ProtoTrust: An Environment for Improved Trust Management in Internet Auctions

Tomasz Kaszuba, Piotr Turek, Adam Wierzbicki, and Radoslaw Nielek

Polish-Japanese Institute of Information Technology*
Warsaw, Poland
kaszubat@pjwstk.edu.pl

1 Introduction

Internet auctions are used everyday by millions. However, despite frequent criticism, only the most simple reputation systems are used by the most popular Internet auctions today. As a consequence of this, an experienced auction user is forced to undergo the menial task of reading and judging comments about his potential transaction partners. While it is true that the human mind is the best possible method of evaluating this information, the task is time-consuming and error-prone: the sheer number of comments is sometimes an obstacle to making a good decision under uncertainty. An inexperienced auction user, on the other hand, is often daunted by the task of understanding the information provided to him. It takes some learning to understand the value of a negative feedback, to evaluate the contents of the comments, or to understand what it means that the auction system has returned the handling fee. The reason for this situation may be the fact that the management of auction sites uses other mechanisms, like auction insurance or escrow, to protect users against outright fraud. And, as has been argued by the management of auctions sites when we have had an opportunity to discuss the issue, the simplicity of the presently used reputation system is an added bonus: it creates the impression of a simple, easy-to-understand tool. The fact that this simple tool is vulnerable to several adversary strategies [1,2,3] and that its design has an adverse impact on the reporting behavior of users [4,5] is not a sufficient argument for a change.

We have attempted to find a way out of this situation. Our goal has been the improvement of trust management for Internet auction users. In our view, the trust management (TM) system should have as a goal the simplification of the users' search for relevant information, reducing the time complexity of the task of browsing through all relevant feedback. Also, the TM system should increase the safety and comfort of the user by providing additional information not available on auction sites today, based on algorithms that are better suited to the current auction design. How could we achieve these goals without the cooperation of the managers of the auction sites? To solve this problem, we have designed an extension for a popular Web Browser (Firefox[1]) that gives users access to our

* The work reported in this paper has been funded by the Polish Ministry of Science and Higher Education under the research grant N N516 4307 33.

[1] http://www.mozilla.org/

algorithms. The algorithms themselves are part of a library of trust management tools developed in the uTrust[6] project. The extension obtains its information by automatically performing the task that is performed by an auction user: by crawling parts of the auction site. Using the extension, user can input her preferences into our Trust Management system and select the accuracy she want to obtain. The results of the crawling provide input information for the improved trust management algorithms. The user is presented with a graphical interface that gives access to a wealth of information that should support her in making the right decision. At all times, the user still has access to the original information provided by the auction site.

We have decided to choose simple, yet useful algorithms for the first suite of tools provided by our extension. These algorithms attempt to solve the following problems: how should the various comments be classified and evaluated? How should reputation be calculated so that it takes into account the price context, and category context? How can a user evaluate reputation in a system where feedback is frequently missing? Finally, we decided to implement an algorithm that would give users a completely different trust management tool. Rather than calculating reputation, we calculated risk (valued amount of money that can be lost if the seller is fraudulent).

The algorithms described in this paper have all been implemented and made available to real Internet auction users. This is made possible by our Firefox extension, which is an environment for implementing new trust management algorithms for Internet auction users. (In the future, we hope that other researchers will join us in the enrichment of these tools through their implementation in the uTrust library.) Further, we have tested our algorithms on extensive traces of Internet auction use. The traces contain rich information, including the entire comment and contextual information about the auction.

The rest of this paper is organized as follows: in the next section, we discuss the architecture of the extension and of the uTrust library. Section three concerns the algorithms that have been implemented in the first version of the extension. Section four describes the evaluation of the proposed algorithms using traces of Internet auctions. Section five concludes and presents ideas for future work.

2 Related work

In the area of Internet Fraud, most of recent work has been focused on the seller's profile [7,8]. Much work has been devoted to inducing users to behave properly [9,8] as well as detecting fraudulent users [2,1]. The work of Dellarocas [9] applies in situations where users can intentionally give unfair ratings to each other. The authors have proposed to conceal the identities of buyers and sellers to prevent such discrimination. Gavish and Tucci [3] have presented the seller's swindling methods in Internet auctions. Gregg and Scott [7] have proposed a model of complaints against sellers.

Currently, some stand-alone applications aim to solve a similar problem [2][10], but as long as they have no integration with web browsers they are not user friendly. Since ProtoTrust uses the Firefox extension mechanism to install and update itself, it is easily accessible to all potential on-line auction users.

3 ProtoTrust Architecture

We have decided to implement our solution as an extension of Firefox. Integration with web browser has several advantages such as availability (system independent), easy installation and access to information directly from the website. ProtoTrust consists of three major modules: the thread management module, the presentation module and the Trust Management (TM) module. Figure 1 shows the ProtoTrust architecture in details.

After the initialization of the extension in the web browser the user can add her preferences to tune the system. Preferences can be changed at any time even after computation. Extension is activated by specifying context and target user (visit the item page). Thread management initiate all objects and synchronizes all crawling and presentation threads with the web browser's main thread. This module also controls the network load and the memory load of the system.

The TM module is the most crucial element in the ProtoTrust. It accesses the network and uses local storage (if available) in the search for information that will be used to support the user. New information is obtained by crawling Internet auction sites. Since the crawling is done by the browsers of auction users, it cannot be banned by the auction provider. Our implementation of the TM module uses Jruby[2] and Hpricot[3] libraries to obtain the information from the Internet. The Web crawler starts crawling the Internet auction service in a manner similar to normal user behavior. It checks public information about the sellers that the user is interested in (the sellers' reputation, previous auctions, prices, comments) and also context information like average prices in the category, probability of failure for similar auctions. Every information found by Web crawler is stored in locally in order to future use.

After a fixed amount of time or when sufficient data is collected, ProtoTrust uses the uTrust library [6] of universal trust management methods to compute the desired TM algorithms. Results can be simple (like the amount of negative opinions or minimal price without fraud) or can depend on each other (for example to compute risk we need to compute respective probabilities first). Due to lack of space, we do not discuss the details of library design.

When the computation is complete, our system applies user preferences and presents the computed results to the user. The presentation module based on user preferences, can suggest the right decision to make. User can follow the suggestion or can change her preferences and thus fits the system into new condition.

---

[2] http://jruby.codehaus.org/
[3] http://code.whytheluckystiff.net/hpricot/

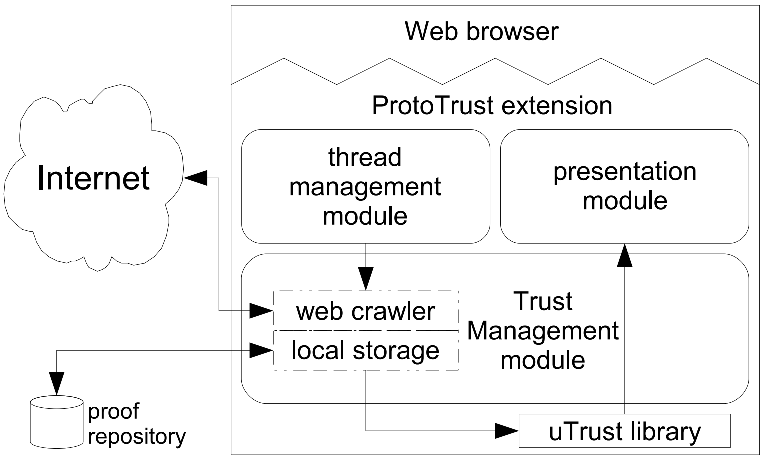

Fig. 1. ProtoTrust architecture

4 ProtoTrust algorithms

We have designed several methods to help the users with decision to buy an item from a certain seller or not. The methods currently implemented in ProtoTrust perform three tasks: they support the decision of the user based on various TM algorithms, they increase the amount of information available from the auction site by considering implicit non-positive feedbacks, and they perform a classification of auction comments to support the user in understanding them. In this paper we focus on the decision support algorithms that aim to directly recommend whether the user should buy an item from a specific seller. Two types of decision support methods are used by ProtoTrust: price dependent and probability dependent methods. Both types of methods take into consideration user preference such as *risk propensity* R_{prop} that is an amount of money that a user is willing to risk, or a *risk threshold* R_{thres} that is a threshold level for the probability methods. The exact meaning of these values depends on the used algorithm and will be explained further. Probability dependent methods are based on a proportion between count of negative feedback to all feedback collected by an agent. These methods are similar to original reputation methods used currently in on-line auction services. In addition, we formulate decision rules for all the algorithms that may help the user to make decision about participating in an auction or discarding it.

4.1 Comment classification

Our extension gathers the feedbacks for a seller taking into account the feedback age and context. We distinguish four age levels (starting with the smallest time period):

– feedbacks not older than 1 week
– feedbacks not older than 2 weeks
– feedbacks not older than 4 weeks (about 1 month)
– all feedbacks

We distinguish four context domains that partition the space of the auction category tree (starting with the smallest range):

– feedbacks for an agent in one category
– all feedbacks for an agent (from all categories)
– all feedbacks from one category
– all feedbacks from entire auction system

Both age and context can be used by the TM algorithms to select appropriate feedbacks. Another kind of feedback classification is based on feedback content. We have created regular expressions, that extract the comment from users' nonpositive feedbacks. In order to do such classification, we have designed the typology of users' complaints similar to [7]. Classification rules can specify the type of fraudulent seller (for example seller, who sends damaged goods or seller, who send the items too late). Our ProtoTrust environment can present outcome of such data mining as well as entire content of feedback. Due to lack of space we do not describe this work in detail.

4.2 Implicit negative feedbacks

In the real auction platforms like eBay, Taobao or Allegro not every transaction is followed by feedback[11]. This is due to many different reasons. Firstly, both parties, sellers and buyers, lack motivation to devote some time to leave an evaluation which is helpful for other users of the auction site but doesn't have direct influence on payoffs of the agents who have left this feedback. The second reason why users don't leave any evaluation is the fear of a reciprocated negative evaluation which probably appears as an answer to the publicly expressed dissatisfaction of service, quality of products or dishonesty or monkey business of either seller or buyer. The research conducted by Morzy and Wierzbicki [11] shows that in most cases lack of feedback can be explained by this effect. Thus, it's crucial for individual users/transactions for the purpose of successful prediction of the fraud probability to rediscover the silent meaning of the non-existent evaluation.

The aim to support users in their decision making process at a specific point in time raises yet another problem. For every moment in time there are many auctions unsupported by evaluation (sometimes only either buyer or seller has bothered to give feedback and sometimes neither of them). Typically, the number of absent feedbacks decreases over time and, on average, two weeks after the transaction, the most positive comments are already present in the auction system. Thus, we can presume that if the auction finished over two weeks ago ($t_{threshold}$) and it still has not been evaluated, and then the non-existent feedback should be treated as (at least) non-positive.

In the cases where one, either buyer's or seller's, evaluation is already present in the reputation management system the missing feedback can be inferred from the existing one, even without waiting until $t_{threshold}$ will be met. Every positive comment is almost in every case awarded by reciprocated positive evaluation. Proneness to answer with positive feedback on every positive comment regardless of transaction outcome is slightly stronger by sellers. Similar effect of adequate answer can be also observed for the negative evaluations.

Using the rules shown in Table 1 within rules-based system (ProtoTrust is one in which those rules have already been implemented) enables addressing virtually all situations and squeeze as much information from the crawled data as possible and as necessary to correctly predict future behavior of particular sellers and, thus, decrease the risk for honest users of the auction platforms.

Table 1. Rules deducted from the collected allegro trace and using in the ProtoTrust plug-in for prediction implicit negative feedbacks

Buyer	Seller	Reasoning
X	X	if $t_{now} - t_{transaction} > threshold$ then (NEG,NEG) or (X,X)
positive	X	(POS,POS)
negative	X	0.16 (NEG,POS), 0.02 (NEG,NEU), 0.82 (NEG,NEG)
X	positive	(POS,POS)
X	negative	0.05 (NEG,NEU), 0.95 (NEG,NEG)
Feedbacks types:		NEG - negative, NEU - neutral, POS - positive

4.3 Decision support methods

The main focus of this paper are decision support methods that use various TM algorithms to recommend the right decision to the user. In this section, we describe the TM algorithms implemented in uTrust for this purpose, and used in ProtoTrust.

Fraud probability Fraud probability $FraudProb$, or simply reputation, is the standard measure provided by any Internet auction service. It is a proportion between the number of negative feedbacks M and total number of feedbacks N.

We compute several variations of this measure that depend on the time class of the feedback. For seller s the fraud probability is defined as:

$$FraudProb_{st} = M_t/N_t \tag{1}$$

where M_t is amount of seller's s negative feedback not older than t, and N_t is total count of feedback not older than t.

ProtoTrust warns the user about this seller's auction when:

$$FraudProb_{st} > R_{thres} \tag{2}$$

This means that seller s has carried out too many fraudulent operation in time t. The value of R_{thres} is the threshold chosen by the user from the range of $[0, 1]$.

The fraud probability $FraudProb_c$ for every category c is computed in a similar manner, but we do not use it in this algorithm since we want to pick out every possible fraudulent seller. On the other hand, we use $FraudProb_c$ in other algorithms described below.

Reputation with price context Many researchers [4,5,12,13] have proven that the reputation of a seller is related to selling prices. Therefore we propose our measures which can be complementary to the standard reputation.

We propose to compute the weighted average price for the sellers' auctions. $AvgPrice_s$ - is the seller's s weighted average price, in which weights are dependent on the value of the auction's feedback. For each seller's auction i we multiply the final price P_i and the buyer's feedback value $FVal_i$. Weights are -1 for negative, 0 for neutral and 1 for positive feedback. Let n be the number of auction carried out by seller s. Average price for the seller s is given by the equation:

$$AvgPrice_s = \frac{\sum_{i=0}^{n}(FVal_i * P_i)}{n} \tag{3}$$

We are not willing to compute the average price for the entire category too often, due to the computational cost. Since most of the auction systems are mature markets, the value of the average price for a category does not change frequently. We compute the average price and standard deviation σ for each category using the full TM system information. These values may be computed once a fixed period of time and kept as constant values in the TM system.

When the user's $AvgPrice$ is significantly lower than the $AvgPrice$ in a context, there is a possibility that the seller will cheat in such an auction (by selling cheap items and not sending them to buyers or selling defective or illegal goods). Thus our system alerts the user by testing the seller s in a category c when:

$$AvgPrice_s + R_{prop} < AvgPrice_c \tag{4}$$

where R_{prop} is the *risk propensity* parameter defined by the user.

If we include the standard deviation σ we get:

$$AvgPrice_s + R_{prop} < AvgPrice_c + \sigma \tag{5}$$

However sometimes it is hard to point out a fraudulent seller basing only on his transaction history. Such sellers can establish a certain level of reputation before carrying out fraudulent auctions. Most of them gain reputation by selling many low-cost items. Note that one positive feedback from a €1000 auction is worth the same as from a €1 auction.

To protect from such cheating techniques we propose to compute the minimal price with a negative feedback $MinPriceWithNeg_s$. We should be wary of all offers from seller s which are much above the minimal price with some parameter R_{prop} which is the *risk propensity.* Our system alerts when the actual bid in an auction i is higher than the seller's s minimal price with a negative feedback. There is no reason to alarm when user has no negative feedback.

$$P_i - R_{prop} > MinPriceWithNeg_s \tag{6}$$

Risk The measures proposed above may be not understandable for an inexperienced user. Sometimes it is more convincing for a user to compute the amount of money she can lose if the seller is fraudulent. Our risk measure $Risk_i$ is the multiplication of the actual bid P_i by the fraud probability in a context. It is given by the equation:

$$Risk_i = P_i * FraudProb_c \tag{7}$$

We compare risk to the risk propensity R_{prop} that is the amount of money that a user wants to risk in an auction. A user can set her risk propensity value to tune the TM system to her preferences.

$$Risk_i > R_{prop} \tag{8}$$

Our system alerts when $Risk_i$ is greater than a user's risk propensity R_{prop}.

5 ProtoTrust algorithms evaluation

We have evaluated ProtoTrust using a real world dataset. The dataset has been acquired from *www.allegro.pl* that is the leading Polish on-line auction provider. In this service, each auction has an explicit deadline and all current bids are exposed to all participants. Moreover, all information about all participants is accessible. In most actions the bidders can specify a maximum price they want to pay for an item, and the proxy bid system automatically raises the bid, using only as much of the bid as is necessary to maintain the top position. Bidders can also increase their maximum price at any moment. When the auction terminates, the bidder with the highest bid wins. There are also multi-item (*Buy now!*) type auctions in which are sellers can sell more than one item (and hence there is more than one winner). In such auctions, every bid is a winning bid.

We have selected the subset of 9500 sellers and their 186000 auctions listed in 6300 categories. We have tested our decision support algorithms using all 328000 feedbacks that are sent by the buyers. The unequal amount of auctions and feedbacks is caused by existence multi-item type auctions.

5.1 Experiments

We have reimplemented some of the uTrust algorithms to work with our off-line data. To recreate the on-line environment, we have sorted the auctions according to the termination date. For each auction in the set we have computed all the algorithms using only the data that was available until that moment. After computing all algorithms we have tested if they are good predictors of the real feedback value. For each algorithm we store: the count of successful detections of negative feedbacks $True_{neg}$ (the accuracy of the algorithm), and the count of unsuccessful detection $False_{neg}$ (Type II error in statistics) of negative feedback.

5.2 Evaluation criteria

For the evaluation criteria of our algorithms we use two values: the probability of fraud detection FrD and frequency of alerts FoA.

Let N be the total number of feedbacks and M the total number of negative feedbacks. Fraud detection is given the by equation:

$$FrD = True_{neg}/M \tag{9}$$

and the frequency of alerts is given by:

$$FoA = \frac{True_{neg} + False_{neg}}{N} \tag{10}$$

We have also computed the difference between fraud detection FrD and frequency of alerts FoA. We have used the random classifier as a reference level. For the random decision the FrD and FoA are equal (for example if we choose to alert in 50% of cases we discover 50% of true negatives $True_{neg}$).

5.3 Evaluation results

We have evaluated the algorithms presented in the previous section from two different perspectives: probability dependent and price dependent. Results in each group depend on the user preferences (risk threshold R_{thres} and risk propensity R_{prop}). For better presentation, we have selected three best algorithms from each group. We have presented detailed results achieved by all algorithms in Table 2.

Probability dependent methods For probability dependent algorithms, we have run the experiment several times, changing the risk threshold parameter R_{thres}. R_{thres} is the acceptable probability of fraud and it is expressed in permils [‰]. On Figure 2 we present the effect of the risk threshold R_{thres} on detection of fraud FrD and frequency of an alert FoA.

Best fraud detection was achieved by the algorithm that used all available feedbacks. However, this algorithm also had a high frequency of alerts. Moreover, in a real situation we would not gather all historical data about the seller because of time and network usage.

Table 2. Best performance achieved by algorithms

$Algorithm_type$	R_{prop} $[PLN]$	R_{thres} [‰]	FrD	FoA	$Performance$ $(FrD$ - $FoA)$
$FraudProb_inf$	–	5	0.42	0.12	0.3
$FraudProb_2$	–	5	0.35	0.09	0.26
$FraudProb_4$	–	5	0.41	0.12	0.29
$Risk$	1	–	0.48	0.07	0.41
$AvgPrice$	19	–	0.45	0.18	0.27
$MinPriceWithNeg$	19	–	0.51	0.23	0.28

Similar results have been achieved using only feedbacks that are not older than 4 weeks. Using feedbacks that are at most 2 weeks old gives us a 10% lower detection rate, but also has a much lower frequency of alerts.

Both Fraud Detection and Frequency of Alerts decline linearly with increasing of risk threshold R_{thres}. When we increase the risk threshold, our system is less likely to alert the user about fraudulent sellers, because it accepts some sellers' negative feedbacks.

As shown in Table 2, the best trade-off between fraud detection and frequency of alerts was achieved when the risk threshold value was fixed at 5 ‰. This is because we observe a significant drop of the frequency of alerts and slight decrease of fraud detection when this value of the risk threshold is exceeded.

Price dependent methods Similarly to the previous methods, we have run the experiment several times, with different risk propensity R_{prop} parameter values.

We have selected three algorithms described in 3.3S. Figure 3 presents the fraud detection FrD and frequency of a alerts with regard to the risk propensity R_{prop}. Best results have been achieved by $Risk$. This algorithm has detected every fraudulent auction while it was alerting every second auction. With an increase of the risk propensity, the algorithm detects less fraudulent auctions. We can observe a drastic gap between fraud detection and frequency of alerts for $Risk$ when R_{prop} equals to 1 PLN (1 Polish Zloty is about €0,25). The algorithm can eliminate half of possible fraudulent auctions while alerting only in 7% of all offers. The two other algorithms provide similar a detection rate with a much higher frequency of alerts (about 25%). We have modified the $AvgPrice$ algorithm (described in 3.3) by including information about sellers' minimal price with a negative feedback. As a result, the $MinPriceWithNeg$ algorithm has a slightly better score than the original $AvgPrice$.

Performance (FrD - FoA) of the algorithms is presented on Figure 4. The best trade-off was achieved by $Risk$ with a risk propensity equal to 1. It is 41% better than the random algorithm. Detailed results are presented in Table 2.
The risk algorithm tends to be the most effective. It provides a good fraud detection rate with a very low frequency of alerts. When the risk propensity is

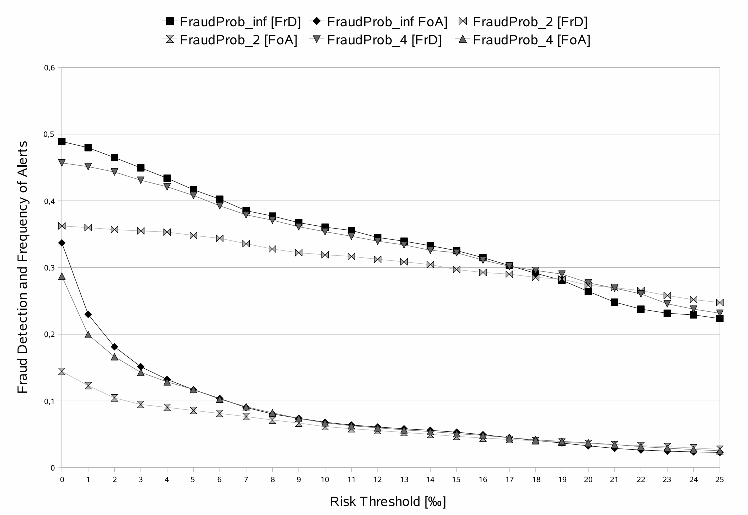

Fig. 2. Probability of Detection of Fraud FrD and Frequency of an Alert FoA with respect to the risk threshold R_{thres} [‰].

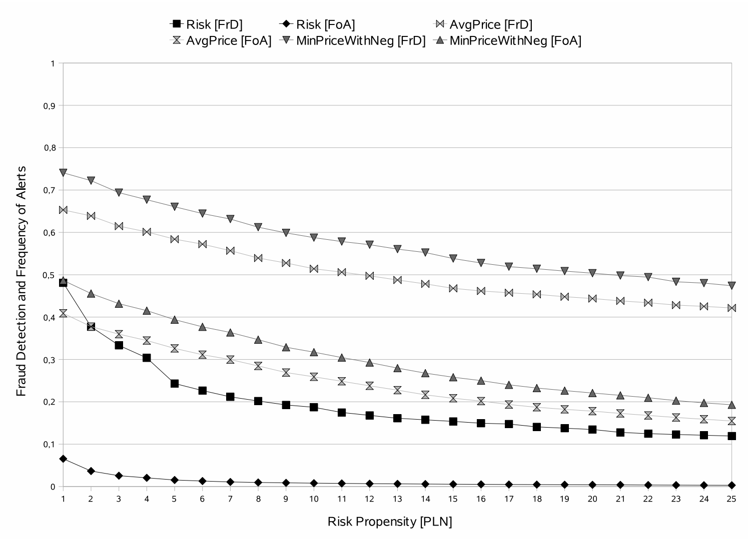

Fig. 3. Probability of Detection of Fraud FrD and Frequency of an Alert FoA in respect of risk propensity R_{prop} [PLN] (Polish zloty)

between 1 and 4PLN (€1), the risk algorithm performs better than any other presented algorithm. Using this measure we can warn user against almost half (48% to 30%) of the frauds before the they occur.

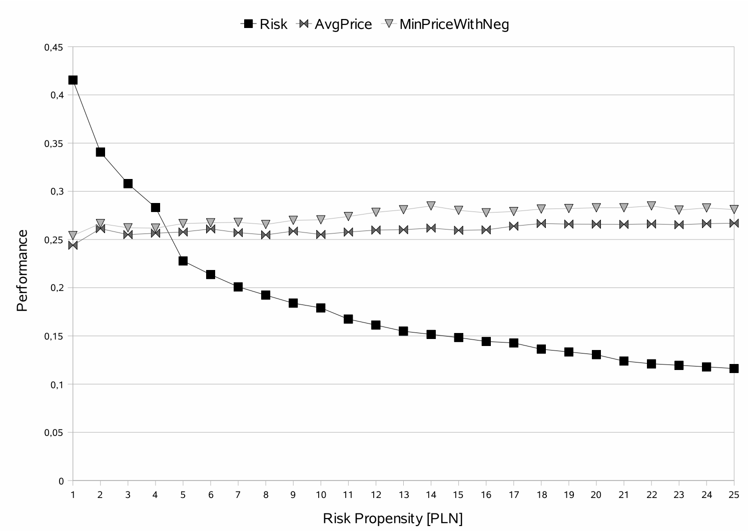

Fig. 4. Algorithm performance (FrD - FoA) in respect of risk propensity R_{prop} [PLN] (Polish zloty)

6 Conclusion and future Work

In this work, we have presented ProtoTrust, an environment for improving Trust Management for Internet auctions that operates independently of the auction providers. ProtoTrust allows to implement new Trust Management methods for real Internet auction users. At the same time, ProtoTrust uses a highly general library of TM methods: the uTrust library. The uTrust library is developed as a research project and it is open for experimentation and development by the research community of Trust Management.

We have designed new algorithms that detect Internet auction frauds. We have also proposed the decision making rules that help users to detect fraudulent sellers. We have evaluated our algorithms on real data from the largest Polish Internet auction provider (Allegro), and have shown that we can protect users from almost half of fraudulent auctions when ProtoTrust alerts users in only 7% of auctions. Our best algorithm performs 41% better than random classification. We have also proposed simple rules to induce implicit nonpositive feedbacks which can affect existing reputation systems. Algorithms and decision rules presented in this paper are simple and easy to understand even for inexperienced users. Separately our tools provide only some information about the seller and his auctions, but in conjunction they create a very powerful Trust Management tool.

In the future we are planning to adapt our ProtoTrust environment to work with other major auction services. This requires a reimplementation of the crawling module (WWW crawler) because of a different design of every auction website. As long as the crawling module is compatible with the *uTrust* Trust Management library, no further changes are required to ProtoTrust. We want also to evaluate and decrease the network and system load of ProtoTrust. In the current solution, every user has to carry out her own crawling, which is globally not efficient. We plan to distribute the crawling tasks between all active instances of ProtoTrust. Our present work is still focused on comment classification. Currently we are developing the subsystem in which we enable ortography correction for users' feedback. We are going to include Polish and English dictionary into ProtoTrust environment.

References

1. Rubin, S., Christodorescu, M., Ganapathy, V., Giffin, J.T., Kruger, L., Wang, H., Kidd, N.: An auctioning reputation system based on anomaly detection. In: CCS '05: Proceedings of the 12th ACM conference on Computer and communications security, New York, NY, USA, ACM (2005) 270–279
2. Chau, D.H., Faloutsos, C.: Fraud detection in electronic auction. In: In European Web Mining Forum at ECML/PKDD. (2005)
3. Gavish, B., Tucci, C.L.: Reducing internet auction fraud. Commun. ACM **51**(5) (2008) 89–97
4. Melnik, M.I., Alm, J.: Does a seller's ecommerce reputation matter? evidence from ebay auctions. Journal of Industrial Economics **50**(3) (September 2002) 337–49
5. Ye, Q., Li, Y., Kiang, M., Wu, W.: The impact of seller reputation on the performance of online sales: evidence from taobao buy-it-now (bin) data. SIGMIS Database **40**(1) (2009) 12–19
6. Kaszuba, T., Rzadca, K., Wierzbicki, A., Wierzowiecki, G.: A blueprint for universal trust management services. Technical report, Polish-Japanese Institute of Information Technology, Warsaw, Poland (2008)
7. Gregg, D.G., Scott, J.E.: A typology of complaints about ebay sellers. Commun. ACM **51**(4) (2008) 69–74
8. Resnick, P., Zeckhauser, R.: Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In Baye, M.R., ed.: The Economics of the Internet and E-Commerce. Volume 11 of Advances in Applied Microeconomics. Elsevier Science (2002) 127–157
9. Dellarocas, C.: Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In: EC '00: Proceedings of the 2nd ACM conference on Electronic commerce, New York, NY, USA, ACM (2000) 150–157
10. Pandit, S., Chau, D.H., Wang, S., Faloutsos, C.: Netprobe: a fast and scalable system for fraud detection in online auction networks. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, New York, NY, USA, ACM (2007) 201–210

11. Morzy, M., Wierzbicki, A.: The sound of silence: Mining implicit feedbacks to compute reputation. In Spirakis, P.G., Mavronicolas, M., Kontogiannis, S.C., eds.: WINE. Volume 4286 of Lecture Notes in Computer Science., Springer (2006) 365–376
12. Grtler, O., Grund, C.: The effect of reputation on selling prices in auctions. Discussion Papers 114, SFB/TR 15 Governance and the Efficiency of Economic Systems, Free University of Berlin, Humboldt University of Berlin, University of Bonn, University of Mannheim, University of Munich (May 2006)
13. Schlgel, C., Wolff, B.: Country-specific effects of reputation and information: A comparison of online auctions in germany, the uk, and the us. FEMM Working Papers 07027, Otto-von-Guericke University Magdeburg, Faculty of Economics and Management (December 2007)