

Generating Realistic Online Auction Data

Sidney Tsang, Gillian Dobbie, and Yun Sing Koh

The University of Auckland

stsa027@aucklanduni.ac.nz, {gill,ykoh}@cs.auckland.ac.nz

Abstract. To combat online auction fraud, researchers have developed fraud detection and prevention methods. However, it is difficult to effectively evaluate these methods using commercial or synthetic auction data. For commercial data, it is not possible to accurately identify cases of fraud. For synthetic auction data, the conclusions drawn may not extend to the real world. The availability of realistic synthetic auction data, which models real auction data, will be invaluable for effective evaluation of fraud detection algorithms. We present an agent-based simulator that is capable of generating realistic English auction data. The agents and model are based on data collected from the TradeMe online auction site. We evaluate the generated data in two ways to show that it is similar to the TradeMe data. Evaluation of individual features show that correlation is greater than 0.9 for 8 of the 10 features, and evaluation using multiple features gives a median accuracy of 0.87.

1 Introduction

Over a three-month period in 2011, 37% of New Zealanders have purchased at least one item from an online auction site based in New Zealand called TradeMe [1]. The popularity of online auction sites such as TradeMe has made them a lucrative target for dishonest users to commit fraud. There have been several methods proposed by researchers to identify different types of online auction fraud [2–6]. These methods are evaluated using synthetic data, or using data collected from commercial sites such as eBay.

There are disadvantages to using both types of data. One of the drawbacks of synthetic auction data is that it may be dissimilar to data in the real world. Evaluation performed using this data will be less reliable, since results obtained may not extend to real auction data. Though it is possible to modify the synthetic data to simulate the real world, it is difficult to model bidding and selling behaviour because of the large range of possible actions. This is shown by Shah et al. [7], who found that 88% of users do not frequently use any of the five bidding strategies described: evaluate, skeptic, snipe, late bidder and unmask. The advantage of using synthetic data is that the instances and types of fraud can be controlled.

Evaluation using auction data from commercial sites may give results that are more representative of the real world than synthetic data. However, it is difficult and time consuming to manually classify data, in the case of fraud detection, into fraudulent and non-fraudulent users or auctions. For larger datasets, manual classification is infeasible. In addition, it is often time-consuming to gather and process data from auction sites, as

seen in [3] which used an elaborate system to gather auction data from eBay. Also, data may be selectively missing or removed by operators of the auction site, such as profiles of users that have been identified as fraudsters, or auctions that have been identified afterwards as fraudulent. This complicates the evaluations done using this data.

In this paper, we present an agent-based simulator of an online auction site where agents are modelled on users in real auction data. The resulting simulation can be used to quickly generate realistic auction data. We then validate our simulator by performing quantitative evaluations comparing the generated data to commercial TradeMe auction data, and show that the generated and commercial data shows a high degree of similarity.

The simulator can be easily extended in the future with additional agents to inject fraudulent behaviour. Since the simulator is modelled on a per-user basis, the synthetic data with injected fraud will be valuable for effective evaluation of fraud detection algorithms which often focus on per-user features.

2 Existing Work

Agent-based simulation and modelling has been used in a wide range of complex systems. Macal and North [8] list nine different areas in which agent-based modelling has been applied, including air traffic control, anthropology, and energy analysis. Macal and North propose possible reasons for the widespread use of agent-based modelling. They state that agent-based modelling allows assumptions used in the past for modelling complex systems to be relaxed, such as perfect market or homogeneous agents in economic markets. Collection of data at finer levels of granularity also supports individual-based simulations. We list three reasons that they presented that apply to our work here. First, previous work on formal auction analysis used assumptions such as perfect rationality, perfect market and homogeneous agents, which can be removed in agent-based models. Second, there exists precise bidding and auction information for modelling of individual users. Third, auction users are suitably modelled by agents since they share many similarities; they are both self-contained, autonomous, and have defined ways of interacting with other users.

One of the earliest works in agent-based simulation in auctions was done by Mizuta and Steiglitz [9], who modelled agents of two types: early bidders and snipers, and investigated the effects of the strategies on the probability of winning, auction price increment, and final auction price. More recently, agent-based modelling of bidders has been used to perform evaluations of fraud related algorithms [2, 6], or to investigate the effects of modifying different aspects of the model, such as changing the bid increment [10]. In both [2] and [6], the authors do not evaluate their models against reality.

In [10], Bapna et al. constructed a simulation model of Yankee auctions. They identified three broad types of bidders according to their bidding strategies: evaluators, participants, and opportunists, and created agents that used each strategy. The simulation model is validated by comparing the simulated revenue with the observed revenue from data collected from real auctions. Using this model, they altered the auction bid increment to optimise seller revenue.

We know of no previous work that attempts to accurately model bidder and seller behaviour in commercial online English auctions using agent-based methods without having previously classified bidders.

3 Data Generator Overview

Our data generator is an agent-based simulation of online English auctions, the auction type used in online auction sites such as eBay and TradeMe. For more details on the English auction and other auction types, see [11]. Auction data can be generated by running the simulation and recording the actions of all of the agents, such as auctions submitted and bids made. By carefully specifying the behaviour of the agents, we can generate synthetic auction data that closely approximates real auction data.

The decision to use user-level features instead of auction-level features is because in the future, we plan to inject fraud into the data and use it to evaluate fraud related algorithms which tend to use per-user features instead of per-auction features [2, 3, 6].

We start by giving a brief overview of the stages involved in the implementation of the generator:

Stage 1. Auction and User data is gathered from TradeMe in October 2011 using a web crawler, and stored in a database. A total of 93,179 users and 161,895 auctions were collected.

Stage 2. An agent-based simulator is built in Java with the following characteristics:

1. Each agent corresponds to a bidder or seller.
2. Agents communicate with an Auctioneer entity to submit bids or auctions. The Auctioneer also maintains the state of all current auctions and bids and ensures that they conform to the auction rules.

Stage 3. We define the rules that each auction and bid must follow, which are then enforced by the Auctioneer. E.g., auction starting prices must be greater than zero. These rules can be easily changed to model other auction sites, such as eBay.

Stage 4. We specify the behaviour of bidder-agents according to the bidding patterns seen in the TradeMe users. This is done by representing bidders in TradeMe as a set of features, and defining agent rules that gives similar values for those features. Examples of rules is given in Section 4.4.

Stage 5. Next we specify the rules for seller-agents. The two main parameters are rate of auction submission and auction starting price. We match the average rate and distribution of auction submission of the seller-agents to TradeMe users.

Stage 6. We evaluate the synthetic auction data generated using the bidder- and seller-agents by comparing them with the TradeMe user data we have collected. If the synthetic and real data are not sufficiently similar, we modify the agent behaviour, then re-evaluate.

3.1 Generated Auction Example

Table 1 gives an example of an auction generated by the generator. Table 1a lists the bids made for that auction. Table 1b gives information about the auction. *Bid Time*,

Start Time, and *End Time* represent time units since the beginning of the simulator execution, with each unit representing five minutes. *Duration* is the number of time units the auction runs for.

There are 8 bids made for this auction, and User 2285 wins the item for \$40.61 at time 2026. Note that *Duration* is less than the difference between the start and end times because the auction was extended due to a bid submission in the last 3 time units.

Table 1. Example of a generated auction

(a) Auction Bids				(b) Auction Information	
BidID	UserID	BidTime	BidAmount (cents)	ListingID	0
1	801	871	250	StartTime	11
2	1,820	1,710	2,361	EndTime	2,029
3	801	1,735	2,661	Duration	2,016
4	2,910	1,963	2,761	ItemValuation (cents)	3,935
5	1,820	1,998	3,761	itemTypeId	1
6	2,285	2,014	3,861	startPrice (cents)	100
7	801	2,025	3,961		
8	2,285	2,026	4,061		

4 Implementation

In the following section we describe some of the details and difficulties of implementing the generator. First, we describe the method and type of information collected from TradeMe. Second, how this information is represented using a collection of features. Third, how agents were designed using this information. Fourth, what happens during the simulator's execution. And fifth, the limitations of the simulator.

4.1 Data Collection

Data is collected from the TradeMe website [12] using the web crawler Scrapy and stored in a database. Crawling begins at one user's auction history page which contains a list of completed auctions. Completed auctions contain a list of bids which link to the auction history of each bidder. Unvisited pages are placed into a FIFO queue, so that the available auction history for users will be completely collected in the order they were discovered. This is important because the crawler only visits a small percentage of the existing user and auction pages. Using a FILO (first in last out) queue will result in incomplete auction histories for a large number of users.

From user pages, we extracted the number of positive, neutral and negative feedbacks they received, and the number of auctions they participated in. From completed auctions, we extracted bid times and amounts, the winning bid and bidder, item name, seller name and reputation. After cleaning, 56,256 auctions and 53,951 user profiles remained.

There are several limits to the information that can be collected. Firstly, only the most recent 20 bids are viewable for a completed auction. Secondly, bid histories for auctions are available only for 60 days after it has completed. Thirdly, some user pages are unavailable. The reason is not given, but one possibility is that those pages belong to identified fraudsters. The last limitation may be an advantage since we currently only wish to simulate legitimate users; removal of suspicious users will improve data quality.

4.2 User Features

It is difficult to identify patterns in bidder behaviour from lists of bids. Therefore, we represent bidders using a set of features that are easier to visualise and interpret. The set of user features used while developing the bidder agents are:

- a.* **Auction Count** the number of auctions in which the bidder made one or more bids
- b.* **Reputation** the difference between the number of positive and negative feedback
- c.* **Bid Amount** the average bid value for all bids the user made
- d.* **Excess Bid Increment** the average increment over the preceding bid, minus the required minimum increment
- e.* **Bids Per Auction** the average number of bids made per auction
- f.* **First Bid Time** the average number of minutes the first bid is made before the end of the auction
- g.* **Bid Time** the average bid time as a fraction of the auction time that has elapsed
- h.* **Win Proportion** the number of auctions won divided by Auction Count
- i.* **Bid Amount Proportion** the average bid value of all bids, with each bid scaled as a proportion of the winning bid
- j.* **Bid Proportion** the average fraction of bids that are made by this user in an auction

Features *a-f* are log-transformed so that values are spread out more evenly for clustering, since many values are very similar for some features.

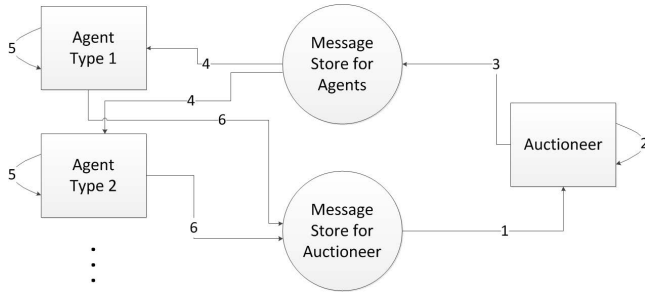
Some features, such as Auction Count, Bids Per Auction, and First Bid Time, can be directly used in rules for bidder-agents. Other features, such as Bid Amount Proportion, Win Proportion, and Bid Proportion, cannot, since their values depend on the behaviour of other agents. However, those features can be used to determine whether synthetic data from the generator is sufficiently similar to the TradeMe data.

4.3 Simulation Execution during 1 Time Unit

Figure 1 shows what happens during one time unit in the simulator. The time unit is split into two stages; the Auctioneer phase (A), and the Agent phase (B). During the Auctioneer phase, only the Auctioneer sends or receives messages, and performs actions; all other agents are inactive. During the Agent phase, this is reversed, and only agents perform actions. Agent actions can be performed in parallel using multiple threads. The Message Store for Agents is simply a hash map, where the key is the agent ID, and the value is the message. The Message Store for Auctioneer is a list.

4.4 Bidder-Agent Behaviour

The TradeMe users are clustered using the features listed in Section 4.2 using the simple *k*-means algorithm. To select the optimum number of clusters, we use the method proposed by Lange et al. [13] to find the number of clusters that gives the lowest stability value, where a lower value represents a better clustering solution. Stability was calculated for two to ten clusters, and stability was lowest when the number of clusters was four. Therefore, we used four centroids during simple *k*-means clustering.



- A. Beginning of Auctioneer phase
 1. Retrieve messages from Agents
 2. Process messages
 - a. update auction states, terminate finished auctions
 - b. ensure auctions and bids have valid states before and after update
 3. Send messages (current time, new auction, end of auction, price change messages)
- B. Beginning of Agent phase
 4. Retrieve messages from Auctioneer
 5. Decide on what actions to take, if any
 6. Send messages (new auction, bid messages)

Fig. 1. Events during the execution of the simulator in one time unit

From the clusters we identified patterns, for example: a significant number of users only bid close to the end of the auction and rebid quickly after they have been outbid; users who bid close to the end win more often; and users who bid early and win are often the only bidder in an auction. Bidder agents are given behaviours that allow the synthetic data to contain these same patterns. For example, for the last pattern, the rule may be to increase the likelihood of entering a bid as the number of existing bids in the auction increases, which leaves a fraction of auctions to have only 1 participating bidder.

There are additional rules that determine other aspects of behaviour: the moment a bidder-agent should begin bidding; how often and how much to bid above the minimum required bid; when to stop bidding, etc. These rules generally provide probabilities of whether an action should be taken, and a decision is made using a random number. For instance, an agent will bid, with 10% probability, on an item whose price has reached the agent's perceived value.

4.5 Limitations

A known limitation of this simulator is that it is unable to accurately model the bidding times of bids made close to the end of the auction. This is because the simulator uses time units of five minutes, which means that the fastest speed that an agent can respond to another bid is five minutes. The choice of five minute time units was a calculated trade-off between the simulation performance and precision: it allowed us to quickly generate data while retaining sufficient simulation precision.

As a result, a bidding war close to the auction end time may occur over 30 minutes or more, instead of within minutes. Thus, while the simulator can accurately model user bidding time patterns earlier in the auction, it cannot do so close to the end of the auction and distribution of values for features i and j in Section 4.2 from the simulator will not closely match those from the TradeMe data.

5 Evaluation

In this section we present the method and results for two evaluation techniques to validate our simulation. The method and results for the first evaluation are given Sections 5.1 and 5.2; and those for the second evaluation in Sections 5.3 and 5.4.

5.1 Evaluation Method Using Individual Features

We compared the similarity in the distribution of per-user values, for each feature described in Section 4.2, between the generated and TradeMe data. If the distribution of values from the generated data match those from the TradeMe data, then our model is more likely to be valid. The comparison is done using two correlation measures: Pearson's correlation coefficient and Spearman's rank correlation coefficient [14]. Pearson's correlation is a parametric test, while Spearman's rank correlation is a non-parametric test. The correlation measures are calculated individually for each feature.

For Pearson's correlation coefficient, a value of $+1$ or -1 indicates that a linear equation describes the relationship between the two variables perfectly. For Spearman's rank correlation coefficient, $+1$ or -1 occurs when each of the variables is a perfect monotone function of the other. A value close to $+1$ shows that the TradeMe and synthetic data are similar.

5.2 Evaluation Results Using Individual Features

Figure 2 shows the distribution of values for six features for both the synthetic and TradeMe datasets. To construct the figures, all values from the 30 synthetic datasets were used, giving 111,677 values for each feature. Values are split into 20 equidistant bins using the maximum and minimum values for that feature. The plots reflect the proportion of users that have a value that falls into a particular bin. We see that distribution of values for the 30 synthetic datasets generally match those from the TradeMe dataset.

Table 2 shows the correlation between synthetic and TradeMe data for each feature as measured by Pearson's correlation coefficient and Spearman's rank correlation coefficient. The correlation values for both measures are greater than 0.9 for all features, with the exception of Bid Amount, with a Pearson's correlation of 0.53 and Excess Bid Increment with 0.86. The low similarity for Bid Amounts, and to a lesser extent, Excess Bid Increment, is due to the lack of information on the perceived and actual value of items, making it more difficult to accurately model bid values.

As a whole, the high correlation values show that the value distribution of features from our generated data is similar to TradeMe data.

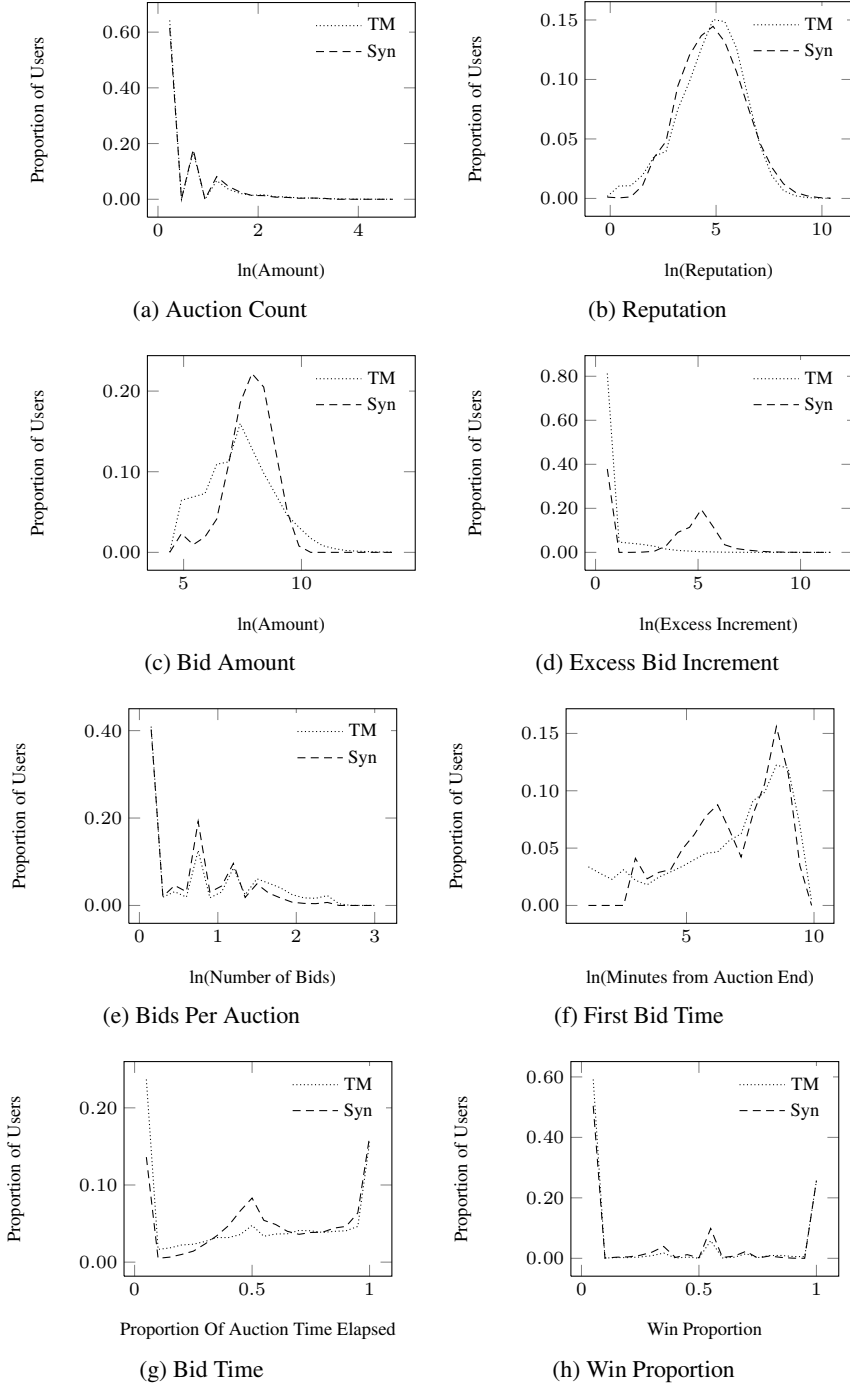
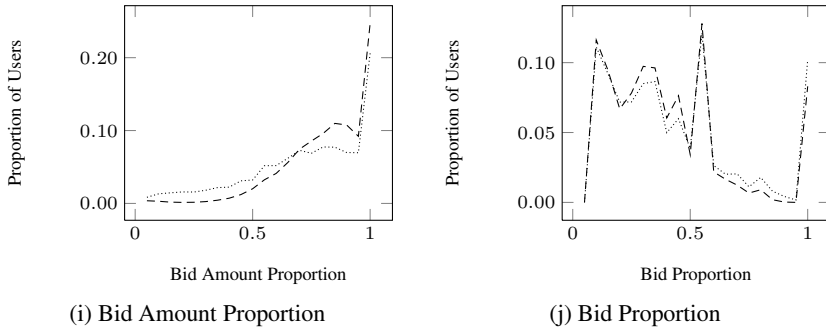


Fig. 2. Value distributions of features

**Fig. 2.** (continued)**Table 2.** Correlation results for all features

Feature	Pearson's Correlation		Spearman's Rank
	95% CI	Correlation	ρ
Auction Count	0.9468 - 0.9485	0.9476	0.9625
Reputation	0.9379 - 0.9399	0.9389	0.99997
Bid Amount	0.5207 - 0.5329	0.5269	0.9999
Excess Bid Increment	0.8545 - 0.8590	0.8567	0.8497
Bids Per Auction	0.9702 - 0.9711	0.9706	0.9930
First Bid Time	0.9889 - 0.9892	0.9890	0.99998
Bid Time	0.9701 - 0.9710	0.9706	0.9967
Win Proportion	0.9752 - 0.9760	0.9756	0.9515
Bid Amount Proportion	0.9873 - 0.9877	0.9875	0.9987
Bid Proportion	0.9841 - 0.9846	0.9844	0.9981

5.3 Evaluation Method Using Multiple Features

Lange et al. proposed a method to solve the model order selection problem in cluster analysis; that is, to find a suitable number of clusters k . They state that a cluster solution should be robust: that the cluster analysis should be reproducible using other datasets drawn from the same source, and not depend on the particular sample [13]. The degree to which the cluster solution is reproducible can be used to determine the number of clusters k that should be inferred from a dataset. However, since only one dataset is usually available, the dataset is split into two, and treated as two samples.

Conversely, if the number of clusters for a dataset is given, the degree of reproducibility gives an indication of the similarity of the second dataset to the first. We adapt the method proposed by Lange et al. to evaluate the similarity of TradeMe data with synthetic data. There are two main changes to the method; (1) there is no need to normalise results according to the number of clusters inferred since the number remains unchanged, and (2) two different datasets are used: the TradeMe dataset, and a synthetic dataset.

For evaluating the similarity between datasets, the equation for measuring the normalised Hamming distance between two labelling vectors (in [13]) can be slightly modified to:

$$d(\phi(X'), Y') = \frac{1}{n} \sum_{i=1}^n 1\{\phi(X'_i) = Y'_i\} \quad (1)$$

where $1\{\phi(X'_i = Y'_i)\} = 1$, if $\phi(X'_i) = Y'_i$. This gives the proportion of instances that are given the same labelling via the classifier trained on X (which in our case is the TradeMe dataset) and the clustering solution Y' for X' (the synthetic dataset). Since the corresponding clusters may be assigned different labels in two different clustering solutions even if the datasets are the same, agreement is maximised using the Hungarian method as suggested by Lange et al.

We used the simple k -means algorithm for clustering the TradeMe and synthetic data into four clusters. Clustering the TradeMe dataset X , gives a set of labels Y , one for each instance. Together, (X, Y) is used to construct a closest neighbour classifier ϕ . Clustering the synthetic dataset X' gives a set of labels Y' . The classifier ϕ is then used to classify instances in X' , giving the labels $\phi(X')$. With Y' and $\phi(X')$ we can calculate the accuracy using Equation 1.

Features Used. We used four features for evaluating the similarity between the synthetic and TradeMe data: Proportion of Max Bid, Win Proportion, Bids Per Auction, Bid Proportion. These four features were chosen because the others:

1. Have almost the same values for all users, e.g. excess bid increment as shown in Figure 2(d), and are less useful for clustering.
2. Are identically distributed regardless of their cluster assignments, e.g. reputation, bid amount, and are not useful for clustering.
3. Are known not to closely approximate real data (features i and j), as discussed in Section 4.5.

5.4 Evaluation Results Using Multiple Features

We evaluated our simulator using 30 synthetic datasets, each with roughly 3,700 users. We report the results for evaluations using random starting centroids for simple k -means, and using manually defined centroids.

Using 50 sets of random initial centroids for evaluating each synthetic dataset, the average accuracy is 0.745 with a standard deviation of 0.0957. Using the first set of centroids shown in Table 3, the average accuracy is 0.878, an increase of 0.143, with a standard deviation of 0.0282.

Centroid Selection. The k -means algorithm is sensitive to the centroids chosen during initialisation [15], which in turn heavily influences the result given by Equation 1. In our case, this problem may be worsened by characteristics of our dataset. First, clusters are not well defined in the Trade Me data, which means clusters will tend to grow from their initial position. Secondly, the Win Proportion feature is almost binary, where the majority of users have a value of zero or one. Given four initial centroids, there are five configurations the centroids could take for Win Proportion, e.g., all centroids have a Win Proportion of zero. If the configuration of centroids were different for two datasets, the accuracy value will be very low even though the datasets may be very similar. Therefore, it is necessary to manually define the initial centroids so that two will have a value of one, and two with zero. This is similar to the method by Fang et al. [15] to select centroids that maximise inter-centroid distance.

Table 3. Selected centroid centers

	Centroid ID	Bid Amount Proportion	Win Proportion	Bids Per Auction	Bid Proportion
Appropriate Centroids	1	0.8	0.0	0.0	0.2
	2	0.8	0.0	0.0	0.2
	3	1.0	1.0	0.0	0.5
	4	1.0	1.0	0.0	1.0
Inappropriate Centroids	1	0.8	0.0	0.0	0.2
	2	0.8	0.0	0.0	0.2
	3	1.0	0.0	0.0	0.5
	4	1.0	1.0	0.0	1.0

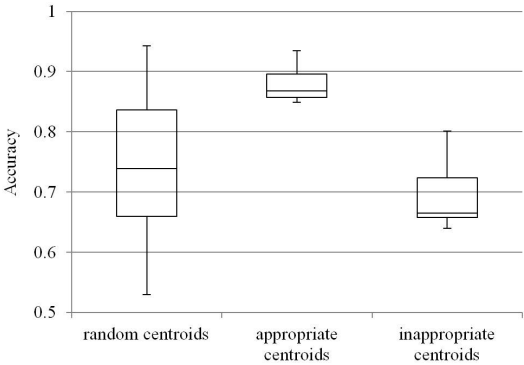


Fig. 3. Accuracy using different centroids for clustering using four features: Bid Amount Proportion, Win Proportion, Bids Per Auction, Bid Proportion

We give evidence of this in Table 3 and Figure 3. Table 3 shows two sets of centroids. The set named “appropriate centroids” have two centroids with the value of 0, and two centroids with the value of 1 for Win Proportion. For the set named “inappropriate centroids”, the difference is that the third centroid has as value 0 of instead of 1 as Win Proportion. Figure 3 shows that this change reduces median accuracy by 0.203 and mean accuracy by 0.192.

6 Conclusion

We successfully implemented a agent-based simulator for online auctions. We evaluated the quality of data generated by our simulator in two ways; using two correlation measures, and a modified version of the Stability measure proposed by Lange et al. [13]. The results show that our synthetic data closely matches the TradeMe data. Correlation of individual features compared to TradeMe data exceeds 0.9 for 8 out of 10 features for both measures, and median accuracy as calculated by Equation 1 is 0.87 for user defined centroids.

In the future, we intend to reduce the time unit length from five minutes, so that we can better model bidding at the end of auctions. In addition, we intend to implement additional agents to add fraudulent bids into the auctions. This will allow us to recreate various types of fraud, and test fraud detection algorithms using the generated synthetic data.

References

1. Scherer, K.: Retail store wars move online (2011)
2. Bhargava, B., Jenamani, M., Zhong, Y.: Counteracting shill bidding in online english auction. *International Journal of Cooperative Information Systems* 14 (2005)
3. Pandit, S., Chau, D.H., Wang, S., Faloutsos, C.: Netprobe: a fast and scalable system for fraud detection in online auction networks. In: *Proceedings of the 16th International Conference on World Wide Web*, pp. 201–210 (2007)
4. Chae, M., Shim, S., Cho, H., Lee, B.: An empirical analysis of fraud detection in online auctions: Credit card phantom transaction. In: *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, p. 155a. IEEE Computer Society
5. Chau, D., Pandit, S., Faloutsos, C.: Detecting Fraudulent Personalities in Networks of Online Auctioneers. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) *PKDD 2006. LNCS (LNAI)*, vol. 4213, pp. 103–114. Springer, Heidelberg (2006)
6. Trevathan, J., Read, W.: Detecting collusive shill bidding. In: *Fourth International Conference on Information Technology, ITNG 2007*, pp. 799–808 (2007)
7. Shah, H., Joshi, N., Sureka, A., Wurman, P.: Mining eBay: Bidding Strategies and Shill Detection. In: Zaïane, O.R., Srivastava, J., Spiliopoulou, M., Masand, B. (eds.) *WebKDD 2003. LNCS (LNAI)*, vol. 2703, pp. 17–34. Springer, Heidelberg (2003)
8. Macal, C.M., North, M.J.: Agent-based modeling and simulation. In: *Proceedings of the Simulation Conference (WSC)*, pp. 86–98 (Winter 2009)
9. Mizuta, H., Steiglitz, K.: Agent-based simulation of dynamic online auctions. In: *Simulation Conference Proceedings*, vol. 2, pp. 1772–1777 (Winter 2000)
10. Bapna, R., Goes, P., Gupta, A.: Simulating online yankee auctions to optimize sellers revenue. In: *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, p. 10 (2001)
11. Menezes, F.M., Monteiro, P.K.: Auction Types. In: *An Introduction to Auction Theory*, pp. 10–11. Oxford University Press, USA (2005)
12. TradeMe (April 17, 2012), <http://www.trademe.co.nz/>
13. Lange, T., Roth, V., Braun, M.L., Buhmann, J.M.: Stability-based validation of clustering solutions. *Neural Comput.* 16(6), 1299–1323 (2004)
14. Myers, J.L., Well, A.D.: *Research Design & Statistical Analysis*, 2nd edn. Lawrence Erlbaum Associates (2003)
15. Haw-ren, F., Saad, Y.: Farthest centroids divisive clustering. In: *Seventh International Conference on Machine Learning and Applications, ICMLA 2008*, pp. 232–238 (2008)