Identifying Suspicious Bidders Utilizing Hierarchical Clustering and Decision Trees*

Benjamin J. Ford, Haiping Xu, and Iren Valova
Computer and Information Science Department
University of Massachusetts Dartmouth, North Dartmouth, MA 02747, USA
E-mail: {u_bford, hxu, ivalova}@umassd.edu

Abstract - *Identifying bidders with suspicious bidding activities related to possible online auction fraud is a difficult task due to a large number of users participating in online auctions. In order to reduce the number of users to be investigated, we examine observable features of a bidder's behavior, and utilize a hierarchical clustering technique to divide a collection of bidders into normal and deviant groups. Based on the clustering results, we generate a decision tree that can be used to efficiently characterize new bidders as normal, suspicious, or highly suspicious. To illustrate the effectiveness of our proposed approach, we collected real auction datasets from online auctions, and used 3-fold validation approach to show that the error rates of the generated decision trees are reasonably low.*

Keywords: Online auctions, suspicious bidder, shilling behavior, hierarchical clustering, decision tree.

1 Introduction

Shill bidding is a type of auction fraud, which refers to the practice of sellers using a faked bidder account or asking another bidder to place bids on their auctions for the purpose of raising the final price [1]. Sellers typically do this through accomplices or by creating fake bidder accounts – an easy task in an anonymous environment such as the Internet. Shill bidding is unique in that it is very difficult to detect. Unlike blatantly obvious forms of other auction fraud, such as non-delivery fraud, shill bidding typically goes undetected by those victimized, especially those who do not know how to recognize the signs of shill bidding that may look like normal bidding activities.

In this paper, we present a series of attributes to describe suspicious bidding activities related to shilling behavior in online auctions. Once a set of bidders from a dataset have been characterized using these attributes, we can utilize hierarchical clustering to identify suspicious groups. As observed in [2], online auction participants belong to heterogeneous groups based on their bidding behavior. However, current literature focuses primarily on identifying groups of bidding behavior based on legitimate bidding intentions. Thus, there is a pressing need to design an effective method to identify suspicious bidders with illegitimate bidding intentions for efficient detection of shill bidders in online auctions.

Furthermore, the clustering results, which are labeled as normal, suspicious or highly suspicious, can be used as a training dataset to create a decision tree. With the decision tree, we can efficiently classify new bidders in an online auction immediately following its closure. If a new bidder is classified as suspicious or highly suspicious, we can use existing verification techniques, such as Dempster-Shafer (D-S) theory [3], to verify shill bidders. Note that existing approaches, such as D-S theory, are not efficient for analyzing large datasets. By efficiently identifying suspicious bidders in online auctions, our approach strongly complements existing techniques for shill detection, which suffer from time inefficiency.

Previous work employed various data mining techniques to categorize groups of bidders based on their bidding behavior. Bapna, et al. utilized *k*-means clustering to generate five distinct groups of bidding behaviors for Yankee Auctions [2]. They observed that users can improve the execution of their bidding strategies over time by adopting agent bidding to lower their bidding costs. Shah, et al. analyzed collected auction data from eBay to generate four distinct groups of bidding behavior [4]. The analysis revealed that there are certain bidding behaviors that appear frequently in online auctions. Hou and Rego used hierarchical clustering to generate four distinct groups of bidding behaviors for standard eBay auctions, namely goal-driven bidders, experiential bidders, playful bidders, and opportunistic bidders [5]. They concluded that online bidders are a heterogeneous group rather than a homogenous one. Although the above approaches look closely related to our proposed approach, they focus on creating clusters based on the assumption that bidders are honest ones with no malicious intentions. Thus, the clusters generated using these approaches reflect that bidders are all normal, even though there is significant evidence for shilling behavior. Unlike these approaches, we attempt to uncover suspicious bidders in online auctions. Once suspicious bidders are identified, we may use existing approaches such as D-S theory [3] to verify shill bidders.

* This material is based upon work supported by the U.S. National Science Foundation under grant numbers CNS-0715648 and CNS-0715657.

2 Identifying suspicious bidders

A user's bidding behavior in an online auction can be described using a variety of measurable attributes or features. According to statistics we observed for typical datasets, the majority of bidders exhibit the same behavior. For example, in a collected dataset for "*Used Playstation 3*" auctions, 57% of bidders only bid once during an auction, 19% bid twice, and 8% bid three times. Furthermore, 11% of bidders only bid in the early hours of an auction, whereas 46% of bidders only bid in the final hours of an auction. In addition to these behaviors, we specifically define attributes that describe behaviors related to shill bidding. For example, *bid unmasking* refers to the practice of placing many small bids to uncover the true valuation of the current high bidder. An attribute that measures the average time span between a user's bids is useful for identifying bid unmaskers. By using hierarchical clustering to organize bidders into groups, bidders can be grouped with those who exhibit similar behavior. Since most bidders exhibit the same behavior, bidders that do not appear in the large groups deviate from the norm and are possibly suspicious. Further investigation into the characteristics of a group of bidders may reveal whether or not those bidders are potential shill suspects.

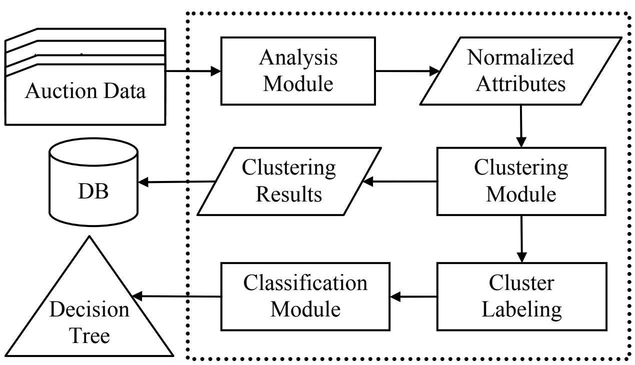

Figure 1. A framework for identifying suspicious bidders

As shown in Figure 1, using our framework for identifying suspicious bidders, we first retrieve real auction data and calculate values for various attributes of interest. Next, we normalize, weight, and then process the values using a hierarchical clustering algorithm. Due to the data distribution, outliers may exist outside of the main clusters. Since outliers deviate from the norm, by definition, they should also be marked as suspicious. Finally, we label these clustering results, and use them to generate a decision tree for classification of new bidders.

3 Bidder attributes

Each bidder possesses various attributes that can be measured directly from the bidding history of an auction. Attributes such as the number of bids placed, the average bid increment, and average time between a bidder's own bids are common to all bidders. Bidders that possess extraordinary values for these attributes are possibly suspicious bidders. In addition, we also choose other types of attributes that may provide evidence of suspicious behavior. We categorize the attributes into three groups, namely *user attributes*, *stage attributes*, and *auction attributes*. We now give a few examples of attributes from each group in the following sections.

3.1 User attributes

User attributes are specific for each bidder, which examine immutable information about a user. Two examples of such attributes are described as follows.

Elapsed Time before First Bid (ETFB) is the time that elapses from the start of the auction to a bidder's first bid. A high value of *ETFB* indicates that the bidder started participating late in the auction, whereas a small value indicates that the bidder participated very early in the auction. As noted in [6], shill bidders tend to place bids in the earlier portions of auctions due to the low risk of winning. Although it is possible for a normal bidder to place bids near the start of the auction, placing bids extremely close to the start of the auction implies the bidder's possible prior knowledge about the auction. Note that a normal bidder is less likely to notice a newly created auction and participate very early in that auction.

Bidder Feedback Rating (BFR) is useful in describing a bidder's experience level and established trustworthiness [3, 5]. However, we should also notice the potential for fabricating feedback rating through fraudulent bidding rings. Thus, a feedback rating should not be considered as a primary factor for describing the trustworthiness or experience level of a user.

3.2 Stage attributes

The mutable attributes for each bidder are specified as stage attributes. Following the definitions in [6], we divide the auction duration into three stages, namely *early stage*, *middle stage* and *final stage*. The early stage refers to the first quarter of the auction duration; the middle stage refers to [0.25, 0.9] of the auction duration; and the final stage refers to the last 10% of the auction duration. Thus, each stage attribute has three values that correspond to the three stages, respectively. Three examples of stage attributes are described as follows.

Average Competitive Interval (ACI) refers to the amount that a bidder outbids the current high bidder when placing a bid in a certain auction stage. For example, if the current high bid is \$30.00 and a bidder places a new bid for \$40.00, the bidder's competitive interval is \$10.00. A bidder's *ACI* is the average of his outbid amounts for an auction during the auction stage under consideration.

A very high value for this attribute may indicate suspicious behavior. Although a high value may be due to a bidder's significant interest in an item, this is unlikely for auction items that are in relatively high supply. Note that there is little logical reason to have an *ACI* that is significantly higher than thousands of other bidders.

Furthermore, a high *ACI* value at the early stage or middle stage is more suspicious than a high *ACI* value at the final stage because most shill bidders would not risk placing a significantly high bid in the final stage that results in a high possibility of winning the auction. The *ACI* value can be calculated as follows.

$$ACI = \frac{\sum_{i=1}^{n} X_i - Y_i}{n} \tag{1}$$

where X_i is the user's new bid, Y_i is the previous bid of X_i, and n is the total number of bids placed by the user in this stage.

Average Time between User Bids (ATUB) refers to the average time that elapses between two bids placed by the same bidder. Since we try to identify aggressive bidders with this attribute, we define *ATUB* as the inverse of the average elapsed time between bids as follows.

$$ATUB = \begin{cases} \dfrac{n}{\sum_{i=2}^{n} T_i - T_{i-1}} & if \quad n > 1 \\ 0 & otherwise \end{cases} \tag{2}$$

where T_i is the time of user's bid, and n is the total number of bids placed by the user in this stage. Note that if n equals to 0 or 1, it returns 0 because the calculation of *ATUB* requires at least two bids.

A large value of *ATUB* indicates the bidder is actively participating in the auction, placing bids soon after they are outbid; while a small value of *ATUB* implies that the bidder is not participating heavily in the auction and is cautious before placing a new bid. A large value of *ATUB* in the early or middle stage indicates a suspicious bidder's desire to raise the auction price and uncover the true valuations of other bidders, otherwise known as bid unmasking. For example, the proxy bidding system used at eBay facilitates the practice of bid unmasking because a new bid is immediately outbid if another user's maximum bid is higher than the new bid. Therefore, a shill bidder can bait a proxy bidding system with small bids over the current high bid, and let the auction price quickly climb to other bidders' true valuation [7].

On the other hand, a large value of *ATUB* in the final stage indicates a bidder's strong desire to win the auction, because a bidder participating heavily in the final stage will likely win. In contrast, a shill bidder usually does not have a large value of *ATUB* in the final stage due to the risk of winning the auction. Thus, the stage at which the value occurs has a significant impact on the analysis.

Number of Bids (NB) refers to the number of bids placed in a particular stage of the auction. A large value of *NB* at the early stage typically indicates a suspicious bidder's desire to raise the price of the auction. A large value of *NB* at the middle stage could indicate suspicious behavior but not as strongly as a large value at the early stage. A suspicious bidder may attempt to uncover the true valuations of other bidders before the final stage, and a normal bidder may participate in a bid fight in the middle stage. A large value of *NB* at the final stage typically indicates a strong desire to win the auction. Due to the risk of winning the item, suspicious bidders typically do not participate at all in the final stage of an auction.

3.3 Auction attributes

Various properties of an auction may also influence a bidder's decision to participate, and particular values of auction attributes may cast more suspicion on the trustworthiness of an auction as well as the likelihood of shill bidding taking place. We now give two examples of auction attributes as follows.

Auction Starting Price (ASP) may influence a bidder's decision to participate in the auction. A high starting price can deter many bidders from participating in the auction, especially if they are bargain hunters. A low starting price combined with a very early bidder may indicate a seller's attempt to avoid the additional fees associated with setting a high starting price.

Reference [3] discussed the effects of the starting price on the possibility of an auction involving shills. It was concluded that an auction with a low starting price is more likely to involve reserve price shilling. Instead of setting a reserve price and paying additional fees, sellers can set a low starting price, place shill bids to raise the price up to an acceptable price, and profit. Thus, bidders that participate in auctions with a low starting price are more likely to be suspicious.

Seller Feedback Rating (SFR) is an important factor in determining the behavior of bidders in an auction. Some bidders are much more likely to bid in an auction if the seller has significant positive feedback [8, 9]. However, as described by other researchers, *SFR* can also be fabricated through fraudulent bidding rings, and thus cannot be entirely trusted to be accurate [1].

4 Hierarchical clustering

We utilize hierarchical clustering techniques to cluster collected data points. Hierarchical clustering is known to have significant advantages over flat clustering techniques such as *k*-means [10]. Flat clustering algorithms require the number of clusters to be defined prior to execution, which significantly affects the clustering results. However, determining the proper number of clusters is not a trivial and arbitrary task. Flat clustering algorithms may also lead to nondeterministic results. For example, given an input, the algorithm may generate numerous different sets of results, but it is impossible to know whether the set of results is complete or the optimal result has been generated. Thus, justifying a set of results generated using a flat algorithm is very difficult. Although hierarchical clustering algorithms are at least quadratic in time complexity, this does not affect our decision since the cluster analysis is performed offline and not constrained by time [10].

An important aspect of clustering is the similarity measure, which determines when elements shall be added into a cluster. At a given point in a clustering process, two clusters deemed to be the most similar are the ones to be combined into a single cluster. For our approach, we use *centroid clustering* as our similarity measure. Centroid clustering is known to be not as heavily affected by outliers as single-link or complete-link similarity measures [10]. Since we are certain to have outliers to be identified from our collected dataset, we require the similarity measure to possess this quality.

The similarity between two clusters is determined by the similarity between the centers (centroids) of the two clusters (as shown in Equation 3). The centroid of a cluster is equivalent to the vector average of the cluster's members (as shown in Equation 4). In the following equations, $\vec{x}$ refers to the vector average of cluster C_a's members, and $\vec{y}$ refers to the vector average of cluster C_b's members.

$$SIM(C_a, C_b) = \vec{x} \cdot \vec{y} = \sum_{i=1}^{n} \vec{x}_i * \vec{y}_i \tag{3}$$

$$\vec{x} = \frac{1}{N_A}\sum_{i=1}^{N_A} a_i \qquad \vec{y} = \frac{1}{N_B}\sum_{i=1}^{N_B} b_i \tag{4}$$

In addition to the similarity measure, it is also important to specify a minimum similarity cutoff. The cutoff value determines when the clustering process should terminate. If two clusters' similarity does not exceed this cutoff value, they are not combined. Note that if a cutoff point is not specified, the clustering algorithm eventually outputs a single cluster containing all of the elements.

To fully utilize the available domain knowledge, we make use of weighted attribute values. Although the values for a dataset are normalized after analysis, we require that certain attributes have more emphasis than others. For instance, bidders with low feedback rating might be simply new users, who are not necessary suspicious; while bidders who place a lot of bids in the early stage are more likely to be suspicious bidders. Thus, certain attributes should be more important in determining cluster membership than others. By weighting the normalized values, two effects are realized. First, since the other values are either weighted from 0 to 1 or from -0.5 to +0.5, an attribute with a weight of 2 will range from 0 to 2 or -1 to +1. When utilized in the similarity measure calculations, these values will skew the similarity calculation in their direction, which is our objective. Secondly, weights also disperse values within a particular attribute. If two bidders were originally separated by a distance of 0.25 on their *NB (early)* value, a weight of 2 will cause them to be separated by a distance of 0.5 on that value. As a result, the two bidders are more likely to be in different clusters than without the weighted value. Because we want high values for particular attributes to stand out more so than others, applying weights to the normalized values allows us to accomplish this.

Once the auction data has been analyzed, normalized, and weighted, it is passed to a cluster generation algorithm as parameter *dPoints*. The cluster generation algorithm is defined recursively as follows.

Algorithm 1: Cluster Generation

```
generateClusters (DataSet dPoints,
                  ClusterSet clusters, double minSimilarity)
  if size(clusters) == 0   // initially, there are zero clusters
     for each element e in dPoints
        create a new cluster c for e and add c into clusters
     return generateClusters(dPoints,clusters,minSimilarity)
  else if size(clusters) == 1  // there is only one cluster
     return clusters
  else   // there are at least two clusters in set clusters
     initialize maxSimilarity to 0
     initialize mergeClusters to false
     for each pair of clusters c and d in clusters
        calculate the similarity between c and d
        if similarity > maxSimilarity &&
                  similarity ≥ minSimilarity
           maxSimilarity = similarity
           set cluster1 and cluster2 to c and d, respectively
           set mergeClusters to true
     if mergeClusters == true
        merge cluster1 and cluster2 into a new cluster3
        replace cluster1 and cluster2 by cluster3 in clusters
        return generateClusters(dPoints,clusters,minSimilarity)
     else   // no more clusters can be merged
        return clusters
```

Note that the cluster set *clusters* initially contains no clusters, but as a starting point for the clustering process, the algorithm creates a set of clusters in which each cluster consists of a single data point. The basic idea of the hierarchical clustering approach is to repeatedly merge the two closest clusters until there is only one cluster left or the similarity measures of all pairs of clusters become less than a predefined minimum similarity *minSimilarity*.

The set of clusters generated using the cluster generation algorithm are merely numbered without any semantic meaning associated with membership in a particular cluster. In order to give each cluster and its members semantic meaning, we need to properly label the clusters. This is done by manually inspecting each cluster and assigning a label (i.e., *normal*, *suspicious*, or *highly suspicious*) based on its most prevalent features. In Section 6 (case study), we give an example to show how such labeling can be done.

5 Decision tree generation

After proper labeling, the generated clusters can be used as a training set for creating a decision tree. The decision tree can be used to classify new bidders in online auctions, i.e., to identify suspicious bidders immediately following the conclusion of an auction. If a suspicious bidder is detected, the corresponding account is put under further investigation for verification of shill bidding.

The decision tree generation requires a set of input attributes and an output attribute that determines a bidder's appropriate classification. When creating a decision tree, the dataset used as a training set determines the structure of the tree. Each node of the tree, including the root, corresponds to a set of examples from the training set. The leaf nodes in this tree contain a set of examples such that they all have the same value for the output attribute. A splitting function is used to determine which input attribute should be assigned to a non-leaf node. The chosen attribute determines the number and contents of the node's children. For example, if the input attribute *NB (early)* has five possible values and is chosen for the root node, the root node can have up to five children, each corresponding to a value for *NB (early)*. Each of these children contains a subset of the training set such that all the children's elements correspond to the value chosen for its parent's corresponding attribute.

We utilize the information gain ratio as our splitting function. Because most attributes are continuous values, the number of discrete values for most attributes can be quite large. Data binning is used to compensate for this difficulty. For attributes that contain non-integer values, such as *ETFB*, four bins are created. The four bins correspond to values equal to zero, less than or equal to $M/2$, greater than $M/2$ but less than M, and values equal to M, where M is the maximum value. This binning procedure is necessary because the information gain function is known to favor attributes with a large number of values for the attribute. For example, an attribute with thousands of values could be used to classify the entire training set, but if this attribute is a unique identifier, such as social security number, it is obvious that any new examples will not be properly mapped. This issue is referred to as *overfitting*. The information gain ratio function overcomes this favoritism. The equations to calculate the Shannon's entropy (E) as a criterion for selecting the most significant attribute, the information gain ($GAIN$) for choosing an attribute to split upon, and the gain ratio for choosing an attribute to split upon are given in Equation 5, 6, and 7, respectively. In Equation 5, X can be an input or output attribute for the current training set, and $P(v_i)$ refers to the probability that the attribute has the value of i, which represents a class number. In Equation 6, X refers to an input attribute that has been chosen for a split, Y refers to an output attribute, and $E(Y|X = v_i)$ refers to the information contained in the output attribute for the training set that results from choosing X for the split. In Equation 7, the gain ratio is calculated as a quotient of information gain over Shannon's entropy.

$$E(X) = \sum_{i=1}^{n} -P(v_i)*\log_2 P(v_i) \tag{5}$$

$$GAIN(X) = E(Y) - \sum_{i=1}^{n} P(X = v_i)*E(Y|\, X = v_i) \tag{6}$$

$$GAINRATIO(X) = \frac{GAIN(X)}{E(X)} \tag{7}$$

Based on the information gain ratio, a decision tree can be generated using the following recursive decision tree generation algorithm.

Algorithm 2: Decision Tree Generation

```
generateDT(ClusterSet clusters, DecisionTree tree,
                          int maxDepth, float minGain)
if no nodes in tree can be split || maxDepth is reached
   return tree
else       // split a node
   choose a node n that is not perfectly partitioned
   curSet = n.trainingSet
   remainingAttributes = n.remainingAttributes
  if there are no more remaining attributes
     mark n as being perfectly partitioned
     return generateDT(clusters,tree,maxDepth,minGain)
  else
     initialize maxRatio to 0, and splitNode to false
     for each attribute x in remainingAttributes
        calculate x's gain ratio r based on curSet
        if r > maxRatio && r >= minGain
           maxRatio = r; bestAttribute = x
           set splitNode to true
     if splitNode == false
        mark n as being perfectly partitioned
        return generateDT(clusters,tree,maxDepth,minGain)
     else  // splitNode == true
        partition curSet into new nodes based on bestAttribute
        link and add the new nodes into tree
        return generateDT(clusters,tree,maxDepth,minGain)
```

The decision tree generation algorithm first checks if any nodes can be split without violating the previously described constraints. If so, then it chooses a node that is eligible for splitting. If there are remaining input attributes, the information gain ratio is calculated for each of them. The attribute with the highest ratio is chosen as the attribute for node splitting if the highest ratio is no less than the predefined minimum information gain *minGain*. In this case, children nodes of the splitting node are created, and each child's training set contains the elements corresponding to their values for their parent's attribute. This process continues until no nodes can be split or the maximum depth has been reached.

Once a decision tree has been generated using a training set, a test set is chosen to evaluate the effectiveness of the decision tree. This process begins at the root node and ends at a leaf node. For each test element, its value for the root node's attribute is retrieved and compared to the various branches from the root node. The next node is obtained by following the branch with a matching value, and the element's value for that node's attribute is retrieved. This process repeats in a similar fashion until a leaf node is reached. Once a leaf node is reached, the value of the leaf node is used to classify the test set element as a member belonging to a certain class. When all test set elements have been classified, their classifications are compared to the labels assigned to them in the clustering process. If a decision tree's classification matches with its pre-assigned cluster label, the classification is said to be *correct*.

6 Case study

To demonstrate the feasibility of identifying suspicious bidders using hierarchical clustering and decision trees, we tested our methodology using real auction data from eBay.

The dataset we collected consists of completed eBay auctions for "*Used Playstation 3*." This dataset was collected over the course of 30 days, and the estimated price of the auctioned item is $220.00 at the time of data collection. The dataset is divided into two subsets, one with duration of 1 day and the other with duration of 7 days. Our observations regarding differences in bidding behavior based on the duration of an auction motivated this dataset division. The 1-day auction subset consists of 153 auctions with a total of 1,845 bidders, and the 7-day auction subset consists of 94 auctions with a total of 1,064 bidders. The differences in dataset sizes are due to the number of auctions available in the 30 day period.

Before the dataset can be processed for clustering, the attribute weights need to be specified. This is a subjective process based on the perceived importance of the attributes. The default weight for an attribute is 1. Certain attributes such as *NB (early, middle), ETFB, and ATUB (early, middle)* provide strong evidence that a bidder is suspicious, thus they are assigned a weight of 3. Attributes such as *ACI* provide evidence leaning toward a suspicious bidder but can also be attributed to normal behavior, thus *ACI (early, middle)* is assigned a weight of 2. All other attributes focus more on typically legitimate bidder behavior. For example, a bidder that places bids in the final stage is typically legitimate. As a result, such attributes are assigned the default weight of 1.

To validate our results, we utilize a 3-fold cross validation process. Each dataset is first divided into three equal portions – 2/3 of the dataset is used as a training set and 1/3 of the dataset is used a testing set. Then three complete sets of experiments are performed. The first experiment (called *fold* 1) consists of the first 1/3 of the dataset as the test data, the second experiment (called *fold* 2) consists of the second 1/3 of the dataset as the test data, and so on. The advantage of this method is that all data points are used for both training and validation, and each data point is used for validation exactly once.

We performed the clustering process twice for each fold: once on the training set, *A*, and another on the complete set $A \cup B$, where *B* corresponds to a test set. The clusters generated from subset *A* are used in generation of a decision tree. The clusters generated from $A \cup B$ are used to find the correct clusters for test set *B*. Once the clusters are generated for *A*, they are labeled and passed to the decision tree algorithm as a training set. Meanwhile, test set *B* is extracted from the complete set, labeled, and passed to the decision tree algorithm for evaluation.

Upon analyzing the clustering results for both the 1-day and 7-day datasets, we determined that a minimum similarity cutoff point of 86.9% led to the best results. For example, using cutoff point of 86.9%, the 1-day dataset is divided into 19 clusters and 6 outliers for the entire dataset and 16 clusters and 7 outliers for the first fold. Note that less restrictive cutoffs result in clusters that contained a mixture of normal bidders and suspicious bidders whereas more restrictive cutoffs result in multiple clusters that contained similar suspicious behavior.

Upon careful inspection of the cluster results, we discovered that the clusters can be mapped to one of the three classes: *normal*, *suspicious*, or *highly suspicious*. Furthermore, most of the highly suspicious activity can be directly compared to shilling behaviors described in [6]. As an example, Table 1 shows 16 clusters for 1-day fold 1 experiment. Since cluster 16 has the highest values for attribute *NB (early)*, it is labeled as highly suspicious. Cluster 13 has a moderate value for attribute *NB (early)*, and does not have high values for any other attributes; thus, it is labeled as suspicious. Cluster 4 has few bids placed in the middle stage of the auction, and it also has low values for all other attributes; thus it is labeled as normal. Once the clustering results have been generated and labeled, they are used to generate and test the decision trees. Due to

Table 1. Training data clusters for 1-day, fold 1

Cluster	Size	Class	Description
1	63%	Normal	Bids placed very late in auction (later middle stage or final stage).
2	<1%	Highly Suspicious	Very high bidding amounts in middle stage.
3	4%	Suspicious	Bids placed close together in middle stage. Possible bid unmasking.
4	9%	Normal	Few bids placed in the middle stage of auction.
5	8%	Normal	Similar to cluster 4, but bids placed later in the middle stage.
6	1%	Suspicious	Bids placed fairly early in auction.
7	<1%	Normal	Few bids placed in the middle stage of auction.
8	<1%	Highly Suspicious	Highest bid amounts in the middle stage.
9	1%	Suspicious	Bids placed close together in the middle stage. Possible bid unmasking.
10	<1%	Suspicious	Bids placed fairly early in auction and bids placed close together in the middle stage.
11	<1%	Highly Suspicious	Bids placed in quickest succession in the middle stage. Possible bid unmasking.
12	11%	Suspicious	Bids placed very early in auction (early stage).
13	<1%	Suspicious	Moderate number of bids in early stage.
14	<1%	Suspicious	Bids placed close together in early stage. Possible bid unmasking.
15	<1%	Highly Suspicious	Bids placed in quickest succession in the early stage. Possible bid unmasking.
16	<1%	Highly Suspicious	Highest number of bids in the early stage.

overfitting concerns, the maximum depth of the decision trees is set to 3, and the minimum information gain is set to 10%. Figure 2 shows a decision tree generated for the 1-day, fold 1 auction data set.

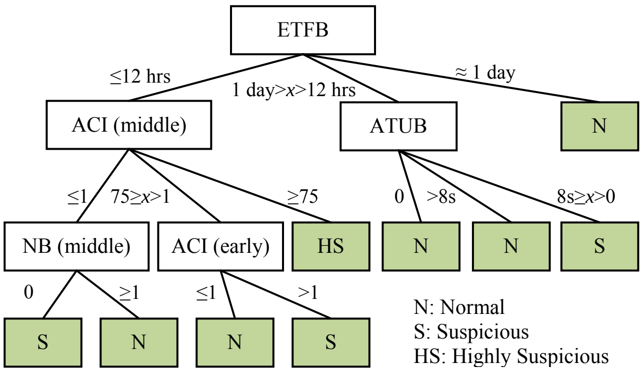

Figure 2. Decision tree for 1-day, fold 1

We now evaluate decision trees using the test sets according to the procedure described in Section 5. Table 2 shows the decision tree analysis results for the 1-day, fold 1 and 7-day, fold 2 experiments, which are the best results among the 3 folds for 1-day and 7-day, respectively.

Table 2. Decision tree analysis results

Experiment	Training Set	Test Set	Test Results
1-Day, Fold 1	81% Normal 18% Suspicious 1% Highly Suspicious	614 data points	94% Correct 6% Incorrect
7-Day, Fold 2	82% Normal 16% Suspicious 2% Highly Suspicious	354 data points	98% Correct 2% Incorrect

The error rates of the decision trees for all folds of the 1-day and 7-day auctions are illustrated in Figure 3. The experimental results show that error rates of the generated decision trees for classification are reasonably low.

Figure 3. Comparisons for decision tree error rates

7 Conclusions and future work

Due to the increasingly large population of bidders in online auctions, shilling behavior becomes more and more popular and difficult to detect. However, there is a lack of training data that can be used to create effective classifiers for shill detection purpose. In this paper, we introduced a new set of attributes to describe bidder behavior, which can accurately measure suspicious bidding activities that are related to shilling behavior. We used a hierarchical clustering algorithm and demonstrated that suspicious bidders can be successfully identified among large number of bidders. After proper labeling, we used the labeled clusters as a training set and created decision trees to efficiently classify new bidders. Our approach can be used in tandem with existing shill detection techniques [3, 6] to improve efficiency. For future work, we will investigate using neural networks for possibly more efficient classification and improved accuracy. We will also study for how to create stage-based classifiers, so suspicious bidders may be identified in real-time.

8 References

[1] F. Dong, S. M. Shatz, and H. Xu, "Combating Online In-Auction Fraud: Clues, Techniques and Challenges," *Computer Science Review (CSR)*, Vol. 3, No. 4, November 2009, pp. 245-258.

[2] R. Bapna, P. Goes and A. Gupta, "User Heterogeneity and its Impact on Electronic Auction Market Design: An Empirical Exploration," In *MIS Quarterly*, Vol. 28, No. 1, March 2004, pp. 21-43.

[3] F. Dong, S. M. Shatz, and H. Xu, "Reasoning Under Uncertainty for Shill Detection in Online Auctions Using Dempster-Shafer Theory," To appear in *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, Vol. 20, No. 7, November 2010.

[4] H. S. Shah, N. R. Joshi, A. Sureka, and P. R. Wurman, "Mining for Bidding Strategies on eBay," In *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 2003.

[5] J. Hou and C. Rego, "A Classification of Online Bidders in a Private Value Auction: Evidence from eBay," *International Journal of Electronic Marketing and Retailing*, Vol. 1, No. 4, 2007, pp. 322-338.

[6] H. Xu, C. K. Bates, and S. M. Shatz, "Real-Time Model Checking for Shill Detection in Live Online Auctions," In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP'09)*, July 2009, Las Vegas, Nevada, USA, pp. 134-140.

[7] J. Engelberg, J. Williams, "EBay's Proxy Bidding: A License to Shill," *Journal of Economic Behavior & Organization*, Vol. 72, No. 1, October 2009, pp. 509-526.

[8] D. Houser and J. Wooders, "Reputation in Auctions: Theory and Evidence from eBay," *Journal of Economics and Management Strategy*, Vol. 15, No. 2, Summer 2006, pp. 353-369.

[9] J. A. Livingston, "How Valuable is a Good Reputation? A Sample Selection Model of Internet Auctions," *The Review of Economics and Statistics*, Vol. 87, No. 3, 2005, pp. 453-465.

[10] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, England, 2009.