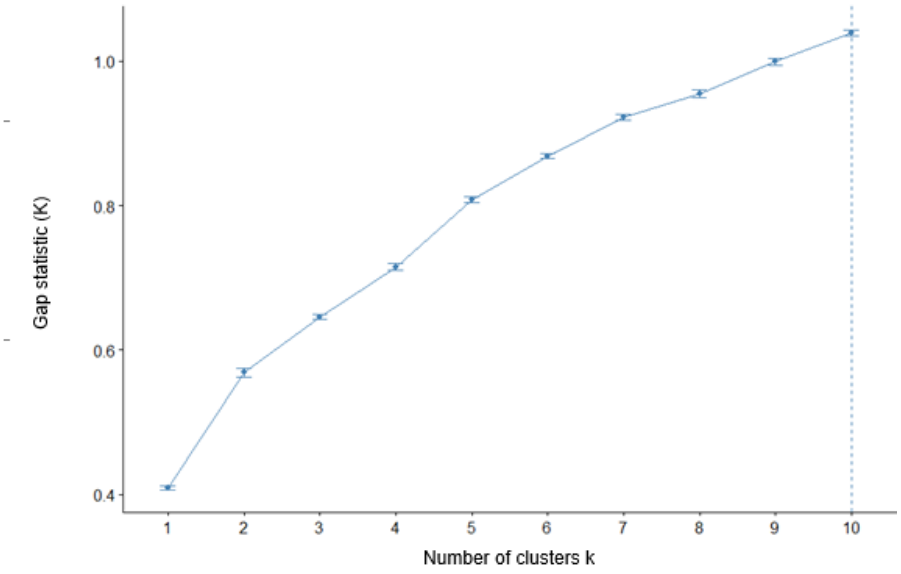

Figure 4: Optimal clusters obtained from Gap Statistic method

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

4.4 Cluster Content

Now we plot our SB dataset and generate the 10 clusters as presented in Figure 5. We show which SB instance belongs to which cluster. Moreover, from Table 5, 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%.

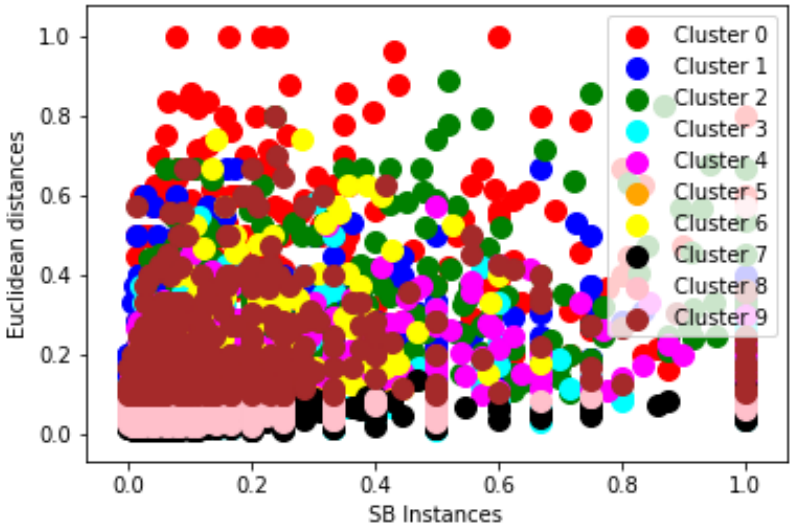

Figure 5: SB dataset partitioned into ten clusters

Table 5: Statistics of SB Clusters

ID	Number of Instances	Percentage
Cluster 0	274	4.33%
Cluster 1	1914	30.28%
Cluster 2	210	3.32%
Cluster 3	882	13.95%
Cluster 4	362	5.72%
Cluster 5	29	0.46%
Cluster 6	114	1.80%
Cluster 7	1516	23.98%
Cluster 8	528	8.35%
Cluster 9	492	7.78%
Total	6,321	≈ 100%

5 Labeling of SB Dataset via Clustering

We are going to 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 10 optimal clusters. Before labeling the clusters, we first categorize each SB pattern of each cluster into two groups named *Low Category* and *High Category* as done in [14]. In fact, for 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*.

Next, to label a cluster, we manually analyze the category and weight of the SB patterns to make a decision. In Table 6, we demonstrate the labeling of 2 clusters. For example, if we consider cluster 2, we have five fraud patterns that fall in the *High Category*, including the high weighted pattern *Successive Outbidding*. An instance with several high values of the SB features is probably fraud, especially when high 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, if we consider cluster 3, we obtain only one pattern in the *High Category*, and this pattern 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.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

Table 6: Labeling the clusters of bidders

Cluster ID	Average Value in a Cluster	Weight	Size (%)	Label
Cluster 2	Low Category Bidder Tendency (0.387139) Bidding Ratio (0.344252) Auction Bids (0.375692) High Category Successive Outbidding (0.969048) Last Bidding (0.892746) Starting Price Average (0.70136) Early Bidding (0.862606) Winning Ratio (0.899158)	 Medium Medium Low High Medium Low Low Medium	3.32%	Suspicious
Cluster 3	Low Category Bidder Tendency (0.138401) Bidding Ratio (0.081071) Successive Outbidding (0.019841) Auction Bids (0.31197) Last Bidding (0.085009) Early Bidding (0.070112) Winning Ratio (0.224646) High Category Auction Starting Price (0.973062)	 Medium Medium High Low Medium Low Medium Low	13.95%	Normal

From Table 7, we can see that among the 10 clusters, we got 4 clusters with approximately 9% of instances that show an indication of being *Suspicious*. The rest of the clusters, representing 91% of the SB dataset, are labeled *Normal*.

Table 7: Overall analysis of cluster labeling

Bidders		Clusters		Dataset Size (%)		Imbalance Ratio	
Normal	Suspicious	Normal	Suspicious	Normal	Suspicious	Normal	Suspicious
5694	627	6	4	91%	9%	9	1

6 Sampling of SB Dataset

Several classification problems, such as fraud detection, software quality prediction and text categorization, possess only 1% or less of the count of instances of the class of interest. With imbalanced training data, classifiers may have a bias towards the majority class. 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 vital to detect since it carries the highest cost of misclassification. Furthermore, a screwed class distribution always deteriorates the predictive performance as demonstrated in [15].

SB data are imbalanced in nature, and the ratio of majority to minority instances can attain 100:1, sometimes more extreme as 10000:1 [1]. This is due to the inaccessibility to fraud data as the organizers jam them. Additionally, using artificial data is not recommended because they do not represent the real bidding behavior [13], [24]. Several techniques exist to handle this *Imbalanced Learning Problem* and those are grouped to two categories: data-level approach (*Data Sampling*) and algorithm-level approach (*Cost Sensitive Learning*) [1]. The first approach re-adjusts the class ratio to obtain a balanced class distribution. The second approach assigns different weights to the classes, such as a higher weight to the class of interest. We focus on data sampling method that increases data to the minority class (*Over-Sampling*) or eliminates data from the majority class (*Under-Sampling*) or does both (*Hybrid Sampling*).

Sophisticated over-sampling methods have been introduced, i.e. the famous *Synthetic Minority Oversampling Technique* (SMOTE). The idea behind SMOTE is to overcome over-fitting on testing data provided by oversampling by duplication and help classifiers to enhance their generalization [12]. Moreover, 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 [17]. In addition to over-sampling the minority class via SMOTE, the distance-based method TomekLink and ENN under-sample the majority class by deleting instances that form TomekLinks i.e. borderline and noisy samples that lowers the predictive performance. Actually, ENN eliminates more instances than TomekLink.

Regarding under-sampling, NearMiss is a well-known technique. It keeps only the instances from the majority class whose mean distance to the K-nearest neighbors is the lowest. One major problem of using under-sampling is that

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

significant information may be lost from the majority class. This can cause overly general rules since under-sampling abolishes instances from majority class. Hence to overcome this problem, ClusterCentroids method has been introduced (Site 11), [19]. ClusterCentroids under-samples the majority class by replacing cluster of samples by the cluster of centroids using K-means algorithm. In Table 8, we apply the five sampling methods, and consequently we produce five different SB training datasets with different sizes.

Table 8: Sampling of SB dataset

Method	Type	Normal Instances	Suspicious Instances	New Size
SMOTE	Over-sampling	5,694	5,694	11,388
SMOTE-ENN	Hybrid-sampling	5,595	4,890	10,485
SMOTE-TomekLink	Hybrid-sampling	5,681	5,681	11,362
NearMiss	Under-sampling	627	627	1,254
ClusterCentroids	Under-sampling	627	627	1,254

7 Classifier Optimization and Evaluation

We select two standard classification algorithms, SVM [14] and MLP-based ANN [6], [26] as well as the ensemble classification algorithm Random Forest [8] to develop the fraud classifiers by learning from the imbalanced and balanced SB datasets.

7.1 Hyperparameters Tuning

The tuning of the hyper-parameters of the classification algorithms is usually considered as an optimization problem [4] whose objective function controls the predictive performance. 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 utilize the Randomized Search Cross Validation (RSCV) in order to tune efficiently the parameters for SVM, Random Forest, and leave MLP-based ANN with default parameters for the sake of simplicity. The reason we have chosen RSCV 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 at once. After specifying the parameter values, Grid Search tries every possible combination, which is not time-efficient**.** RSCV, on the other hand, takes some distributions to sample from, and a maximum number of iterations to try. This allows us to focus the search on areas where the parameters perform the best. RSCV uses the *StratifiedKFold*, and we fix the number of folds to 5 and 10. RSCV has two major advantages over the exhaustive search (Site 12): 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 that has been proved efficient in many auction fraud detection problems, as auction data are non-linearly separable. Also, when we used RSCV, it provided us with RBF as the best kernel. Tables 9 and 10 present the SVM, Random Forest hyperparameters and some of the ANN parameters and their values.

Table 9: Hyperparameters used in SVM and Random Forest

Classifier	Hyper-Parameters	Description and Initialization
SVM	Misclassification Cost (C)	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. As a result, we assign the range [100, 500] to C.
	Gamma of RBF (γ)	If the free kernel parameter (γ) has large value then the area of influence for the support vectors can not 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 γ to [0.1, 10].
Random Forest	Number of Estimators (n_estimators)	It represents the number of trees to create 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 this decreases the diversity of the individual tree. The default value is Auto in sci-kit learn for max_features. RSCV 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 the maximum number of levels for each decision tree. It reduces complexity in the learned model and lowers over-fitting. The range we give for max_depth is [10, 110].

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

Table 10: Parameters used in ANN with MLP classifier

Classifier	Parameter	Description and Initialization
ANN with MLP classifier	Hidden Layer Sizes (hidden_layer_sizes)	This passes a tuple consisting of neuron number for each layer of the MLP. For the sake of simplicity, we choose 3 layers with the same number (30) of neurons.
	Activation	For hidden layer, we use rectified linear unit function as activation function.
	Solver	As the weight optimization solver, we use adam, a stochastic gradient-based optimizer.
	Learning Rate Initialization (learning_rate_init)	It regulates step size while updating the weights. It is used only with the adam solver. We set learning_rate_init to 0.001.
	Learning Rate (learning_rate)	Learning rate schedules the weight updates. In our case, we use learning_rate equals to constant given by learning_rate_init.
	Maximum number of Iterations (max_iter)	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.

7.2 Performance Metrics

In the fraud detection area, we put more emphasis on the suspicious class as it has a much higher misclassification cost. First, we focus on the detection rates of suspicious bidders by using the two metrics *Reca*ll and *Precision*. We focus 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 in 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 utilize the metric called *Area Under the Receiver Operating Characteristic Curve* (*AUROC*) representing the likelihood of a model distinguishing observations from the two classes. Next, we evaluate the misclassification rates of the fraud class based on *False Negative Rate* (FNR) and *False Positive Rate* (FPR) where FNR denotes the percentage of suspicious bidders incorrectly identified, and FPR is the percentage of normal bidders incorrectly classified**.**

8 Classification with Balanced SB Datasets

In Tables 11, 12 and 13, we provide the performance results obtained with SVM, Random Forest and ANN with MLP classifier. We develop in total 15 classification models (with 5 and 10-fold RSCV) for the three classifiers using the five sampling techniques.

Table 11: SVM performance using sampled SB datasets

Sampling Method	CV	C	γ	Precision	Recall	F-Measure	AUROC
SMOTE	5	269.97	3.08	0.95918	0.97295	0.96577	0.96545
	10	443.27	1.14	0.96579	0.97366	0.96464	0.967456
SMOTE-ENN	5	266.09	1.59	0.98429	0.98154	0.98272	0.98277
	10	344.65	3.08	0.98215	0.92960	0.95811	0.95980
SMOTE-TomekLink	5	437.50	3.36	0.97731	0.88006	0.93578	0.93957
	10	437.98	1.94	0.98557	0.96404	0.97955	0.97276
NearMiss	5	221.27	1.44	0.79856	0.86046	0.82835	0.81547
	10	396.98	0.64	0.85106	0.88235	0.86642	0.84987
Cluster Centroids	5	232.49	0.24	0.800000	0.89922	0.84671	0.83075
	10	175.60	1.01	0.74418	0.80000	0.77108	0.77404

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

Table 12: Random Forest performance using sampled SB datasets

Sampling Method	CV	n_estimators	max_features	max_depth	Precision	Recall	F-measure	AUROC
SMOTE	5	148	Log2	77	0.97410	0.95352	0.96370	0.96414
	10	344	Sqrt	63	0.97687	0.95211	0.96433	0.96484
SMOTE-ENN	5	467	Auto	48	0.98649	0.97198	0.97918	0.97803
	10	442	Sqrt	79	0.98578	0.97128	0.97848	0.97726
SMOTE-TomekLink	5	442	Sqrt	79	0.97900	0.96549	0.97220	0.97197
	10	206	Sqrt	46	0.97973	0.96756	0.97361	0.97336
NearMiss	5	467	Auto	48	0.88749	0.85542	0.87116	0.86690
	10	148	Log2	77	0.87974	0.83734	0.85802	0.85448
ClusterCentroids	5	148	Log2	77	0.91447	0.83734	0.87421	0.87475
	10	459	Log2	10	0.92207	0.85542	0.88749	0.88717

Table 13: MLP-based ANN performance using sampled SB datasets

Sampling Method	Precision	Recall	F-measure	AUROC
SMOTE	0.96548	0.95450	0.95996	0.96007
SMOTE-ENN	0.97467	0.97552	0.97509	0.97254
SMOTE-TomekLink	0.96016	0.97169	0.96016	0.97169
NearMiss	0.85826	0.82575	0.84169	0.83724
ClusterCentroids	0.84172	0.88636	0.86346	0.85074

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

To sum up, for every fraud classifier, SMOTE-ENN outperforms the other four sampling techniques. This is not surprising that SMOTE-ENN is superior since it has already been shown in [28] 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 14, SVM returns the best Recall value (0.98154) than 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, so we give more 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 14: Comparison between best SB classifiers using sampled datasets

Classifier	Detection Rate		Misclassification Rate	
	Recall	AUROC	FNR	FPR
SVM	0.98154	0.98277	0.01846	0.01571
Random Forest	0.97198	0.97803	0.02802	0.01351
ANN with MLP Classifier	0.97552	0.97254	0.02448	0.02533

9 Comparison on Balanced and Imbalanced Datasets

Table 15 exposes the evaluation of SVM, Random Forest and ANN when using the original imbalanced SB dataset given in Table 7. As we can see Precisions are satisfactory except Precision 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),

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

deteriorated very badly. For the non-sampled SB dataset, the best overall performance is obtained with MLP-based ANN classifier.

Table 15: Classification with the original imbalanced dataset

Classifier	CV	Precision	Recall	F-measure	AUROC
SVM	5	0.90000	0.06338	0.11842	0.53134
	10	0.68269	0.50000	0.57723	0.73853
Random Forest	5	0.96629	0.60563	0.74458	0.80177
	10	0.97802	0.62676	0.76394	0.81268
ANN with MLP Classifier		0.84042	0.67521	0.74881	0.83107

The following table exposes the best performance results for each of the three classifiers with and without using the sampling techniques.

Table 16: Comparison of the classification results with and without data sampling

Best Classifier	Sampling	CV	Precision	Recall	F-measure	AUROC
SVM	Yes	5 (SMOTE-ENN)	0.98429	0.98154	0.98272	0.98277
	No	10	0.68269	0.50000	0.57723	0.73853
Random Forest	Yes	5 (SMOTE-ENN)	0.98649	0.97198	0.97918	0.97803
	No	10	0.97802	0.62676	0.76394	0.81268
MLP-based ANN	Yes (SMOTE-ENN)		0.97467	0.97552	0.97509	0.97254
	No		0.84042	0.67521	0.74881	0.83107

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. And finally, for MLP-based ANN, SMOTE-ENN has again 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 Table 17, we compare between the best classifiers with and without sampling the SB dataset. The ANN classifier provides a Recall of 0.67521, which means almost 32.479% of suspicious bidders have been labeled as *normal* wrongly. On the other hand, after using sampling with SVM, we have a Recall 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. Also, in Table 17, we include the Log-Loss function to measure the accuracy of the two fraud classifiers. A perfect classification model has a Log-Loss of 0. An increased Log-Loss value means that the predicted labels are diverging from the actual labels. Therefore, minimizing the Log-Loss value corresponds to maximizing the prediction of classifiers. SVM classifier with the sampled SB dataset returns a very low value (0.07903) for the Log-Loss function, which reflects that the predicted labels diverged very little from the actual labels. On the other hand, ANN with MLP classifier obtains quite high log-Loss value (1.39247).

To conclude, Tables 16 and 17 demonstrate that sampling techniques greatly improve the predictive performance of classification models.

Table 17: Best classification results with and without data sampling

Best Classifier	Precision	Recall	F-measure	AUROC	Log-Loss function
SVM (with sampling)	0.98429	0.98154	0.98272	0.98277	0.07903
ANN with MLP Classifier (without sampling)	0.84042	0.67521	0.74881	0.83107	1.39247

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 15 / ISSUE 1 / JANUARY 2020 / 81-98

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

10 Conclusion and Future Work

Online auctions have become an ever-growing industry due to the increasing need in numerous domains. 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 user trust, online auction companies, such as Trade Me and eBay, must take measures to ensure that their sites are as fair as possible. One way of doing this is to implement systems capable of detecting SB activities before payments are processed. These systems will prevent financial losses for genuine buyers. There have been few studies on SB, primarily due to the difficulty associated with producing the required training data. Nonetheless, it has become immensely important to develop monitoring systems that are capable of tracking SB activities. To this end, we developed an efficient SB classification model based on real SB data obtained from auctions of eBay. First, we applied a robust hierarchical clustering technique to partition the training data into clusters of bidders with similar behavior. Second, we utilized a semi-automated approach to label each cluster and its bidders as either normal or suspicious. Next, to solve the imbalanced learning problem, we applied advanced sampling methods to our SB dataset. We develop the fraud classifiers and learn from the balanced SB datasets based on two standard classification algorithms, SVM and MLP-based ANN, as well as on an ensemble classification algorithm, Random Forest. From the experimental results, we highlight the following findings:

- The hybrid method, SMOTE-ENN, is the highest performing among all of the data sampling methods (under, over, and hybrid) across all the classifiers.
- SVM displayed the highest accuracy in detecting suspicious bidders, with a Recall of 98.154%; in comparison, Random Forest had a Recall of 97.198%, while MLP-based ANN had a Recall of 97.509%.
- When imbalanced data was used, all of the SB classifiers returned poor performance except for Precision. SVM provided the worst Recall of 0.06338.
- After data sampling, all three classifiers improved their performance. SVM improved considerably, with gains of 30.160%, 48.154%, 40.549%, and 24.424% in Precision, Recall, F-measure, and AUROC, respectively. Furthermore, Random Forest improved by 0.847%, 34.519%, 21.524%, and 16.535%, while ANN improved by 13.425%, 30.031%, 22.628% and 14.147%.
- All performance metrics were significantly improved with SVM using sampling in comparison to ANN without sampling.

In summary, this study has demonstrated that data-sampling techniques considerably improve the predictive performance of classification models, and that the combination of SVM and SMOTE-ENN provides the most satisfactory detection and misclassifications rates. Indeed, as previous empirical studies have shown, the classification accuracy can be maximized by combining data sampling with SVM [28].

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 problem, our fraud classifier maybe deployed on multiple autonomous agents. More precisely, for each new auction, an agent is created dynamically to classify the bidders, and then destroyed once the auction is completed or cancelled as done in [21].
- Develop a SB classifier that is capable to evolve constantly with new bidding data and trends. For this purpose, we will develop an adaptive fraud classifier based on the incremental and decremental learning with SVM. We select SVM as it returned the best classification accuracy in our study.

Websites List

Site 1: Internet Crime Report in 2015 | Internet Crime Complaint Center
https://pdf.ic3.gov/2015_IC3Report.pdf

Site 2: The Statistics Portal | Statista
https://www.statista.com

Site 3: Does eBay effectively fight with Shill Bidding? | eCommerce Bytes
https://www.ecommercebytes.com/C/abblog/blog.pl?/pl/2010/7/1278465003.html

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

Site 4: Shill bidding policy | eBay Customer Service
https://www.ebay.com/help/policies/selling-policies/selling-practices-policy/shill-bidding-policy?id=4353

Site 5: Safe buying advice | Trade Me
https://help.trademe.co.nz/hc/en-us/articles/360007275072

Site 6: Shill bidders - not welcome at Trade Me | Trust & Safety
https://www.trademe.co.nz/trust-safety/2012/9/29/shill-bidding

Site 7: The nation's leading online real estate marketplace | Auction.com
https://www.auction.com/

Site 8: Lawsuit accuses Auction.com of using shill bidder | New York Post
https://nypost.com/2014/12/25/lawsuit-targets-googles-auction-com/

Site 9: Auctioneer Gets Nearly 5 Years for Shill Bidding | eCommerce Bytes
https://www.ecommercebytes.com/2016/02/09/auctioneer-gets-nearly-5-years-shill-bidding/

Site 10: There may be a smoking gun in the lawsuit that accuses banks of rigging the bond market | New York Post
https://nypost.com/2017/11/16/wall-st-bankers-secretly-used-chat-rooms-to-rig-treasury-bond-trades-suit/

Site 11: Resampling strategies for imbalanced datasets | Kaggle
https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets

Site 12: Tuning the hyper-parameters of an estimator | Scikit-Learn
http://scikit-learn.org/stable/modules/grid_search.html

References

[1] A. Ali, S. Shamsuddin and A. Ralescu, Classification with class imbalance problem: A review, International Journal of Advances in Soft Computing and its Applications, vol. 7, no. 3, pp. 176-204, 2015.
[2] A. Alzahrani and S. Sadaouii. (2018, June) Scraping and preprocessing commercial auction data for fraud classification. Cornell University. [Online]. Available: https://arxiv.org/abs/1806.00656
[3] B. Arora, Exploring and analyzing internet crimes and their behaviours, Perspectives in Science, vol. 8, pp. 540-542, 2016.
[4] J.S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, 2011, Algorithms for hyper-parameter optimization, Advances in Neural Information Processing Systems, vol. 10, no. 4, pp. 2546-2554, 2011.
[5] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, Efficient agglomerative hierarchical clustering, Expert Systems with Applications, vol. 42, no. 5, pp. 2785-2797, 2015.
[6] E. Carneiro, L. Dias, A. Cunha, and L. Mialaret, Cluster analysis and artificial neural networks: A case study in credit card fraud detection, in Proceedings IEEE International Conference on Information Technology-New Generations (ITNG), USA, 2017, pp. 122-126.
[7] Y. Chen, H. Chu, J. Wu, N. Tsembel, and Y. Shen, A case study on attitude towards online auction use applying quantile regression analysis, Journal of Total Quality Management & Business Excellence, vol. 25, no. 3, pp. 1-21, 2017.
[8] M. Denil, D. Matheson and N. Freitas, Narrowing the gap: random forests in theory and in practice, in Proceedings International Conference on Machine Learning (ICML), China, 2014, pp. 665-673.
[9] S. Ding, H. Li, C. Su, J. Yu, and F. Jin, Evolutionary artificial neural networks: A review, Artificial Intelligence Review, vol. 39, no. 3, pp. 251-260, 2013.
[10] F. Dong, S. Shatz and H. Xu, Combating online in-auction fraud: clues, techniques and challenges, Computer Science Review, vol. 3, no. 4, pp. 245-258, 2009.
[11] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Zomaya, S. Foufou, and A. Bouras, A survey of clustering algorithms for big data: taxonomy and empirical analysis, IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 3, pp. 267-279, 2014.
[12] A. Fernandez, S. Garcia, F. Herrera, and N. Chawla, SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary, Journal of Artificial Intelligence Research, vol. 61, pp. 863-905, 2018.
[13] B. Ford, H. Xu and I. Valova, A real-time self-adaptive classifier for identifying suspicious bidders in online auctions, The Computer Journal, vol. 56, no. 5, pp. 646-663, 2012.
[14] S. Ganguly and S. Sadaoui, Online detection of shill bidding fraud based on machine learning techniques, in Proceedings International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, (IEA-AIE), Canada, 2018, pp. 303-314.
[15] S. Ganguly and S. Sadaoui, Classification of imbalanced auction fraud data, in Proceedings Canadian Conference on Artificial Intelligence, (CAI), Canada, 2017, pp. 84-89.
[16] P. Gupta and A. Mudra, Online in-auction fraud detection using online hybrid model, in Proceedings IEEE International Conference on Computing, Communication & Automation (ICCCA), India, 2015, pp. 901-907.

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762020000100107

[17] O. Loyola-González, J. Martínez-Trinidad, J. Carrasco-Ochoa, and M. García-Borroto, Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases, Neurocomputing, vol. 175, pp. 935-947, 2016.
[18] K.M.A. Patel and P. Thakral, The best clustering algorithms in data mining, in Proceedings IEEE International Conference on Communication and Signal Processing (ICCSP), India, 2016, pp. 2042-2046.
[19] M. Rahman and D. Davis, Cluster based under-sampling for unbalanced Cardiovascular data, in Proceedings of the World Congress on Engineering (WCE), UK, 2013, pp. 3-5.
[20] A. S. Sabau, Survey of clustering based financial fraud detection research, Informatica Economică, vol. 16, no. 1, pp. 110-122, 2012.
[21] S. Sadaoui and X. Wang, A dynamic stage-based fraud monitoring framework of multiple live auctions, Applied Intelligence, vol. 46, no. 1, pp. 197-213, 2017.
[22] C. Sun, Y. Chiu and M. Hsu, The determinants of price in online auctions: more evidence from quantile regression, Bulletin of Economic Research, vol. 68, no. 3, pp. 268-286, 2016.
[23] T. Thinsungnoena, K. Nuntawut, P. Durongdumronchaib, K. Kerdprasopb, and N. Kerdprasopb, The Clustering Validity with Silhouette and Sum of Squared Errors, vol. 3, pp. 44-51, 2015.
[24] S. Tsang, Y. Koh, G. Dobbie, and S. Alam, Detecting online auction shilling frauds using supervised learning, Expert Systems with Applications, vol. 41, no. 6, pp. 3027-3040, 2014.
[25] T. Yoshida and H. Ohwada, Shill bidder detection for online auctions, in Proceedings Pacific Rim International Conference on Artificial Intelligence, Berlin, Heidelberg, 2010, pp. 351-358.
[26] D. Zafeiris, S. Rutella and G. Ball, An artificial neural network integrated pipeline for biomarker discovery using Alzheimer's disease as a case study, Computational and Structural Biotechnology Journal, vol. 16, pp. 77-87, 2018.
[27] A. Zambelli, A data-driven approach to estimating the number of clusters in hierarchical clustering, F1000Research, vol. 5, pp. 411-423, 2016.
[28] B. Zhu, B. Baesens and S. vanden Broucke, An empirical comparison of techniques for the class imbalance problem in churn prediction, Information Sciences, vol. 408, pp. 84-99, 2017.