Finding the needle: a risk-based ranking of product listings at online auction sites for non-delivery fraud prediction

V. Almendra[a]

[a] *University of Bucharest, Strada Academiei 14, Bucharest - sector 1, Romania 010014*

Abstract

Non-delivery fraud is a recurring problem at online auction sites: false sellers that list nonexistent products just to receive payments and afterwards disappear, possibly repeating the swindle with another identity. In our work we identified a set of publicly available features related to listings, sellers and product categories, and built a machine learning system for fraud prediction taking into account the high class imbalance of real data and the need to control the false positives rate due to commercial reasons. We tested the proposed system with data collected from a major Brazilian online auction site, obtaining good results on the identification of fraudsters before they strike, even when they had no previous historical information. We also evaluated the contribution of category-related features for fraud detection. Finally, we compared the learning algorithm used (boosted trees) with other state-of-the-art methods.

Keywords: Fraud detection, Non-delivery fraud, Boosted trees, E-commerce, Online auction sites, Machine learning, Data collection

1. Introduction

Online auction sites like EBAY offer unprecedented business possibilities for sellers and buyers through the creation of virtual marketplaces of global reach. Criminals also realized the opportunities opened by such virtual marketplaces. Among the several types of fraudulent behavior that take place in

Email address: vinicius.almendra@gmail.com (V. Almendra)

online auction sites, the most frequent one is non-delivery fraud (Gavish & Tucci, 2008; Gregg & Scott, 2008): fake sellers list nonexistent products for sale, receive payments and disappear, possibly reentering the market with a different identity. According to the Internet Crime and Complaint Center (2011), non-delivery fraud is the fourth most reported Internet crime. 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. (Chang & Chang, 2011). In other words, for a given product listing they need to *predict* whether or not it will end up being a fraud case. 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 techniques.

In this paper we will present a system for predicting non-delivery fraud that takes as input a set of product listings of an online auction site and outputs for each listing a fraud score, which can be used to analyze listings in decreasing order of risk. It also chooses a risk threshold so as to satisfy the user constraint on the rate of false positives. The proposed system uses a combination of features from product, seller and category, and, unlike other systems in the literature, it depends neither on historical data nor on social networks about the sellers in question, which is an advantage when dealing with fraudsters without reputation. The features we used can be extracted from the public web pages of online auction sites, which means that our system could be implemented by a third party, without the need of internal information. We evaluated the proposed system using data collected from a major Brazilian online auction site.

In Section 2 we will present the context for our research; in Section 3 we will describe the dataset used to validate our approach and will present the selected features; in Section 4 we will explain our proposed system for predicting non-delivery fraud; in Section 5 we will present the experimental results, and in Section 6 we will discuss them.

2. Background

Bolton & Hand (2002) 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: the high number of cases to be analyzed, the need of fast algorithms, uneven class sizes (class imbalance), uneven misclas-

sification cost, and the problem of false positives. Although they did not mention fraud at online auction sites, these challenges also apply.

There are recently published papers specifically focused on fraud at online auction sites, some from a descriptive perspective (Gavish & Tucci, 2006; Gregg & Scott, 2006; Almendra, 2012), and others aiming fraud prediction (Chang & Chang, 2011; Chau & Faloutsos, 2005; Chiu et al., 2011; Maranzato et al., 2010; Pandit et al., 2007; Zhang et al., 2011, 2012; Almendra & Enachescu, 2012). Fraud prediction systems need to tackle the problems of *feature extraction* and *method selection*. Regarding feature extraction, some works relied on public information obtained from online auction sites portals' (Liu et al., 2010; Chau & Faloutsos, 2005; Chang & Chang, 2011; Pandit et al., 2007); some used features related to seller past transactions e.g. average product price in the last 15 days (Chau & Faloutsos, 2005; Chang & Chang, 2011; Liu et al., 2010); others used information extracted from the social network surrounding sellers (Pandit et al., 2007); one made use of time-related variations of seller behavior (Chang & Chang, 2011). In our research we included 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 (Liu et al., 2010), although with much fewer features. There are also works that used internal information of online auction sites (Zhang et al., 2011; Maranzato et al., 2010; Zhang et al., 2012), 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 (Chau & Faloutsos, 2005; Chiu et al., 2011), Markov random fields (Pandit et al., 2007), instance-based learners (Chang & Chang, 2011), logistic regression (Zhang et al., 2011), online probit models (Zhang et al., 2012), Adaptive Neuro-Fuzzy Inference System (Lin et al., 2012). The present work uses a variant of boosted trees (Friedman, 2001) as its learning algorithm and compares its performance with several others well-know machine learning techniques.

Another problem related to fraud detection is *class imbalance*: the number of instances of the "positive" class (in our case, fraudulent) is much smaller than the number of instances in the "negative" class (in our case, legitimate). Class imbalance is an obstacle for the use of supervised learning systems in fraud prediction (Bolton & Hand, 2002), since algorithms tend to priv-

ilege the prevalent class (in our 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 Oversampling Technique) (Chawla et al., 2002). Some of the above-mentioned works used undersampling (Chau & Faloutsos, 2005; Chang & Chang, 2011), one uses an unsupervised model (Pandit et al., 2007), others did not state the approach adopted (Zhang et al., 2012, 2011; Maranzato et al., 2010).

3. Dataset description

We already described the dataset used in a previous work (Almendra & Enachescu, 2012). We reproduced the description here for sake of completeness, making some small improvements.

3.1. 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. From now on we will refer to MercadoLivre as MELI. In the whole year of 2011 we crawled daily 11 categories of products where we expected more fraud occurrence, extracting information about 2 million product listings. Using a previously developed methodology (Almendra & Enachescu, 2011), we found 1018 listings with clear signs of non-delivery fraud. Of these 1018, we identified 439 listings about which we had enough information for early fraud prediction. These 439 listings were labeled as *fraudulent listings*. All other listings of active sellers were labeled as *legitimate listings*. Notice that we did not include in our analysis listings of sellers sanctioned by MELI due to other kinds of misbehavior (misrepresentation, fee stacking, unpaid taxes etc.).

3.2. Features for fraud prediction

Our unit of observation was the *product listing*, so our features were also directly or indirectly linked to it. The directly linked features were *price*, *date* (when the listing was published), *product category* and *seller* (MELI's user who owned the listing). We also included information related to the seller: *reputation score*, *account age* (how old the seller account was, in days), and *number of recent transactions*. The values of these features were the ones collected at the day the listing was published in MELI's site (the value of *date* feature), since we wanted to predict fraud *before* transactions took place and

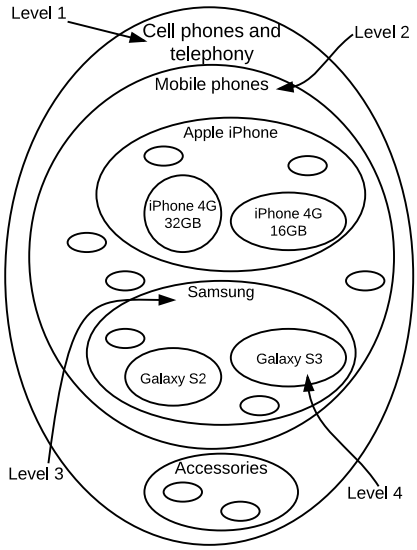


Figure 1: Excerpt of MELI's category hierarchy as a Venn diagram

before any sign of suspicion. This was possible because we did a longitudinal data collection.

We also included features related to the *product category*, since we expected that fraudsters would not choose randomly which products to list. Product categories specify the type of products, their models, characteristics etc., and sellers have to choose the category in which their products will be listed. Category-related features give us a chance to evaluate a product listing in a wider context. Two roughly equivalent product listings do not necessarily have the same risk of being fraudulent if they belong to very distinct categories.

We used our dataset to obtain aggregated measures about product categories over the entire year of 2011. All listings that shared the same category had the same values for these measures.

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

Regarding the product category of the listing (i.e. the one it was as-

Table 1: Summary of the dataset's features

Entity	Feature
Product listing	price, date, category, relative price difference
Product listing's seller	reputation, account age, number of recent transactions
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

signed to by the seller), 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 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 represented 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*, those features were calculated for the category *Cell phones and Telephony > Mobile phones > Apple iPhone.*

Finally, we calculated 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 1 summarizes the features used.

4. Fraud prediction system

The proposed Fraud prediction system takes as inputs a set of labeled listings (training data), a set of unlabeled listings (new data), and assigns a fraud score and a label (fraudulent or legitimate) to each unlabeled listing. Although the labels should be enough, the fraud score is useful when one cannot afford to check all listings labeled as fraudulent and wants to concentrate his efforts on the listings with highest scores.

The proposed system combines a well-known supervised learning model – boosted trees – with other techniques that improve the expected true positives rate and give the user a finer control over the expected false positives rate. In the next subsections we will describe the main elements of our system – boosted trees with resampling, score propagation and threshold optimization –, and finally will show how they are combined to build the fraud prediction system. From now on we will refer to the false positives rate (*fall-out* or $1 - specificity$) as FPR, and to the true positives rate (*recall* or *sensitivity*) as TPR. We will use the term *fraud score* to designate the numerical result of the algorithms applied to a listing. This score lies in the interval $[0, 1]$ and a higher score means a higher probability of being fraudulent.

4.1. Boosted trees with resampling

The idea of boosted 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 (Hastie et al., 2009). 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 (Ridgeway, 2012). Among the several options of loss functions, we opted for the Bernoulli one, since it is recommended for classification tasks.

One important issue was the composition of the training set, which is high imbalanced (many legitimate listings for each fraudulent one). Instead of directly using it to build the model, we used the undersampling technique: we generated a new training set containing all fraudulent listings available for training and a subset of the legitimate listings chosen through sampling without replacement. In the previous version of our system (Almendra & Enachescu, 2012) we varied the degree of imbalance in order to achieve the desired combination of true/false positive rates. In the present work we opted

to always use a balanced training set (equal number of examples for each class) and to rely on a single-class validation set to estimate the threshold that yields the desired false positives rate (see Section 4.3).

Since the majority of legitimate 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 *resamples*, where each resample contained all the fraudulent listings available for training and the same number of legitimate listings chosen randomly from the available ones. This way all resamples shared the same fraudulent listings, but with different subsets of the legitimate ones. Then we trained a new model with each one, applied all models to the new data, and finally we averaged the scores obtained for each example.

Notice that the features *seller* and *date* (of the listing) were not used in the boosted trees models. They were used only for the score propagation algorithm (Section 4.2).

4.2. Score propagation

The previous phase classifies product listings independently. Once a threshold is selected, a fraction of the evaluated listings is classified as fraudulent. When a listing is labeled as fraudulent, this means that the seller who owns the listing is also "labeled" as a fraudster, at least in the moment the listing was posted. When a certain listing is indeed fraudulent, other active listings from the same seller are probably also fraudulent, since fraudsters frequently list several products at once.

Zhang et al. (2011) proposed an idea inspired on Multi-instance Learning: when a listing is classified as fraudulent, all other listings of the same seller posted on the same day should also be classified as fraudulent. We extended this idea in two different ways. First, given a listing classified as fraudulent, we considered as fraudulent all other listings of the same seller posted in the previous 7 days. In the case that one of these listings was really fraudulent, some buyers might have already been swindled in this time interval, but at least some others could be spared, since MELI gives buyers 21 days to give feedback and fraudsters try to convince buyers to delay their feedback, in order to give them the chance to swindle more people.

The second extension was the way to update a listing's classification. In our previous work (Almendra & Enachescu, 2012) we simply "relabeled" as fraudulent the listings that fell in this situation. In the present work we were interested in ranking the listings in decreasing probability of fraud, so

instead of a "relabeling" we did a "rescoring": we propagated the score of the listings with higher fraud scores to the listings of the same seller posted in the previous 7 days. This procedure is described in Algorithm 1, where $maxDelay$ is the maximum temporal distance to do score propagation (the already-mentioned 7 days).

Algorithm 1 Score propagation algorithm

```
candidateFraud ← {l ∈ listings | score(l) > 0.5}
for all cf in candidateFraud do
    for all f in listings do
        if seller(f) = seller(cf)
        and score(f) < score(cf)
        and date(f) < date(cf)
        and date(cf) − date(f) < maxDelay then
            score(f) ← score(cf)
return listings
```

4.3. Threshold optimization

The scores produced by the boosting trees method could be converted to class labels directly using the threshold 0.5: all listings whose score was above 0.5 would be classified as fraudulent, while the others would be considered legitimate. Nevertheless, this approach may not work well in practice, since the number of legitimate listings is some orders of magnitude greater than the number of fraudulent ones. Although dangerous, fraudulent offers are like needles in the big haystack of product listings.

In a scenario where all listings labeled as fraudulent suffer automatically some kind of restriction (e.g. obliging sellers to furnish additional documentation), the problem would be dissatisfaction among legitimate users. In a scenario where all listings labeled as fraudulent were checked by human experts, another problem happens: too many listings would be selected for manual verification. We will call FPR_{max} the maximum FPR tolerated by the online auction site.

The common point among these scenarios is the existence of a constraint on the number of listings that can effectively be treated as suspect, while the remaining ones will be simply considered legitimate. The natural way to achieve this is to rank listings according to the risk of being fraudulent and to apply a threshold: all listings with a fraud score above that threshold

are classified as suspect. However, a question remains: how to calculate this threshold in order to keep the number of listings classified as suspect under control.

Since the number of fraudulent listings is proportionally very small (less than 1%), the number of listings classified as fraudulent will be dominated by the false positives, so we opted to limit the FPR to be lower than a certain FPR_{max}. This way the problem is reduced to finding the threshold t that solves the following:

$$\Pr\left(Score > \mathrm{t} | isFraud = 0\right) = FPR_{max}$$

Similarly to other parameters related to learning algorithms, we could find this threshold through a validation set or through cross-validation. However, we chose a much simpler and faster solution, again due to the prevalence of legitimate listings: we took the trained models and applied them *to the whole set of legitimate listings* available for training. Since each model was trained with just a very small fraction of this whole set, the results are not significantly biased. But the main advantage is the abundance of legitimate listings, because in this case we can safely assume that the empirical distribution of fraud scores for legitimate listings will approximate very well the true distribution. In the end, we obtained the best threshold by ranking the fraud scores for the whole set of legitimate listings and taking the score at the quantile $1 - FPR_{max}$.

4.4. Putting all together

In Figure 2 we depict an overview of the proposed system. The *training phase* outputs the boosted trees classification models and the calculated threshold in order to meet the expected false positives rate. These data are used in the production phase in order to classify new listings.

Algorithm 2 shows the pseudo-code for the training phase, and Algorithm 3 shows the one of the production phase. The score propagation algorithm (see Algorithm 1) is used both in the training and in the production phase. Fra_t is the set of fraudulent listings in the training set; Leg_t is the set of legitimate listings in the training set; *test* is the new data, containing both fraudulent and legitimate listings; *mod* is a list of trained models; *thr* is the estimated optimal threshold.

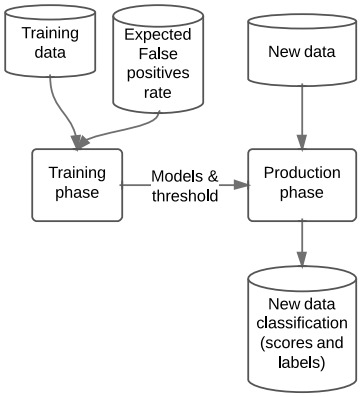
Algorithm 2 Training phase

{Trains boosted trees models}
for $i = 1$ **to** $n_{resamples}$ **do**
  $s \leftarrow rndSubset\,(Leg_t, size\,(Fra_t))$
  $resample \leftarrow Fra_t \cup s$
  $mod\,[i] \leftarrow trainBoost\,(resample)$
{Applies boosted models to the training set, for threshold optimization}
for $i = 1$ **to** $n_{resamples}$ **do**
  $score_t\,[i] \leftarrow predict\,(Leg_t, mod\,[i])$
$scoreAv_t \leftarrow mean\,(score_t)$
{Applies score propagation to the training set, for threshold optimization}
$scorePr_t \leftarrow scorePropagation\,(scoreAv_t, Leg_t)$
{Estimates threshold for the desired FPR_{max}}
$thr \leftarrow quantile\,(scorePr_t, 1 - FPR_{max})$
return *mod*,*thr*

Algorithm 3 Production phase

for $i = 1$ **to** $n_{resamples}$ **do**
  $score\,[i] \leftarrow predict\,(test, mod\,[i])$
{Model averaging}
$scoreAv \leftarrow mean\,(score)$
{Score propagation}
$scoreAvPr \leftarrow scorePropagation\,(scoreAv, test)$
{Final labeling using threshold}
for $k = 1$ **to** $size\,(test)$ **do**
  if $scoreAvPr\,[k] > thr$ **then**
    $label\,[k] \leftarrow$ **fraudulent**
  else
    $label\,[k] \leftarrow$ **legitimate**
return *label*, *scoreAvPr*

Figure 2: Fraud prediction system overview



5. Experimental results

5.1. Train and Test Sets

From the dataset described in Section 3 we created the training and test sets. Their distributions are shown in Table 2. Since many sellers (including fraudsters) post multiple listings, we took care that all listings of each seller were included either in the training or in test set, so as to not artificially improve results.

Table 2: Training and test sets

	Training set	Test set
Fraudulent	326	113
Legitimate	21,422	21,914

5.2. Performance measures

The main performance measures used were TPR and FPR. Since these measures depend on the chosen threshold, we also used Area under the ROC curve – AUC –, since it evaluates the overall ability of the classifier to rank fraudulent listings higher than legitimate ones. For more information, see the work of Fawcett (2006). We further included in the results *Precision*, *Accuracy* and *F-measure*, since they are commonly used in the literature. Although popular, these measures can be misleading when dealing with fraud prediction under class imbalance.

Table 3: Overall results

Score propagation?	FPR_{max}	TPR	FPR	Precision	Accuracy	F-measure	AUC
Yes	Without	**95.6%**	**17.2%**	**84.7%**	**82.8%**	**89.8%**	**0.95**
	10%	**83.2%**	**10.8%**	**88.5%**	**89.2%**	**85.8%**	
	5%	**70.8%**	**5.2%**	**93.1%**	**94.6%**	**80.4%**	
	1%	**48.7%**	**1%**	**97.9%**	**98.7%**	**65%**	
No	Without	86.7%	14.7%	85.5%	85.3%	86.1%	0.941
	10%	78.8%	10%	88.7%	89.9%	83.4%	
	5%	62.8%	5.2%	92.3%	94.6%	74.8%	
	1%	44.2%	0.9%	97.9%	98.8%	60.9%	

Since online auction sites need usually a low false positives rate, we will concentrate our analysis in the zone of the ROC curve where $FPR \leq 10\%$ using partial AUC, i.e. the area under the portion of the ROC curve with $FPR \in [0, 0.1]$. This measure shows how well the classifier pushes fraudsters to the top of the rank.

5.3. Results

We first evaluated the performance of the full fraud prediction system for different values of FPR_{max} and for both with and without score propagation, to highlight the improvement achieved by this algorithm. Table 3 shows the performance measures. Differences between AUC's are statistically significant with $p < 0.001$. Figure 3 depicts the ROC curve, displaying the thresholds for the different FPR_{max} targets.

We compared boosted trees with other learning algorithms, in order to see if it was indeed a good choice. We made a single balanced training set (one resample) and trained all models using it. We applied all models on the test set, without threshold optimization and without score propagation, since these can be applied and potentially improve any model. Table 4 displays for each method the AUC and partial AUC for $FPR \leq 10\%$. The differences between boosted trees and random forests were not statistically significant, while the difference to the other methods was significant with $p < 0.001$.

To test the effect of the features related to product category, we created two new pairs of training/test sets from the full sets: one *without* features related to product category and another one *with only* features related to product category. We tested the system on these three datasets (the original

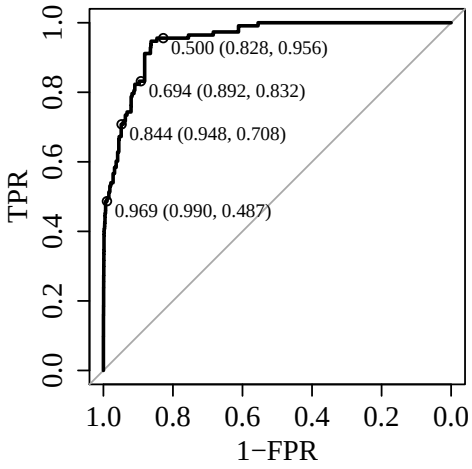


Figure 3: ROC curve with some points highlighted. The number near each point is the threshold the yields the pair $(1 - FPR, TPR)$

Table 4: Comparing different methods

Method	partial AUC ($FPR \leq 10\%$)	AUC
Random forests	**0.060**	**0.939**
Boosted trees	**0.060**	**0.934**
C5.0 decision trees	0.047	0.893
Logistic regression	0.041	0.892
Neural networks	0.039	0.879
SVM	0.032	0.853
k-nearest neighbor	0.032	0.881

Table 5: Effect of product category features

Set	Partial AUC ($FPR \leq 10\%$)	AUC
Without category-related features	0.054	0.84
Only category-related features	0.039	0.86
All features	0.067	0.94

Table 6: Performance for sellers with zero reputation

	Partial AUC ($FPR \leq 10\%$)	AUC
Reputation > 0	0.064	0.948
Reputation $= 0$	0.078	0.945

one with all features and these new two). Results are shown in Table 5. The differences are statistically significant with $p < 0.02$.

We evaluated the performance of the system on the subset of listings whose seller reputation is zero. Existing methods that use sellers' historical or social network information cannot be applied in this case. We split the test set in two: one with sellers without reputation (33 fraudulent listings and 1605 legitimate ones) and one with all other sellers (80 fraudulent listings and 20,309 legitimate ones). In Table 6 we display the performance of the method on these two subsets. The differences found were not statistically significant.

6. Discussion

6.1. Results analysis

The overall results were positive, since the proposed system successfully identified which were the fraudulent listings in the test set. The AUC measure of 0.95 means that the system would rank a random fraudulent listing above a random legitimate listing with probability of 95%. From a practical point of view, the combination of 83% of true positives rate with 11% of false positives rate allows one to implement a fraud prevention process where the top-ranked listings would be automatically suspended, the middle-ranked would go under review and the lower-ranked would suffer restrictions e.g. on the number of simultaneous transactions.

It is worth to remember that the information used for training/test refers to the state of the listings close to the moment they were published. This

means that the system would have stopped fraudulent behavior at its inception. And it is also worth of notice the fact the we used only publicly available information, which is surely much poorer than the internal one.

The problem of class imbalance was successfully tackled with the combination of undersampling and threshold optimization, achieving results comparable to the ones obtained in a previous work (Almendra & Enachescu, 2012), but using a much smaller training set. For example, to obtain the results for $FPR_{max} = 1\%$, in the present work we used in each resample 652 listings, while in the previous work mentioned we needed 30 times more (10,106 listings). Threshold optimization also worked as expected, yielding FPR values on the test set very close to the desired FPR_{max}. Threshold selection is an important aspect for fraud prediction systems (Bolton & Hand, 2002), since generally there are many trade-offs involved and F-measure or accuracy maximization are not necessarily the best criteria.

The score propagation algorithm improved remarkably the TPR metric (around 5% more). The propagation of scores instead of labels brought two important advantages: (i) the output is a ranked set of listings, instead of just a labeling, (ii) threshold optimization can be applied *after* score propagation, which allows us to enforce FPR_{max}, something that did not happen in our previous work.

Boosted trees performed better than the other learning algorithms tested, with the exception of random forests, whose AUC and partial AUC were not statistically significant different from the ones of the boosted trees. This confirms this method as a good choice for the problem being discussed, although one should also consider the possibility of using random forests.

Features of product categories improved classification results in a statistically significant way. This result confirms their importance for the problem of fraud prediction. In fact, product features and category features were somehow complementary, since category-related features were better in general but product features were better in the $FPR < 10\%$ region.

The method's performance on listings belonging to zero-reputation sellers was not significantly different from the one with listings belonging to sellers with reputation above zero. This is an important result, since many existing fraud prediction methods rely on historical information and as such are unable to predict fraud when a seller has no reputation. This opens space for simple "hit-and-run" fraud attempts. Naturally, sellers without reputation have less chances of attracting victims, albeit this does not stop them: in our dataset, 125 of the 439 fraudulent listings (28%) belonged to sellers with

zero reputation.

One limitation of the proposed system is the need of periodical re-training in order to keep pace with new patterns of both fraudulent and legitimate listings. For our dataset this was not an issue, but it might be for bigger ones.

6.2. Future work

The proposed method could be combined with other approaches that use historical and social network information, since the features we used are somehow complementary to theirs.

Another direction is the use of the textual description associated with product listings, since they also convey information about the listing's author.

A third interesting extension would be the inclusion of *expected fraud impact* in the algorithm, that is, what is the potential damage that can be inflicted by a seller if he turns out to be a fraudster. This would allow us to rank suspect listings in a wiser way.

Acknowledgments

This work was sponsored by University of Bucharest, under postdoctoral research grant nr. 15515 of September 28th, 2012. I would like to thank Prof. Dr. Denis Enăchescu for his support on issues related to statistical analysis, and Oscar Ruiz and Radu Ionescu for proofreading.

References

Almendra, V. (2012). A comprehensive analysis of nondelivery fraud at a major online auction site. *Journal of Internet Commerce*, *11*, 309–328. doi:10.1080/15332861.2012.729469.

Almendra, V., & Enachescu, D. (2011). 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)* (pp. 168–174). Timisoara, Romania: IEEE Computer Society. doi:10.1109/SYNASC.2011.15.

Almendra, V., & Enachescu, D. (2012). A fraudster in a haystack: Crafting a classifier for non-delivery fraud prediction at online auction sites.

In *Proceedings of the 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012)*. Timisoara, Romania: IEEE Computer Society.

Bolton, R., & Hand, D. (2002). Statistical fraud detection: A review. *Statistical Science*, *17*, 235–255.

Chang, W.-H., & Chang, J.-S. (2011). A novel two-stage phased modeling framework for early fraud detection in online auctions. *Expert Systems with Applications*, *38*, 11244–11260. doi:10.1016/j.eswa.2011.02.172.

Chau, D. H., & Faloutsos, C. (2005). Fraud detection in electronic auction. In *Proceedings of European Web Mining Forum*. URL: http://www.cs.cmu.edu/~dchau/papers/chau_fraud_detection.pdf.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chiu, C., Ku, Y., Lie, T., & Chen, Y. (2011). Internet auction fraud detection using social network analysis and classification tree approaches. *International Journal of Electronic Commerce*, *15*, 123–147. doi:10.2753/JEC1086-4415150306.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*, 861–874. doi:10.1016/j.patrec.2005.10.010.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*, 1189–1232. doi:10.2307/2699986.

Gavish, B., & Tucci, C. (2006). Fraudulent auctions on the internet. *Electronic Commerce Research*, *6*, 127–140. doi:10.1007/s10660-006-6954-0.

Gavish, B., & Tucci, C. (2008). Reducing internet auction fraud. *Communications of the ACM*, *51 (5)*.

Gregg, D. G., & Scott, J. E. (2006). The role of reputation systems in reducing on-line auction fraud. *International Journal of Electronic Commerce*, *10*, 95–120.

Gregg, D. G., & Scott, J. E. (2008). A typology of complaints about ebay sellers. *Communications of the ACM*, *51 (4)*, 69–74.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. New York, NY: Springer New York.

Internet Crime and Complaint Center (2011). *Internet Crime Report*. Technical Report. URL: `http://www.ic3.gov/media/annualreport/2011_IC3Report.pdf`.

Lin, S.-J., Jheng, Y.-Y., & Yu, C.-H. (2012). Combining ranking concept and social network analysis to detect collusive groups in online auctions. *Expert Systems with Applications*, *39*, 9079–9086. doi:`10.1016/j.eswa.2012.02.039`.

Liu, X., Kaszuba, T., Nielek, R., Datta, A., & Wierzbicki, A. (2010). Using stereotypes to identify risky transactions in internet auctions. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on* (pp. 513 –520). doi:`10.1109/SocialCom.2010.81`.

Maranzato, R., Pereira, A., Lago, A. P. d., & Neubert, M. (2010). Fraud detection in reputation systems in e-markets using logistic regression. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (pp. 1454–1455). Sierre, Switzerland: ACM. doi:`10.1145/1774088.1774400`.

Pandit, S., Chau, D. H., Wang, S., & Faloutsos, C. (2007). 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.

Ridgeway, G. (2012). *gbm: Generalized Boosted Regression Models*. URL: `http://CRAN.R-project.org/package=gbm` r package version 1.6-3.2.

Zhang, L., Yang, J., Chu, W., & Tseng, B. (2011). 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 (p. 2501–2504). New York, NY, USA: ACM. doi:`10.1145/2063576.2064002`.

Zhang, L., Yang, J., & Tseng, B. (2012). Online modeling of proactive moderation system for auction fraud detection. In *Proceedings of the 21st*

international conference on World Wide Web WWW '12 (p. 669–678). New York, NY, USA: ACM. doi:10.1145/2187836.2187927.