A Fraudster in a Haystack: Crafting a Classifier for Non-delivery Fraud Prediction at Online Auction Sites

Vinicius Almendra, Denis Enăchescu
Faculty of Mathematics and Informatics
University of Bucharest
Bucharest, Romania
vinicius.almendra@gmail.com, denaches@fmi.unibuc.ro

***Abstract*—Non-delivery fraud is a recurring problem at online auction sites: false sellers that list inexistent products just to receive payments and disappear, possibly repeating the swindle with another identity. The high transaction volume of these sites calls for the use of machine learning techniques in fraud prediction systems, at least for the identification of suspect sellers which deserve further expert analysis. In our work we identified a set of features related to listings, sellers and product categories, and built a system for fraud prediction taking into account the high class imbalance of real data, since fraud is a relatively rare event. The identified features are all based on publically accessible data, opening the possibility of developing fraud prediction systems independent of site operators. We tested the proposed system with data collected from a major online auction site, obtaining encouraging results on identification of fraudsters before they strike, while keeping the number of false positives low.**

***Index Terms*—fraud prediction, non-delivery fraud, online auction sites, machine learning**

I. Introduction

Online auction sites like eBay offer unprecedented business possibilities for sellers and buyers through the creation of virtual marketplaces of global reach. Fraudsters also realized the opportunities opened by such virtual marketplaces. Among the several types of fraudulent behavior at online auction sites, the most frequent one is non-delivery fraud [1, 2]: false sellers list non-existing products for sale, receive payments and disappear, possibly reentering in the market with a different identity. According to the annual report of the Internet Crime Complaint Center, non-delivery fraud is the fourth most reported Internet crime [3]. The challenge faced by site operators is to identify fraudsters *before* they strike, in order to avoid losses due to unpaid taxes, insurance, badmouthing etc. From now on we are going to refer to this problem as fraud *prediction*, in contrast with the problem of fraud *detection* (identification of fraudulent behavior that already took place). Since online auction sites are huge information systems and all transactions are carried over electronically, a natural approach to the fraud prediction problem is to use machine learning systems.

The problem of fraud prediction using machine learning has been studied for a variety of domains: credit card transactions, telephone networks, money laundering, academic fraud etc. [4]. Research works targeting specifically fraud at online auction sites are more recent [5, 6, 7, 8, 9, 10], probably due to the lack of publically available data.

Although existing fraud prediction methods achieved good accuracy, they may be useless from a practical point of view. Since fraud cases are proportionally few – only 0.01% of listings are fraudulent, according to eBay [11] –, a system with an advertised accuracy of 95% in a real setting would have a precision of 0.19%, which means that for each fraudster correctly classified as such (a true positive), around 500 legitimate sellers would be also classified as fraudsters (false positives). Automatically taking coercive measures, e.g. suspension, against so much bona fide sellers could cause more harm to the online auction site than the fraudsters. If machine classification is complemented with human expert analysis, the effort to find a single fraudster can also be excessive [4]. In other words, fraud prediction systems must take a special care with the rate of false positives, since each day thousands of new products are listed. Although dangerous, fraudulent offers are like needles in the big haystack of product listings.

In this paper we will present a fraud prediction system that explicitly allows one to balance sensitivity (true positives rate) with specificity (true negatives rate) in order to achieve a desired precision (proportion of fraudulent listings among those classified as such). We will also propose a set of features that can be extracted from online auction sites' public information, what eases the replication of this study. We also extended the idea of using category-level features [12], taking advantage of the hierarchical nature of product categories.

In Section II we will present the context for our research; in Section III we will describe the dataset used to validate our proposed approach and present the selected features; in Section IV we will explain our proposed system for predicting non-delivery fraud; in Section V we will present the experimental results; we will conclude our work in Section VI.

II. Background and Related Work

Bolton and Hand [4]did a comprehensive review regarding statistical fraud detection in several domains: credit card

fraud, money laundering, telecommunications fraud, computer intrusion, and scientific fraud. They highlighted some challenges for fraud detection: the high number of cases to be analyzed, the need of fast algorithms, uneven class sizes (class imbalance), uneven misclassification cost, the problem of false positives. Although they did not mention fraud at online auction sites, these challenges also apply.

In the last years appeared papers specifically focused on fraud at online auction sites, some from a descriptive perspective [13, 14, 15], and others aiming fraud prediction [5, 6, 7, 8, 9, 10, 16].

Fraud prediction systems need to tackle the following problems: *feature extraction* and *method selection*. Regarding feature extraction, some works relied on public information obtained from online auction sites portals' [12, 6, 5, 9]; some used features related to seller past transactions e.g. average product price in the last 15 days [6, 5, 12]; others used information extracted from the social network surrounding sellers [9]; one made use of time-related variations of seller behavior (phased models) [5]. We include contextual information related to the *category* of the listed products: average price, number of sellers that listed products in the same category, frequency of fraudulent behavior etc. This allowed us to check for example if a listing's price is much below the average. This idea also appeared in another work [12], although with less features. There are also works that used internal information of online auction sites [10, 8, 16], which offers a richer set of features, at the expense of confidentiality restrictions concerning what can be disclosed.

Regarding the methods employed to create classification models, previous works explored several of them: decision trees [6, 7], Markov random fields [9], instance-based learners [5], logistic regression [10], online probit models with stochastic variable selection [16]. We used a combination of one-class Support Vector Machines and boosting trees.

Class imbalance is an obstacle for the use of supervised learning systems in fraud prediction [4]. This issue in fact appears twice in the process of fraud prediction: in the *modeling* phase and in the *production* phase. We will discuss them separately.

In the modeling phase the class imbalance problem appears in its classical formulation: the need to compensate the imbalance in training data in order to achieve sensitive (high recall) classification models, since algorithms tend to privilege the prevalent class (in this case, legitimate listings). Some common approaches to solve this problem are undersampling of the majority class, oversampling of the minority class, and SMOTE (Synthetic Minority Over-sampling Technique) [17]. In this case, what is being optimized is the sensitivity in the test data. Some of the above-mentioned works used undersampling [6, 5], one use an unsupervised model [9], others did not state the approach adopted [16, 10, 8].

In the production phase the issue is the impact of false positives – legitimate listings classified as suspect –, which depends on the proposed use for the fraud prediction system. This problem is less relevant for those whose final user is the buyer [6, 12, 9], since in this case the impact of false positives is limited to reducing the number of trusted (i.e. not suspect) sellers from whom to buy. If there are enough true negatives (i.e. trusted legitimate sellers), buyers may not even notice this issue. On the other hand, the rate of false positives is crucial for the systems whose objective is to support online auction sites, typically through the identification of the most suspicious listings for further investigations [5, 8, 10, 16]. A high rate of false positives raises investigation costs, eroding the online auction site gains with fraud prevention. Chang et al. [5] do not mention this issue; although their results are promising (93.17% of sensitivity and 94.56% of specificity), they are still not practical, since they imply 583 false positives for each true positive. In sites where each day hundreds of thousands of new listings are added, even that small false positive rate means a high workload for further investigation.

III. Dataset Description

A. Data Collection

We targeted in our research one specific online auction site, named MercadoLivre (www.mercadolivre.com.br). It is the biggest Brazilian auction site, online since 1999. As of September 2011 it had 62 million registered users in Latin America. In the period January-September 2011 were sold MercadoLivre 36,9 millions of products and the Gross Merchandise Volume was US$3.4 billions [18]. It is affiliated with eBay and has similar functionality, although offering fewer options. In sake of brevity, from now on we will refer to MercadoLivre as ML. In the whole year of 2011 we crawled daily 11 categories of products where we expected more fraud occurrence, extracting information about 2,134,292 product listings. Using a previously developed methodology [19], we identified 439 listings with clear signs of non-delivery fraud, among others sellers who owned those listings were suspended by ML and at least two buyers left textual feedback stating that they had paid for the product but it had not been delivered. These listings were labeled as *fraudulent listings*. All listings that were not identified as being fraudulent were labeled as *legitimate listings*.

B. Features for Fraud Prediction

Our unit of observation is the *product listing*, so our features are also directly or indirectly linked to it. The directly linked features are *price*, *date* (when the listing appeared in the site), *product category* and *seller* (ML's user who owns the listings). We also include information related to the seller: *reputation score*, *account age* (how old the seller account is, in days), and *number of recent transactions*. These features have also been used in other works about fraud prediction. The values of these features were the ones collected *at the moment the listing appeared in ML's site*, since we wanted to predict fraud *before* transactions took place and before any sign of suspicion. We were able to do this since we did a longitudinal data collection, tracking sellers and listings since the moment when they appeared in the site [19].

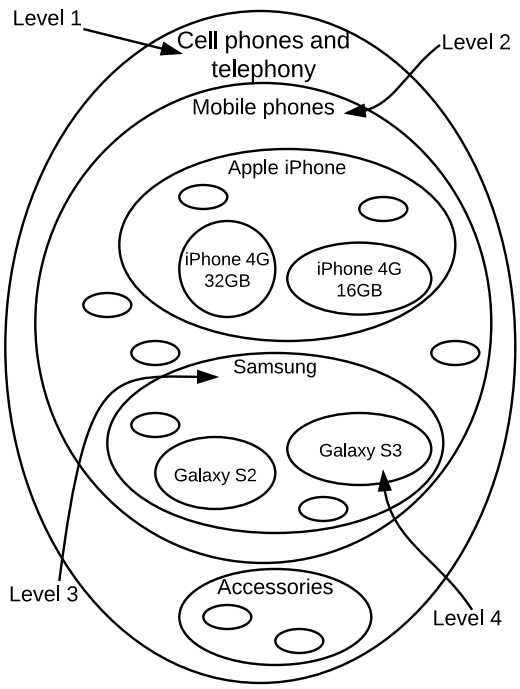

Figure 1. Excerpt of ML's category hierarchy as a Venn diagram

We also included features related to the *product category*, since we expected that fraudsters would not choose randomly which product to list. Product categories specify the type of products, their models, characteristics etc., and sellers have to choose in which category to list their products. The set of available categories is predefined by ML. So we used our dataset to extract aggregated measures about product categories over the entire year of 2011. All listings that shared the same category had the same values for these features.

Product categories in ML are organized as tree, with 23 root nodes and a depth up to 6. Each listing belongs to a specific category *and to all its ancestors*. In other word, each category is a subset of its parent. Figure 1 exemplifies this structure for one top-level category. So, the measures about a product category are calculated using the listings specifically belonging to it and the listings belonging to all its descendants.

Regarding the product category of the listing, we selected three features: the *total number of listings, total number of sellers,* and *average price*. The first two reflect the popularity of the category among sellers, while the third shows how profitable to fraudsters the category can be. It also allowed us to calculate another feature of product listings: the *relative price difference*, given by $(listingprice - averageprice)\,/averageprice$.

We also wanted to capture more general information about the "type" of the product. We observed that deeper categories (level 4 and beyond) generally captured different characteristics of the same main product. So we also calculated the *total number of listings* and *total number of sellers* for the ancestor category at level 3. For example, if a product belonged to the category "Cell phones and Telephony $>$ Mobile phones $>$ Apple iPhone $>$ iPhone 4G 32GB", these features were calculated for the category "Cell phones and Telephony $>$ Mobile phones $>$ Apple iPhone". Finally, since we sampled systematically those product categories, we could also calculate the *category fraud rate* (number of fraudulent listings divided by the total number of listings). We did this for the ancestor category at level 2. We opted to calculate this feature for level 2 instead of level 3 to avoid biasing the classification models, since most categories at level 3 had fraud rate zero, which means that all listings belonging to them would end up automatically classified as legitimate irrespective of the other features. Table I summarizes the features used.

Table I
SUMMARY OF THE DATASET'S FEATURES

Entity	Feature
Product listing	price
	date
	category
	relative price difference
Product listing's seller	reputation
	account age
Product listing's category	number of listings
	number of sellers
	average listing price
Product listing's category (at level 3)	number of listings at level 3
	number of sellers at level 3
Product listing's category (at level 2)	Category fraud rate at level 2

IV. PROPOSED NON-DELIVERY FRAUD PREDICTION SYSTEM

Our proposal is to combine two different supervised learning models – one-class Support Vector Machines and boosting trees – with an extra step we named *finding extra suspect listings* to improve sensitivity based on listing ownership and publishing time. The class imbalance problem is managed through undersampling and resampling. In Figure 2 we depict how these elements are related. In the next sections we explain each one, assuming the existence of a training set – labeled data – containing fraudulent and legitimate listings, and a test set, whose listings should be labeled. Notice that the features *seller* and *date* (of the listing) were used only to find extra suspect listings; the first two phases did not consider them.

A. Filtering Phase (one-class SVM)

The first phase aimed the identification of listings that were "clearly" legitimate before applying the next classification model, so as to reduce the false positives rate. Inspired by a previous work [20], we approached this problem as one of outlier detection. However, instead of treating fraudulent listings as outliers, we treated *legitimate* listings as such. The idea was of identifying which listings were noisy, "far" away from the fraudulent ones in the feature space. After some tests we opted for Support Vector Machines for one-class classification.

We built the one-class SVM model using only the fraudulent listings of the training set. Then we applied this model to

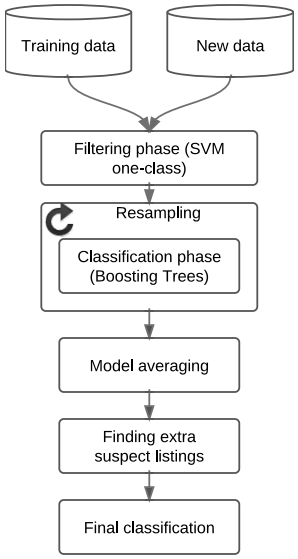

Figure 2. Overview of Fraud Prediction System

the test set. The examples labeled as legitimate ("outliers") were classified right from the start as legitimate. Those labeled as fraudulent (positive examples) were classified in the next phase. We named this as the *filtering phase* since it "filters out" listings that are obviously legitimate, letting it pass those who deserve further consideration. Two important underlying assumptions were that the trained model should have a very high sensitivity (close to 100%) and a nonzero specificity (e.g. $>$10%). The specificity give the fraction of legitimate listings that would be filtered out.

B. Classification Phase (Boosting Trees)

Among existing classification methods, we opted for boosting trees after some exploratory analysis, since its performance changed smoothly with increasing class imbalance.

The idea of boosting trees consists of applying successive times the same classifier, in this case a decision tree, but each time adjusting the weights of the training examples, so as to give more importance to previously misclassified data points [21]. In the end results are averaged, weighted by the relative classification error. There are several variants of boosting; we used the one implemented in the R package GBM [22]. Among the several options of loss functions, we opted for the Bernoulli one, since it is recommended for classification tasks. Model parameters were selected through 5-fold cross-validation with the training set, choosing the set of parameters that maximized sensitivity instead of accuracy, in order to reduce the effect of class imbalance.

The key point here was the composition of the training set, which we assumed to be high imbalanced (less than 5%). Instead of using the training set directly to build the model, we used undersampling to generate a new training set with a predefined degree of imbalance. Let N_F be the number of fraudulent listings in the original training set, N_L the number of legitimate ones, $ratio$ the degree of imbalance, i.e. the proportion between the number of listings in the two classes *in the new training set*. We generated this new set taking all N_F fraudulent listings and joining with $ratio \times N_F$ legitimate listings selected randomly from those N_L. If $ratio = 2$, then the new training set would have two legitimate listings for each fraudulent one. The choice of $ratio$ depended on the accepted trade-off between false positives and false negatives, since a bigger $ratio$ increased specificity but decreased sensitivity.

Since the majority of normal examples were left out due to undersampling, we expected an increased model variance. In order to overcome this, we used a bagging approach: we generated several different training sets (*resamples*) using the procedure outlined above; then we trained the model with each one, and applied all of them to the test data. Finally we averaged all results, obtaining the labeling of test examples for the classification phase.

C. Finding Extra Suspect Listings

Fraudsters frequently list several products at once [16], so when one listing is considered fraudulent, other active listings from the same seller are probably also fraudulent. In order to take advantage of this, we applied an idea borrowed from a similar work [10]: when one listing in the test set was predicted to be fraudulent, all other listings of the same seller which were posted starting from one week before up to one week after that fraudulent listing were also classified as fraudulent. We did this process on the final result of the classification phase.

D. Implementation

Algorithm 1 shows the pseudo-code for our system, which was implemented in R [23]. $trFraud$ is the set of fraudulent listings in the training set; $trNormal$ is the set of legitimate listings in the training set; $testSet$ is the test set, containing both fraudulent and legitimate listings; $resample$ gives a random subset.

V. Experimental Results

A. Training and Test Sets

From the dataset described in Section III we created a training and a test set by random sampling with the following distributions:

- Training: 326 fraudulent listings and 21,422 legitimate ones;
- Test: 113 fraudulent listings and 21,914 legitimate ones.

Since many sellers (including fraudsters) post multiple listings, we took care that the listings of each seller appeared either in the training or in test set, so as to not artificially improve results.

B. Performance Measures

We used as performance measures *sensitivity*, *specificity* and *positive predicted value* (PPV). Sensitivity measured the proportion of fraudulent listings correctly spotted as such; specificity did the same for legitimate listings; PPV gave us the proportion of real fraudulent listings among those classified

Algorithm 1 Fraud prediction algorithm

{svm one-class model using all fraudsters from training data}
$model_{svm} \leftarrow trainSvmOneClassModel\,(trFraud)$
$pred_{svm} \leftarrow predict\,(testSet, model_{svm})$
for $i = 1$ **to** $n_{resamples}$ **do**
  $resample \leftarrow trFraud \cup resample\,(trNormal, ratio)$
  Find best parameters using 5-fold cv on $resample$
  $model_{boost} \leftarrow trainBoostingModel\,(resample)$
  $pred_{boost}\,[i] \leftarrow predict\,(testSet, model_{boost})$
{Model averaging}
$predAvg \leftarrow \sum_i pred_{boost}[i]/n_{resamples}$
for $k = 1$ **to** $n_{listings}$ **do**
  if $predAvg\,[k] \geq 0.5$ **then**
    $predAvg\,[k] \leftarrow 1$
  else
    $predAvg\,[k] \leftarrow 0$
{Mark as legitimate the listings classified as such by the one-class SVM}
for $k = 1$ **to** $n_{listings}$ **do**
  if $pred_{svm}\,[k] = 0$ **then**
    $pred_{both}\,[k] \leftarrow 0$
  else
    $pred_{both}\,[k] \leftarrow predAvg\,[k]$
{List of the sellers of the listings predicted as fraudulent}
$fraudsters \leftarrow \{s | seller\,(L) = s \wedge pred_{both}\,[L] = 1\}$
{Re-classifies as fraudulent the listings classified as legitimate but are close in time to the ones classified as fraudulent that belong to the same seller}
$pred_{final} \leftarrow pred_{both}$
for all s in $fraudsters$ **do**
  {Listings belonging to s classified as fraudulent}
  $L_F \leftarrow \{L | seller\,(L) = s \wedge pred_{both}\,[L] = 1\}$
  {Listings belonging to s classified as legitimate}
  $L_N \leftarrow \{L | seller\,(L) = s \wedge pred_{both}\,[L] = 0\}$
  for all L in L_N **do**
    if exists L' in L_F listed up to 7 days before or after L **then**
      $pred_{final}\,[L] \leftarrow 1$
return $pred_{final}$

as such, but with one more detail: we adjusted it to take into account the *prevalence* of fraud in a real setting. The PPV gave us what would be the precision of the classifier if the proportion of fraudulent and legitimate listings in the test set were equal to the prevalence of fraud. The formula used to calculate *PPV* was:

$$PPV = \frac{sens \times prev}{sens \times prev + (1 - spec) \times (1 - prev)}$$

where $prev$ denotes the prevalence. In our study we assumed $prev = 0.01\%$, the number once advertised by EBAY [11]. PPV measure is interesting because with it we can answer the question of *how many legitimate listings will be misclassified for each fraudulent one correctly classified* – $(1-PPV)/PPV$. This is a proxy for the fraud prediction effort the site has to do with our system.

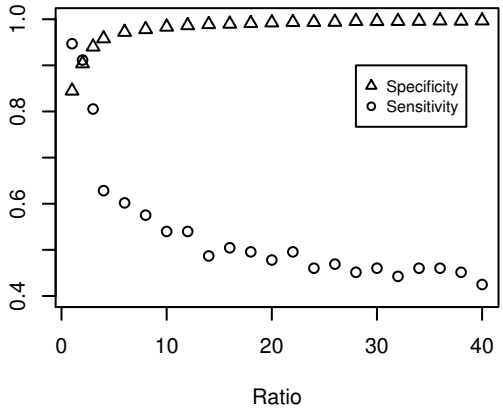

Figure 3. Sensitivity and specificity vs. $ratio$

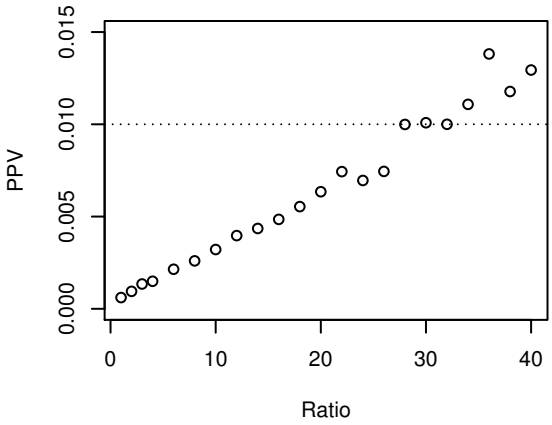

Figure 4. PPV vs. $ratio$

C. Parameter Selection

We chose the ν parameter of the one-class SVM as 0.001, after some tests using only the training set. The parameters of the boosting trees (number of trees, shrinkage and interaction depth) were chosen for each resample using 5-fold cross-validation, as we mentioned before. Regarding the *ratio* parameter, we did not set an *a priori* value for this parameter. Instead, we opted to test with several values in the range 2–40, in order to establish the system behavior.

Finally, regarding the number of resamples, we set it to 10 after some experimentation, since with this value the results already converged.

D. Results

We first measured the effect of the *ratio* parameter of our system. Figure 3 shows the evolution of the sensitivity and specificity with increasing values of the ratio between fraudulent and legitimate listings. Figure 4 shows the same for PPV.

Table II
PERFORMANCE WITH DIFFERENT RATIOS

ratio	Measure	One-class SVM	Boosting trees	One-class SVM + Boosting trees	Extra suspect listings*	One-class SVM + Boosting trees + extra suspect listings
2	Sensitivity	97.24%	77.87%	76.99%	50.00%	88.50%
	Specificity	29.90%	92.91%	93.31%	98.82%	92.21%
	PPV	0.02%	0.11%	0.11%	-	0.11%
	Accuracy	30.91%	92.84%	93.23%	98.76%	92.19%
30	Sensitivity	97.24%	36.28%	35.39%	15.06%	45.13%
	Specificity	29.90%	99.67%	99.69%	99.87%	99.56%
	PPV	0.02%	1.10%	1.11%	-	1.02%
	Accuracy	30.91%	99.35%	99.36%	99.59%	99.28%

* Extra suspect listings phase operates only on the listings considered normal in the previous phase (one class SVM + Boosting trees), hence PPV does not make sense here.

Table III
CLASSIFICATION RESULTS WITH DIFFERENT RATIOS

ratio	Measure	One-class SVM	Boosting trees	One-class SVM + Boosting trees	Extra suspect listings	One-class SVM + Boosting trees + extra suspect listings
2	True positives	317	88	87	13	100
	False positives	15,016	1,553	1,466	241	1,707
	True negatives	6,406	20,361	20,448	20,207	20,207
	False negatives	9	25	26	13	13
30	True positives	317	41	40	11	51
	False positives	15,016	71	69	27	96
	True negatives	6,406	21,843	21,845	21,818	21,818
	False negatives	9	72	73	62	62

Observing the figures above, we can devise two extreme scenarios when using the proposed system: a *high-sensitivity* scenario and a *high-precision* scenario. The first one is good for buyer-side fraud prediction tools. In this scenario clearly the best $ratio$ is two (88.50% sensitivity and 92.21% specificity). However, for a high-precision scenario, there is a trade-off between PPV and sensitivity. The dotted line in Figure 4 shows the threshold of 1%, which means that for each fraudulent listing found, 99 legitimate ones were incorrectly classified. In Table II we present the detailed performance information broken down by method phase for $ratio \in \{2, 30\}$, and in Table III we present the classification results also broken down by phase.

In Table IV we display the results of the proposed system compared with the ones obtained through other classification methods. Parameters where tuned the same way (5-fold cross-validation). We excluded the third phase, since it just improved the result of the previously done classification. We did not present a comparison using bigger values of $ratio$ since other methods did not behave well in this scenario when we used their off-the-shelf versions.

Table IV
COMPARING PERFORMANCE WITH OTHER LEARNING METHODS ($ratio = 2$)

Method	Sensitivity	Specificity	PPV	Accuracy
Ours (one-class SVM+boosting trees)	**66%**	**93%**	**0.09%**	**93%**
SVM with radial kernel	66%	86%	0.04%	86%
Nearest neighbors	63%	84%	0.04%	83%
Random Forests	65%	88%	0.05%	87%

VI. CONCLUSIONS

We validated the proposed fraud prediction system using a comprehensive dataset extracted from a major online auction site. We considered two scenarios where a such a prediction tool could be used: the high-sensitivity scenario and the high-precision scenario. In the high-sensitivity scenario, we achieved a sensitivity of 88.50%, with a specificity of 92.21%, a performance close to the one achieved by existing solutions that use public data from online auction sites. In the high-precision scenario, where false positives play an important role, we managed to achieve a positive predicted value (precision) of 1.02%, with sensitivity 45.13% and specificity 99.56%. Although we consider this sensitivity still unsatisfactory, these results can be a good starting point for a semi-automatic fraud prediction system in a real setting, where prevalence of fraud could be as low as 0.01%. One should notice that our feature set is much more limited than the one available to online auction sites internally, even though the results confirmed that the chosen features were relevant for this task. The combination of one-class SVM and boosting trees outperformed other classification methods we tested.

The main contributions of this research were: (i) a feature set for fraud prediction, (ii) a supervised learning system for fraud prediction based on one-class SVM and boosting trees, (iii) a methodology to cope with the problem of false positives through undersampling and resampling.

The added value of the Filtering phase with one-class SVM has been marginal. We believe that the decision boundary calculated by the one-class SVM was too similar to the one of the boosting trees; further studies are needed in order to verify if it is really worth.

One limitation of our results is that they were restricted to non-delivery fraud, since all cases in our training set were of this type.

As future work we intend to analyze the importance of the features, since some are correlated and could perhaps be dropped or at least combined through Principal Component Analysis. Further work is needed on parameter selection, specially of the *ratio* parameter, in order to automatically choose the value that leads to the targeted precision. The incorporation of temporal features regarding sellers and listings, as in the work of Chang et al. [5], might improve classification performance. We also intend to verify whether existing methods to attenuate class imbalance would improve

the trade-off between sensitivity and positive predicted value. Finally, the proposed system could be tested using data from other online auction sites, since most of the features used are common to many of them, especially on those inspired on eBay.

ACKNOWLEDGMENTS

This work was sponsored by University of Bucharest, under postdoctoral research grant nr. 17824 of October 18th, 2011.

REFERENCES

[1] B. Gavish and C. Tucci, "Reducing internet auction fraud," *Communications of the ACM*, vol. 51 (5), 2008.

[2] D. G. Gregg and J. E. Scott, "A typology of complaints about ebay sellers," *Communications of the ACM*, vol. 51 (4), pp. 69–74, 2008.

[3] Internet Crime Complaint Center, "Internet crime report," tech. rep., 2011.

[4] R. Bolton and D. Hand, "Statistical fraud detection: A review," *Statistical Science*, vol. 17, no. 3, pp. 235–255, 2002.

[5] W.-H. Chang and J.-S. Chang, "A novel two-stage phased modeling framework for early fraud detection in online auctions," *Expert Systems with Applications*, vol. 38, pp. 11244–11260, Sept. 2011.

[6] D. H. Chau and C. Faloutsos, "Fraud detection in electronic auction," in *Proceedings of European Web Mining Forum*, 2005.

[7] C. Chiu, Y. Ku, T. Lie, and Y. Chen, "Internet auction fraud detection using social network analysis and classification tree approaches," *International Journal of Electronic Commerce*, vol. 15, pp. 123–147, Apr. 2011.

[8] R. Maranzato, A. Pereira, A. P. d. Lago, and M. Neubert, "Fraud detection in reputation systems in e-markets using logistic regression," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, (Sierre, Switzerland), pp. 1454–1455, ACM, 2010.

[9] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos, "NetProbe: a fast and scalable system for fraud detection in online auction networks," in *Proceedings of the 16th international conference on World Wide Web*, WWW 2007, (Banff, Alberta, Canada), ACM Press, 2007.

[10] L. Zhang, J. Yang, W. Chu, and B. Tseng, "A machine-learned proactive moderation system for auction fraud detection," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, (New York, NY, USA), p. 2501–2504, ACM, 2011.

[11] F. Manjoo, "EBay fraud law: Any takers?." http://www.wired.com/politics/law/news/2001/06/44831, June 2001.

[12] X. Liu, T. Kaszuba, R. Nielek, A. Datta, and A. Wierzbicki, "Using stereotypes to identify risky transactions in internet auctions," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pp. 513 –520, Aug. 2010.

[13] B. Gavish and C. Tucci, "Fraudulent auctions on the internet," *Electronic Commerce Research*, vol. 6, pp. 127–140, Apr. 2006.

[14] D. G. Gregg and J. E. Scott, "The role of reputation systems in reducing on-line auction fraud," *International Journal of Electronic Commerce*, vol. 10, no. 3, pp. 95–120, 2006.

[15] V. Almendra and D. Schwabe, "Analysis of fraudulent activity in a brazilian auction site," in *Proceedings of the 16th international conference on World Wide Web, Latin American Alternate Track*, WWW 2009, (Madrid, Spain), 2009.

[16] L. Zhang, J. Yang, and B. Tseng, "Online modeling of proactive moderation system for auction fraud detection," in *Proceedings of the 21st international conference on World Wide Web*, WWW '12, (New York, NY, USA), p. 669–678, ACM, 2012.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, p. 321–357, 2002.

[18] MercadoLivre, "Sobre MercadoLivre (About MercadoLivre)." http://www.mercadolivre.com.br/institucional, Sept. 2011.

[19] V. Almendra and D. Enachescu, "A supervised learning process to elicit fraud cases in online auction sites," in *Proceedings of the 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2011)*, (Timisoara, Romania), pp. 168–174, IEEE Computer Society, 2011.

[20] V. Almendra and B. Roman, "Using exploratory data analysis for fraud elicitation through supervised learning," in *13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2011)*, (Timisoara, Romania), pp. 251–254, 2011.

[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY: Springer New York, 2009.

[22] G. Ridgeway, *gbm: Generalized Boosted Regression Models*. 2012. R package version 1.6-3.2.

[23] R. D. C. Team, *R: A Language and Environment for Statistical Computing*. 2012. ISBN 3-900051-07-0.