Predicting Successful Malware Products in the Underground Forum Markets

Bonajo, M.

Award date:
2020

Predicting Successful Malware Products in the Underground Forum Markets

Martijn Bonajo
Department of Mathematics and Computer Science

Eindhoven University of Technology
Eindhoven, The Netherlands
m.bonajo@student.tue.nl

***Abstract*—There are thousands of threads posted to undergrounds markets every month, however not all of these threads are selling a product and not all of the threads selling a product actually become successful. Understanding what products are going to be successful on underground markets can help security researcher focus only on the important products posted to these forums. For this purpose this research aims to identify not only the threads that are selling products, but tries to predict which of these selling threads will become successful. We use natural language processing techniques combined with social network analysis techniques to extract features from a Russian underground market. Using logistic regression we are able to explain 24% of the variance for successful products with evidence collected from the first week of trading, and up to 33% after three weeks.**

***Index Terms*—Security, Cybercrime, Natural language processing, Social network analysis, Underground market**

I. Introduction

Over the last couple of years there has been a steady increase in use of malware, ransomware and other attack on systems from home computers to enterprise systems [1]. In the past the only hackers were tech savvy individuals that liked the challenge of hacking or compromising systems. However nowadays a lot of the attacks are performed by less tech savvy individuals. These individuals can buy hacks, malware, ransomware, etc., from now we refer to this as products, from underground markets. In these underground markets the tech savvy individuals sell their products, such that it can be used by the bigger group of less tech savvy people for attacks. Especially since the advent of Exploit-as-a-service and Crimeware-as-a-service it has become easier for laymen to get into the hacking/exploit world [2], [3]. Another part where these underground markets play a role is in the supply chain of products. Users specialize in certain products and other users of these underground markets buy these specialized products and bundle them together as a separate product or offer these as Eaas/Caas. A big part of these markets originate from less developed countries, such as Russia and China [4], [5]. Popular attacks on these forums do see the light of day in real world attacks. [11] These attacks cause a lot of financial damages. It would be beneficial to know what kind attacks are popular on these markets, such that security professionals can focus on these attacks, before they are being deployed in practice. However monitoring the underground forums is a labor intensive task, which is made even more difficult by the potential language barrier of security professionals.

Scope of this work: The goal of this paper is to create a model to help predict successful products as to help security researchers focus on the high priority products. We do this by analyzing a prominent Russian underground market, extract features from data obtained from this market and build a model to predict the successful products.

II. Background

Over the years there has been a lot of research into cybercrime vulnerabilities and the underground markets that facilitate the trading and development of cybercrime attacks. Initial work in this area looked at the the life cycle of vulnerabilities [13]. In this paper the authors they analyzed security advisories and presented the discovery, disclosure, exploitation and patch date of vulnerabilities. They quantify the gap between exploit and patch availability and already show that so-called 'black hat' attackers create exploits faster then the software vendors create patches for known vulnerabilities. Thus showing us there is a timeframe where attackers can exploit these known vulnerabilities. Frei et al. [14] later showed that the exploit time still exceeds the patch time for vulnerabilities. Thus showing that exploiters still have sufficient time to use their exploits. Furthermore they develop a metric for the success of the responsible disclosure process, showing that the commercial vulnerabilities markets cannot be neglected in the prevention of exploits.

A second stream of research looked into attack resources, particularly as provided by underground markets where vulnerabilities and the relative exploits are traded. Florencio et al. [18] in 2010 started looking into the economic side of these markets. The authors take a look at Internet Relay Chat (IRC) markets. This is what they call a classic example of a market for lemons [29]. They show that these markets are thrive for rippers. Thus showing us that there is a difference in the successfulness of markets based on the type of market. A first quantitative assessment of the risks coming from underground markets is provided in [20]. They show that black market vulnerabilities are a big source of risk for end users. Furthermore they show that by monitoring underground

market activity they can produce better vulnerability mitigation strategies than traditional strategies based on vulnerability severity scores. Thus showing us that there is a real risk to end user from exploits that originate from underground markets. Building on this, [11] provides an empirical investigation into the economics of vulnerability exploitation and the effect of markets factors on the likelihood of exploitation. In this paper he reveals that exploits are priced similarly or above legitimate vulnerability markets. Furthermore he finds strong correlation between underground market activity and exploit likelihood. Finally he showed that the analyzed markets show signs of growth and the Exploit-as-a-Service may allow for cheaper exploit costs.

Aspects related to the success of a cybercrime market have also been considered in the literature. As already shown by Florencio et al. [18] some markets are thrive for rippers and scammers. So the question arises what is the difference between a successful underground market and a unsuccessful/failed one. This is the goal of the research by Allodi et al. [22]. In this research they take a look at a market regulatory mechanisms of a failed underground market and compare this to a thriving underground market. They show that the market structure and design evolved toward a market design that is similar to legitimate, thriving, online forum markets. Thus showing us that not only the type of market is important for the successfulness of the market as shown by Florencio et al. [18], but also how the markets regulatory mechanisms. This means that there are interesting markets that sell legit exploits and there are markets that are filled with scammers, showing us that we should find a market that is "legit" as to not get only scammer data.

On this same line, Motoyama et al. [7] started research into the social dynamics that play a role in the underground markets. The markets they studied are forum based, unlike the IRC based markets studied by Florencio et al. [18] They examined the properties of the social network, contents of products being traded and how individuals gain and lose trust. Giving us our first insight into how social relationships play a role in the trading on underground forums. However for their dataset they used data gained from complete SQL dumps, which includes the private messages between users. Access to users private messages is not normally possible for security researchers analyzing forums in real time, therefore their statistical approach is not something we can use in our methodology. Pastrana et al. [26] looked at underground forum data and tried to predict who would become key actors. The goal of their research was to predict key actors such that law enforcement can intervene at early stages. They build their model using logistic regression with k-means clustering and social network analysis. Trying to predict who would become key actors on the forum has many similarities with trying to predict successful products. However there is one oversight in this paper that we try to avoid. They use social network analysis on a graph that spans all the data from beginning to the end of their dataset to retrieve features and then use these features to train their model.

III. Research Question and Hypotheses

Overall, the extant literature shows how underground markets evolved from IRC based to forum based, how their regulatory mechanisms and structure changed to mimic that of legit market places and how exploits traded on these markets do see actual exploitation in the wild. Thus showing is that these markets are a legit threat for end-users. Furthermore it showed us that many researchers using only vulnerability severity data is not a good predictor of whether those (potential) attacks represent a real threat. By contrast, the activity of underground markets appear to be of greater and greater importance in 'forecasting' which attack comes next. However, whereas it is clear that not all attacks are equally successful in the underground markets, it is still unclear whether it is possible to *predict* which be by monitoring a market's activity.

Further extending this line of work, in this paper we investigate the following research question:

Research question: How can we use social features characterizing interactions in the underground markets to predict the successfulness of a traded product?

In the following, we define *author* as being the user in an underground market that posts a new *product* (i.e. an attack technology) through a new *thread* on the market board. The first post on the thread where the *author* gives information about the *product* is the *thread post*. A *commenter* is an user that replies or comments to a product.

A. Hypotheses formulation

To guide the definition of our methodology, we here formulate the following hypotheses from the extant literature.

As already shown by Pastrana et al. [26] general forum statistics, such as number of posts and threads in each category, are useful in predicting who would become key actors. Therefore it is probable that these general forum statistics also help predict which authors post successful products. This leads to the first hypothesis.

Hypothesis 1: Products posted by authors with greater number of posts, and who have been on the forum for longer, are more likely to be successful.

If this hypothesis is true, author characteristics may be strong indicators of product successfulness. If this hypothesis is not correct this means that it doesn't matter how active an author has been on the forum or how long the author has been on the forum.

It would make sense that when an author that has been more active, posts a new product, that there is a higher change of this new product also being successful. This would be even more probable if the author has several of these successful products in the same category as this new product, as the author can than be regarded as an expert in that field. This leads to the second hypothesis.

Hypothesis 2: How successful previous products sold by the same authors have been, is an important indicator of the successfulness of the newly posted product.

If this hypothesis is true than that would mean that successful products are posted by authors that have already been successful. If this hypothesis is not true, than that means that previous success is not indicative of new success.

In our previous hypothesis we assume that the author of a thread can only be an expert in certain categories if the author has already posted successful products in that category. However it can also be that the author of a thread has shown interest in certain categories by making posts on threads in that category, without having posted a product in that category (yet). This leads to the third hypothesis

Hypothesis 3: Successful products are more likely to be posted by authors that have shown interest in the same product category during previous activity in the forum.

If this hypothesis is true, that would mean that an author that has shown interest in the same category of products as the one currently traded, has a higher likelihood of posting a successful product in that category. If this hypothesis is not correct than it would not matter if the author of the thread has shown interest in the category of the product before.

As the only way to know how a thread is received is by the comments on that thread it would make sense that these comments play a big role on the probability of success. However not every commenter on a thread is equal, some commenters are more knowledgeable in certain topics or have better reputation than others. So a comment on a thread doesn't tell the whole story without knowing how important the commenter is. This leads to the fourth and final hypothesis.

Hypothesis 4: Products that receive comments made by important commenters are more likely to be successful than products that do not.

Comments expressed by commenters that are often quoted and appear often in other threads on the forum may boost the likelihood of successfulness of a product.

IV. Methodology

In this section we discuss the methodology we developed to test our hypotheses. A global overview of the methodology can be seen in figure 1. As all malware underground markets are operated through forums [37], we build the methodology considering the employment of forum-based data. Specific information on how each methodological step is implemented (e.g. ML model choices) is provided in Section V.

A. Data sanitization

Due to the nature of criminal activities, any analysis (independently of the data source) relies on unstructured data. Sanitization practices may depend on the source of the data; as markets for cyber-attacks are mostly forum-based [7], [11], removing markup tags and other tags that might be in the data (such as HTML, markdown, BBCode, etc.) typical of forum data may be required.

A more subtle but important aspect of the data sanitization is language specific, which may require employing procedures to identify the language of the data. Note that multiple languages could be used in a given forum. Strategies include removing sentences that are in a different language than the majority of the thread post/comment (e.g. if a thread post advertises in both English and Russian, remove the text that is not the majority). Additionally, one can consider stop words to be noise and thus remove those as well. Additional standard techniques such as lemmatization should be employed.

B. Data preparation

1) Data classification: We employ standard practices for data classification. Two main classification tasks are relevant in this work, namely: 1) identifying threads presenting products that are advertised for sale; and 2) identifying the positivity/not positivity of a comment on a product (*sentiment analysis*). For both these tasks we employ manual labeling on a random sample of selected observations (forum threads and comments respectively), and train machine learning models to extend the classification to the unlabeled data. Details on which models are employed for the implementation of each task are given in Sec. V. Regardless of the model selection, to know how well our models perform we report the conventional information retrieval statistics, precision, recall and F1-score. To be able to calculate these values we first need counts of all true positives (TP), false positives (FP) and false negatives (FN). We get these by comparing the model prediction with the ground truth. We can now calculate the precision, recall and F1-score as defined below:

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$
$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

To evaluate the generalizability of our classification results we employ cross validation. The type of cross validation we use is k-fold cross validation where we use two folds and three repeats. This means that for a given dataset we generate through random selection two equal size datasets d_0 and d_1. d_0 and d_1 are both employed iteratively as the training and testing sets. The sampling, training, and testing process is then repeated three times.

2) Threads.: From threads we need two more relevant dimensions, namely the thread type and the thread category. The thread type is a label that is either *selling* or *other*. Similarly, the thread category is used to categorize threads selling products of the same type together. Categories can generally be defined by the forum sections related to the types of products traded in the market.

3) Comments.: To understand the impact of comments on the overall discussion on the traded product, we evaluate: a) the addressee of the comment (another commenter or the thread author); and b) the sentiment of the comment (positive/not positive). The sentiment of a comment is used in Section IV-B5 to see how positive and negative comments relate to each other in the forum discussion.

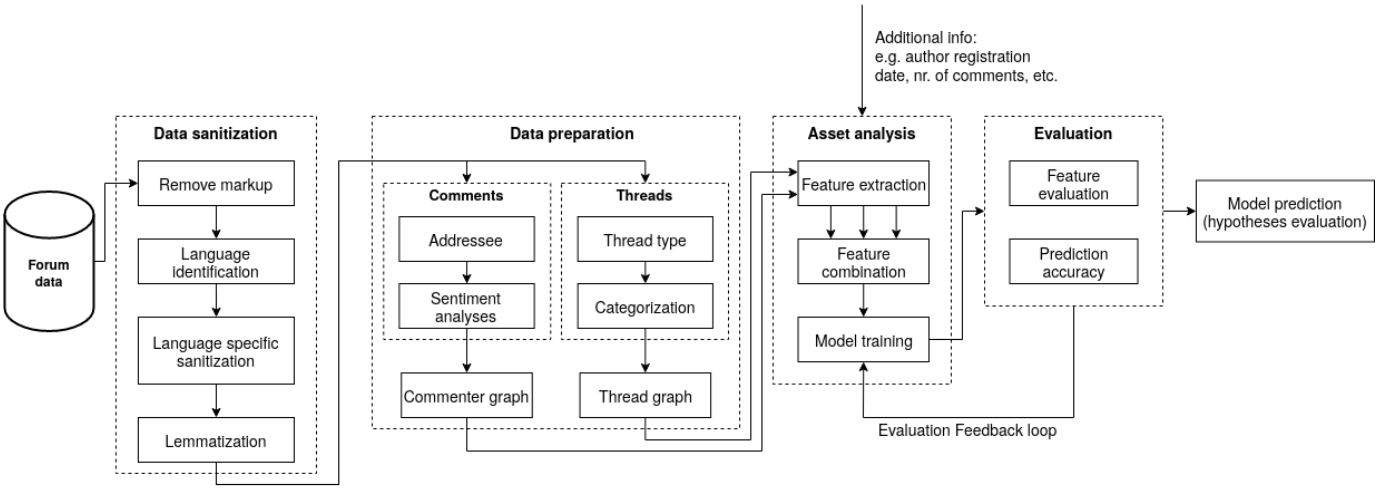

Fig. 1. Graphical overview of all steps in the methodology

4) Establishing the ground truth on product successfulness: We define as successful a product for sale that received a number of positive comments that exceeds the number of negative comments on the product thread.

5) Social network analysis: In this subsection we discuss what graphs we build, how we build them and how the graphs relate to the methodology.

a) Thread graph: First we define a **thread graph** as follows: A *thread* graph $G_t = (V_t, E_t)$ is a directed graph where each node in the set V_t is a thread. There are two types of edges: the first edge indicates that the author of a thread V_t^i is also the author of another thread V_t^j ($V_t^i \leftrightarrow V_t^j$). The second type of edge indicates that the author of a thread V_t^i comments on a thread by a different author V_t^j ($V_t^i \rightarrow V_t^j$). We label edges of the second type with the sentiment (positive/not positive) of the comment. Threads (i.e., nodes in the graph) are also labeled with their positive to not positive ratio. An illustrative example of a *thread* graph is reported in fig. 2.

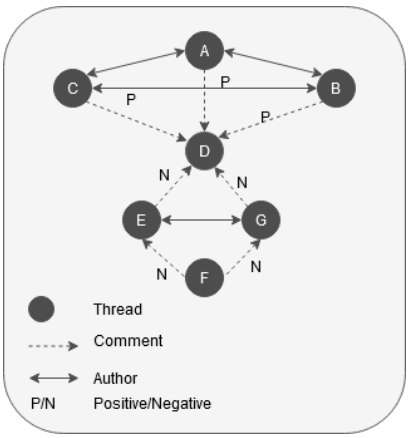

Fig. 2. Example of what a thread graph could look like

The goal of this graph is to extract relevant information on the relations between authors of different threads. By looking at the author edges we can see what threads the author has made and in what categories the author is mainly active. By looking at the commenter edges we can see the interest of the author. Furthermore the comment edges gives us insight into how different authors influence each other by taking a look at each others threads. By looking at the nodes we can see how well the threads have been received by the community.

b) Commenter graph: Secondly we define a **commenter graph** as follows: A *commenter* graph $G_c = (V_c, V_a, E_t)$ is an directed graph where each node in the set V_c is a commenter on a forum thread, and V_a is the author of a thread. There are two commenter-commenter edges. The first edge indicates that a commenter directly quotes another commenter in a thread (e.g. using the "quote" functionality of a forum); this will be represented by an edge $V_c^i \rightarrow V_c^j$. The second edge is if the commenter doesn't directly quote someone, and is represented as ($V_c^i \rightarrow V_a^j$). Similarly to the thread graph, edges in the commenter graph are label by their sentiment and whether or not the edge corresponds to a quote, or a direct reply to the main thread. An example of this *commenter* graph can be seen in figure 3.

The goal of this graph is to extract relevant information on the relations between commenters. In the thread graph we already have edges based on comments, however these edges only exists for authors and we therefore miss comments made by non-authors, which may encode previous relations or 'importance' of specific commenters in the community. Measures typical of social network analysis (e.g. centrality, betweenness, ..) can be employed to evaluate how important or central a commenter is to a community.

C. Feature selection

Now we have everything we need to start selecting the features that we think collaborate with our hypotheses.

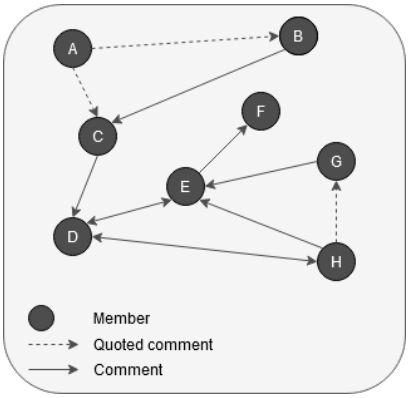

Fig. 3. Example of what a commenter graph can look like

1) Hypothesis 1: For hypothesis one we take features that can be extracted from general forum data. These features include: age of the account, total number of threads posted per category and total number of comments posted per category.

2) Hypothesis 2: For hypothesis two we want to know how successful previous products posted by the author were. For this we take the number of successful and unsuccessful products posted per category. We also compute a success score per member for a given category as follows:

$$SS(M,C) = \sum_{t \in T}(t_{success})$$

Where M is the member, C is the category, where T is all selling threads made by M in category C and $t_{success}$ is whether the thread was successful (1) or unsuccessful(-1). This shows us if the author has posted more successful than unsuccessful products per category or not.

Additional features describing the prominence of a thread are taken from standard social network analysis practice, and are calculated from the thread graph: in-degree[1], out-degree[1], eigenvector[2], betweenness[3], hubs[4] and authorities[4]. We consider a thread to be prominent if the distribution of one or more its features is in the 80th or higher percentile of the overall distribution. We then calculate the thread success score as a simple sum of the important threads.

$$TSS(A) = \sum_{t \in T}(t_{important})$$

Where A is the author of the thread, T is all previous threads by the author and $t_{important}$ is whether the thread is important (1) or not (0) as identified by the SNA metrics.

3) Hypothesis 3: For hypothesis three we have to select features that represent the interest of the author of the product. For this we look at the number of threads and the number of comments posted per category. We calculate an interest score for a member in a certain category, for this we use the same formulation as Pastrana et al. [9] and we define it as follows:

$$IS(M,C) = 3 * N_t(M,C) + N_c(M,C)$$

Where M is the member, C is the category and $N_{\{t,c\}}$ denotes the total number of threads/comments in category C by member M.

4) Hypothesis 4: For hypothesis four we select features from the commenter graph that we think shows how important the commenter is. For this we use standard SNA features again, such as the in-degree, out-degree, eigenvector, betweenness, hubs and authorities. To determine commenter prominence we employ the same criteria used for the thread analysis in hypothesis 3. We then compute the commenter score as the sum of all the important commenters on a thread.

$$CS(T) = \sum_{c \in C}(c_{important})$$

Where T is the thread, C is all the comments on that thread and $c_{important}$ is whether the commenter is important (1) or not (0) as identified by the SNA metrics.

5) Model prediction: As discussed before we want to apply the principles as discussed by Pendlebury et al. [10]. What this means is that for all the features that we select, including those retrieved from the SNA part, that we cannot use data from the future. Thus we cannot build a thread/commenter graph out of the entire dataset, but we have to build a new one that corresponds to the state of the forum that corresponds to the time when the thread was posted. This allows us to evaluate our prediction model *realistically* by considering only evidence that would be effectively available at the time of the prediction (as opposed to trying to predict product successfulness using evidence available at a time when the product *already* became, in fact, successful).

To evaluate our hypotheses we employ a set of logistic regression models. Logistic regression is used to explain the relation between the dependent variable, in our case successful or not and one or more independent variables, in our case the hypotheses. As our dependent variable is a binary variable we will be using a set of binomial logistic regression models of the form:

$$Y = \beta_0 + \beta_1 X_1^{t_n} + \beta_2 X_2^{t_n} + ... + \beta_n X_n^{t_n},$$

where the dependent variable is Y and every independent variable we have is $X_i^{t_n}$, as measured at instant in time t_n with $n \in \{0,1,2,3\}$ weeks since the time of posting of the thread. First we build a logistic regression per hypothesis to verify how well just that one hypothesis is at explaining the independent variable. We make sure to check that the independent variables are indeed independent, thus not highly correlated. After we have done this for every hypothesis we will then combine all the independent variables of all the hypotheses into one final model. Finally we will test how well our final model is at predicting the dependent variable.

[1]https://en.wikipedia.org/wiki/Directed_graph#Indegree_and_outdegree
[2]https://en.wikipedia.org/wiki/Eigenvector_centrality
[3]https://en.wikipedia.org/wiki/Betweenness_centrality
[4]https://en.wikipedia.org/wiki/HITS_algorithm

V. Data selection and preparation

To test our hypotheses we rely on a dataset gathered from a Russian underground cybercrime market trading malware tools. The dataset is provided by a research partner (Mr. Martin Pozdena, Auxilium), hence the data collection is not part of our contribution. The dataset was built for commercial purposes for the analysis of underground malware operations. A brief description of the forum structure can be found in table X. We refer to the market as RuMarket[5].

A. Building the ground truth

To build the ground truth we manually labeled threads as either being successful or unsuccessful, based on the thread type and the comments on the thread. For threads in Russian we used DeepL[6] to first translate the the thread and comments to English, as we do not have native understanding of the Russian language.

B. Data sanitization

1) Removing markup: As the dataset was gathered by scraping the forum, all the HTML tags that are used for markup are still in the dataset. We use regular expressions and custom quote algorithm to remove the HTML markup tags.

A complication is that for some tags we want to add back boolean metadata, for example whether the text contains a link or code block. Which tags we replace and if we add back metadata for those tags can be seen in table I.

Tag	Remove text in between tags	Metadata
Link	yes	yes
Code	yes	yes
Emoticon	yes	yes
Quote	yes*	yes
Image	no	yes
Chat log	yes	no
Spoilers	yes	no
Others (e.g b, span)	no	no

TABLE I
Rules for removing HTML tags

* We remove the quotes from the text, however we still keep it linked to the comment, as we need it in the Addressee part.

2) Language identification: An informal analysis of the dataset reveals that two main languages are employed on our forum, Russian and English. However there are comments that are Russian, but they are written within the Latin alphabet (i.e., they are *transliterated*), we refer to these types of comments as having language `iso9`[7]. We could not find a library that also supports detecting `iso9`. As we still want to distinguish between all three languages we decided to build our own language identification model. For this task we used the `fastText` library, a sentence classification method, that is developed by Facebook [30]. We manually labeled 1859 thread posts/comments in total with the correct language. Where the majority of labeled text came from thread posts. However after testing the model on comments we found that the model did not perform as expected and therefore we later manually labeled another 2543 comments with the correct language. First we take a baseline score to see the performance of our model. We calculate the baseline using the popular language identification library for python called `langid.py` [31]. The scores can be found in table II. As can be seen `langid.py` has a low accuracy for English as it classifies almost all `iso9` samples as English. The `fastText` model performs well for both English and Russian, however the `iso9` is lagging a bit behind in terms of precision.

Model	English			Russian			ISO9		
	P	R	F1	P	R	F1	P	R	F1
langid.py	**1.0**	0.52	0.68	**1.0**	**0.99**	**1.0**	-	-	-
fastText	0.98	**0.93**	**0.95**	0.99	0.91	0.95	**0.86**	**0.98**	**0.92**

TABLE II
Precision, recall and F1-scores for language classification

3) Language-specific sanitization: After language identification we can also remove stop words for that language. For this task we use the stopwords package from `NLTK` [32].

Sanitization of forum messages written in `iso9` is less straightforward as there is no stopwords package for `iso9`. To still be able to use thread posts/comments that are in `iso9`, we decided to transliterate it back to the Cyrillic alphabet. Doing this has risks that the transliteration is not correct, as many users do not follow the transliteration guidelines as specified in the `iso9` standard. To make sure that the impact of this transliteration isn't too great we test all our classification that are dependent on language with transliteration and without.

4) Lemmatization: For lemmatization we use a NLP python library called `spaCy` [33]. This library provides pipelines for lemmatization and an English model that can be used to lemmatize English text. However it does not provide a Russian model, for this we retrieve a Russian model from Stanza [34]. Stanza is a NLP library made by Stanford's NLP group and provides models for a lot of languages. To use the Stanza model in `spaCy` we can use an additional library from `spaCy`, named *spacy-stanza* that wraps the Stanza model so it can be used in the `spaCy` pipeline, as used for the English lemmatization.

C. Data preparation

In this subsection we have a look at the data preparation phase of the methodology.

1) Thread type: For the thread type we labeled 535 English threads and 674 Russian threads. For English we identified 119 selling threads and for Russian we identified 244 selling threads. After labeling the ground truth we have a reasonable understanding of the words that are used for selling threads and for other threads (e.g. buying, question, flame). Therefore we started with a baseline model using simple regular expressions (regexp) to try and predict the thread type. We further tested the following classifiers: fastText, logistic regression,

[5]We do not disclose the real name of the underground market
[6]https://www.deepl.com/translator
[7]https://en.wikipedia.org/wiki/ISO_9

Language	English						Russian					
Model	**Sell**			**Other**			**Sell**			**Other**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Regex	0.56	0.56	0.55	0.86	0.86	0.86	0.54	0.91	0.68	**0.97**	0.78	0.87
SVM	0.41	0.72	0.52	0.95	0.84	**0.89**	0.84	0.83	**0.84**	0.90	0.90	0.90
SVM meta	0.43	0.62	0.50	0.92	0.84	0.87	0.76	0.83	0.79	0.91	0.87	0.89
SGD	**0.74**	0.51	**0.60**	0.77	**0.91**	0.83	**0.91**	0.75	0.82	0.82	**0.94**	0.87
SGD meta	0.51	0.59	0.54	0.89	0.85	0.87	0.81	0.79	0.80	0.87	0.89	0.88
Logistic regression	0.47	0.72	0.56	0.94	0.85	**0.89**	0.83	0.83	0.83	0.90	0.91	0.90
Logistic regression meta	0.45	0.50	0.47	0.86	0.83	0.84	0.63	0.74	0.68	0.87	0.80	0.83
XGBoost	0.44	0.73	0.54	0.95	0.84	**0.89**	0.66	**0.92**	0.77	**0.97**	0.83	0.89
XGBoost meta	0.45	0.70	0.54	0.94	0.84	**0.89**	0.66	**0.92**	0.77	0.96	0.83	0.89
fastText	0.26	**0.85**	0.38	**0.99**	0.81	**0.89**	0.75	**0.92**	0.82	0.96	0.87	**0.91**

TABLE III
PRECISION, RECALL AND F1-SCORES FOR THREAD TYPE CLASSIFICATION OF ENGLISH AND RUSSIAN (INCLUDING TRANSLITERATED `iso9`) THREAD POSTS

support vector machine(SVM), stochastic gradient descent and XGBoost. We choose these classification algorithms so that we have a wide range of different classifiers. XGBoost is a scalable end-to-end tree boosting system by Chen et al [35].

We run all our classification algorithms twice, once with the metadata and once without the metadata. As can be seen in table III for the English thread posts the regex classifier is a decent baseline, it even outperforms multiple classifiers.[8] As the most important class for us is the *sell* class we choose the logistic regression model as the overall best performer.

As can be seen in table III the scores are higher for all the classification methods for Russian language. Except for XGBoost all classifiers (excluding the meta versions) have an f1 score of 0.80 or higher. Here we choose the SVM model as our classifier for Russian text. In table XI we have the thread type classification for Russian without `iso9`, as can be seen there is no significant difference.

2) Categorization: For the categorization of threads we use the (sub)board the thread was posted in. This is based on the approach by Pastrana et al. where they use the boards to map interests to categories. The mapping between a (sub)board and a category can be found in table XIII.

3) Addressee: During data preparation, we found mismatches in the usernames reported in conversations where a commenter's comment was 'quoted' using the *quote* code of the forum.[9] To address this problem we used the algorithm as described in algorithm 1. The gist of it is as follows: by default the addressee is set to the thread author, if the comment contains a quote, we retrieve the username of the most outer quote (quotes can be nested), we then check if the username exists in our dataset, if so then that username becomes the addressee. If that is not the case we retrieve the quoted text and check the other comments on the thread for the quoted text. If there is a comment with the quoted text, then we set the addressee to the author of that comment and otherwise we keep the addressee as the thread author as we cannot find the quoted user.

Algorithm 1 Addressee algorithm

```
procedure ADDADDRESSEE(thread, comment)
    addressee ← thread.author
    if comment.containsQuote then
        username ← getUsername(comment.quote)
        exists ← usernameExistsInDB(username)
        if exists then
            addressee ← username
        else
            text ← getQuotedText(comment.quote)
            for other in thread.comments do
                if other ≠ comment then
                    if text == other.text then
                        addressee ← other.author
    return addressee
```

4) Sentiment analysis: As we need to know the sentiment to know how a selling thread is received, we manually labeled all the comments on threads that were classified as being selling during the *thread type* section. We manually labeled 1808 comments with their sentiment. The process was that one person labeled all the comments, afterwards a random subgroup was selected and together with a domain expert it was checked that there was agreement on the sentiment as labeled. This does bring with it some limitation that will be discussed in section VII-A. As can be seen in table V the majority of the comments made on selling threads is of the *other* class (questions, general remarks, etc.). This is followed by the *positive* class which is significantly smaller already. The *negative* class is by far the smallest class.

Now that we have the ground truth labeled we use the same approach as was done for the thread type classification. As can be seen in table IV the classification for the English language lags behind the classification for the Russian language as was the case for thread type. For both languages the classification

[8]We do note however that the classification models perform sub par from what we expected. There could be multiple reasons why this is, but we think that the reason is because English is not the mother tongue for the vast majority of the forum members and therefore the English text is not of the same quality as the Russian text.

[9]This might be because the forum uses different encoding between the username field and the text or this might be because the web scraper parsed the username differently.

Language	English									Russian								
Model	**Positive**			**Negative**			**Other**			**Positive**			**Negative**			**Other**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
SVM	0.63	**0.79**	**0.70**	0.22	0.49	0.29	0.88	0.71	0.78	0.79	0.84	**0.81**	0.32	0.56	0.40	0.88	0.77	**0.82**
SVM*	0.67	0.74	**0.70**	0.22	0.42	0.28	0.85	0.71	0.77	0.81	0.80	0.80	0.26	0.51	0.34	0.86	0.78	**0.82**
SGD	**0.69**	0.70	0.69	**0.41**	0.40	**0.40**	**0.75**	**0.75**	0.75	0.80	0.79	0.80	0.32	0.53	0.40	0.84	0.78	0.81
SGD*	0.66	0.74	**0.70**	0.22	0.37	0.27	0.83	0.72	0.77	0.81	0.79	0.80	0.31	0.44	0.36	0.82	0.78	0.80
LogReg	0.65	0.76	**0.70**	0.30	0.50	0.37	0.85	0.73	0.78	0.80	0.83	**0.81**	**0.41**	0.49	**0.44**	0.84	**0.79**	0.81
LogReg*	0.68	0.69	0.69	0.23	0.33	0.26	0.78	0.72	0.75	**0.83**	0.76	0.79	0.33	0.44	0.37	0.79	**0.79**	0.79
XGBoost	0.52	0.69	0.59	0.12	0.51	0.19	0.89	0.67	0.77	0.59	0.77	0.67	0.15	0.55	0.23	0.91	0.70	0.79
XGBoost*	0.52	0.71	0.60	0.12	0.45	0.18	0.89	0.67	0.77	0.60	0.78	0.68	0.15	0.57	0.24	0.91	0.70	0.79
fastText	0.59	**0.79**	0.67	0.10	**0.82**	0.18	**0.93**	0.68	**0.79**	0.74	**0.86**	0.79	0.09	**0.69**	0.16	**0.94**	0.73	**0.82**

TABLE IV
PRECISION, RECALL AND F1-SCORES FOR SENTIMENT CLASSIFICATION OF ENGLISH AND RUSSIAN (INCLUDING TRANSLITERATED `iso9`) COMMENTS. MODEL* IS THE CLASSIFICATION WITH METADATA

	Positive	**Negative**	**Other**	**Total**
English	142	59	242	443
Russian	427	170	724	1321
iso9	8	2	34	44
Total	577	231	1000	1808

TABLE V
SENTIMENT LABELS IDENTIFIED ON THE SELLING THREADS

of *negative* comments is much lower then for the other two classes. This could be because the *negative* class is the smallest class in our ground truth. It can however also be because some of the classified *negative* comments are quite subtle or lose some of their meaning after the preprocessing stage (stopwords removal and lemmatization).

As was the case with the classification of thread types, we again checked that adding the transliterated `iso9` comments to the Russian comments has any effect. As can be seen in table XII in the appendix the classification scoring doesn't change significantly.

D. Social network analysis

With the thread type and comment sentiment classified we can build the graphs needed for the social network analysis part. For this we need to build two separate graphs. For this task we use `graph-tool` library for python by Peixoto [36].

The goal of the graphs is to represent the social features as they were on the forum for a given date, e.g. the date a thread was posted, a week after, etc. This means that we have to build the graphs based on a given date, which we refer to as t_i.

For the thread graph this means we select all the threads that are classified as *selling* and are posted before this date as vertices. Between these threads we add edges if they have the same author, this is safe to do as all threads are from before t_i. Now we have to select all the comments made by authors that have a thread in the graph and where the post date of the comment is before t_i. We can then add outgoing edges from these threads to the commented on threads and label them with the comment sentiment. Because we only select comments that are made before date t_i, all vertices and edges in the graph are now from before t_i. A similar procedure is followed to build the commenter graph.

Having build the graphs we can now run the social network analysis methods. Each of these methods will return a vector of values for each of the vertices in the graph. We then compute for every SNA metric the top 20 percentile to check which of the vertices in the graphs are important.

E. Feature selection

Now that we have everything in place we can start to generate and combine the features that we want. For a given thread T we take the date that it was posted. We refer to this post date t_0, now we create three more dates for which we want to retrieve features by consecutively adding a week to t_0, up to three weeks after posting date (t_3). We adopt this method and perform separate computations at the four times for all hypotheses.

Finally the success label is added to the features, this is the only data point that is selected over the whole dataset, as this is what we are trying to predict.

VI. ANALYSIS

The dataset contains threads and comments posted by forum users from 2005 to the first month of 2019. The data spans 42587 threads, with 308908 comments made by 12588 members. The forums are divided into various different topics which are divided into subtopics. As can be seen in table XIV there are multiple topics on the forum, however the most popular ones are financial, malware and spam.

A member can post a thread to one of these subtopics; we refer to these members as an *author* of a thread. Other members (as well as the author) can then comment on this thread; we refer to these members as *commenters* of that thread. For an overview see figure 4.

A. Data overview

We first provide an overview of the data. First we take a look at the distribution of threads and comments. By plotting the number of threads and comments on a timescale, we find that from 2016 onwards there are more threads and more comments posted per year; this suggests that either the current members

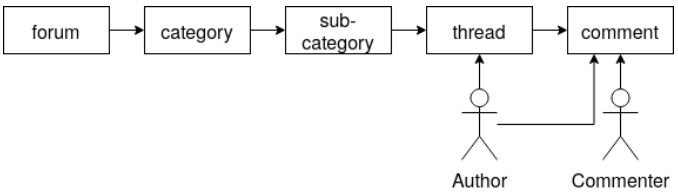

Fig. 4. High level overview of forum structure

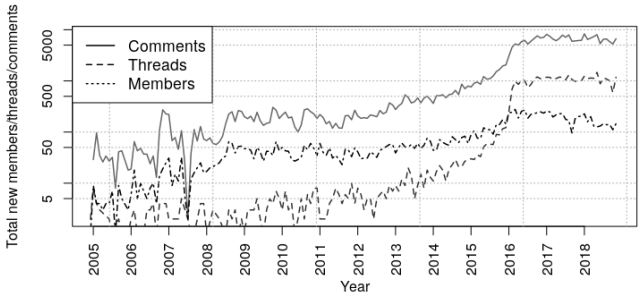

Fig. 5. Line-plot depicting the number of new member, comments and threads per month

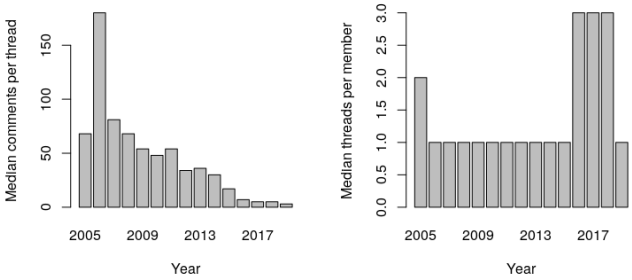

Fig. 6. Median number of comments per thread on the left and median number of threads per member on the right

	Thread posts			Comments			
	Selling	**Other**	**Total**	**Positive**	**Negative**	**Other**	**Total**
English	1%	9%	10%	3%	1%	9%	13%
Russian	23%	60%	83%	23%	6%	57%	86%
iso9	1%	6%	7%	<1%	<1%	1%	1%
Total	25%	75%	100%	26%	7%	67%	100%

TABLE VI
DISTRIBUTION OF LANGUAGE OF THREAD POSTS AND COMMENTS

started being way more active around that period, or that a lot of new members joined the forum. Looking at figure 5, we can clearly see that the rate at which new members joined steadily increased in 2014 and 2015, after which there was a small spike in 2016 and after this spike the number of new members stabilizes. Note that the join date of a member is based on either their first comment posted or their first thread posted, which ever is made first. This is because as discussed before we only have *public* data from the forum, so members that join and do not make any public appearance on the forum we cannot know about.

To further evaluate the commenting rate per thread (i.e. how on average commenters are engaged with a newly opened thread), we plot the median number of comments per thread per year. Looking at figure 6 we see that the median number of comments per thread decreases significantly over the years.

As can be seen in figure 5 around 2016 the number of new threads overtakes the number of new members. This would suggest that members are on average making more new threads. If we look at figure 6 we can see that the median number of threads per member does indeed go up from one thread per member from 2006 to 2015, to three threads per member in 2016 to 2018.

Looking in what categories the selling threads are posted, we see that over the years the most popular categories change and that the forum added new categories that were not available at the start, as can be seen in table XV. One of these categories is the auction subboard, in here members can post their products and other members can bid on them. As seen the auction has 0% of the threads in the years before 2016 and then quickly becomes one of the most popular categories, which might also help explain the growth in threads per member.

Having a look at the distribution of threads with their classified thread type and comments with their classified sentiment in different languages as seen in table VI, we find that the vast majority of content on the forum is in Russian. We see that even though 7% of threads are classified as being written in `iso9`, only 1% of comments are classified as `iso9`. Russian also seems to have higher percentage of selling threads relative to the total number of Russian threads, as well as relatively more positive comments in comparison to both English and `iso9`.

At last we take a look at the distribution of the features that we use as independent variables in the logistic regressions. We only plot the distribution for threads that are labeled as being successful.

As can be seen in figure 7 at t_0 we have that the majority of accounts are zero days old. This makes sense as the join date of a member is based on first thread or comment posted. What this means is that a significant amount of members first interaction on the forum is by posting a thread.

In figure 8 about the relevant threads, we see that the first bin start to shrink after t_0, this means that authors start to post more threads in the same section. The same can be seen for the relevant number of comments posted in figure 9.

Having a look at figure 10, we see that most authors have a negative relevant success. Meaning that most authors have posted majority unsuccessful threads in the relevant category when they post a new thread. The success scores seem to be relatively stable as there is not much change even after

	t_0	t_1	t_2	t_3
(Intercept)	-0.51^{***}	-1.01^{***}	-1.20^{***}	-1.31^{***}
	(0.04)	(0.07)	(0.09)	(0.10)
log(account age)	-0.12^{***}	-0.12^{***}	-0.11^{***}	-0.11^{***}
	(0.01)	(0.02)	(0.02)	(0.02)
log(relevant threads)	-0.72^{*}	0.72^{**}	0.81^{**}	0.91^{**}
	(0.36)	(0.28)	(0.28)	(0.28)
log(relevant comments)	0.11	0.21^{***}	0.20^{**}	0.22^{***}
	(0.06)	(0.06)	(0.06)	(0.06)
scale(relevant success)	0.01	0.03	0.04	0.04
	(0.02)	(0.02)	(0.02)	(0.03)
log(thread success score)	0.07^{*}	-0.05	-0.04	-0.05
	(0.04)	(0.04)	(0.04)	(0.04)
log(relevant interest)	0.47	-0.78^{*}	-0.86^{*}	-1.04^{**}
	(0.39)	(0.35)	(0.36)	(0.37)
commenter score > 0	13.40	2.07^{***}	2.32^{***}	2.47^{***}
	(161.26)	(0.06)	(0.06)	(0.06)
AIC	10414.23	8976.83	8514.14	8239.00
BIC	10470.94	9033.53	8570.85	8295.71
Log Likelihood	-5199.12	-4480.41	-4249.07	-4111.50
Deviance	10398.23	8960.83	8498.14	8223.00
Num. obs.	8854	8854	8854	8854
Pseudo R2	0.02	0.24	0.30	0.33

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Relevant success and thread success score are computed as in IV-C2, relevant interest computed as in IV-C3 and the commenter score as in IV-C4

TABLE VII
STATISTICAL MODELS OF THE FINAL HYPOTHESIS FOR ALL TIME POINTS

three weeks. Same can be said for the thread success score in figure 11 as they seem to be relatively stable. Also here goes that most authors do not yet have important threads before they post a new thread as can be seen by the first bin.

Looking at the interest score in figure 12 we see that after posting a new thread the interest score for that category seems to go up, indicating that the author keeps showing interest in that category.

Finally we look at the distribution of the commenter score in figure 13. We see that at t_0 almost every thread has a commenter score of zero which makes sense as the thread shouldn't have any (important) commenters yet. After one week the commenter score has increased significantly, after this it seems to be relatively stable.

B. Hypotheses evaluation

To test how well our hypotheses are at predicting the successful products, we took all 8854 selling threads as identified by our thread type classification, that were posted before 2019. Out of these selling threads, 2505 are identified as becoming successful. For every thread we retrieve the features for the four time points (when posted, a week later, two weeks later and three weeks later). This gives us a total dataset containing $8854 * 4 = 35416$ samples.

Running the logistic regressions for the four time points gives us four regressed models, whose coefficients reflect the hypotheses formulated in Section III-A. Table VII reports the joint evaluation of all hypotheses parameters across the four time points. A full breakdown is given in the appendix (Tab. XVI-XIX). A looking at table XVI-XIX in the Appendix

Hyp.	Variable	Supported	Coefficients			
			t_0	t_1	t_2	t_3
1	Account age	No	-0.12^{***}	-0.12^{***}	-0.11^{***}	-0.11^{***}
	Relevant threads	Yes	-0.72^{*}	0.72^{**}	0.81^{**}	0.91^{**}
	Relevant comments	Yes	0.11	0.21^{***}	0.20^{**}	0.22^{***}
2	Relevant success	No	0.01	0.03	0.04	0.04
	Thread success score	No	0.07^{*}	-0.05	-0.04	-0.05
3	Relevant interests	No	0.47	-0.78^{*}	-0.86^{*}	-1.04^{**}
4	Commenter score	Yes	13.40	2.07^{***}	2.32^{***}	2.47^{***}

TABLE VIII
OVERVIEW OF HYPOTHESES, THEIR INDEPENDENT VARIABLES AND IF THE LOGISTIC REGRESSION SUPPORTS THE HYPOTHESIS

shows that regressed coefficients for each variable remain largely stable across models irrespective of the time point of the analysis, suggesting that the identified effects are stable.

Next we are going to look at the reported pseudo R2 scores. The pseudo R2 score to tell us how well our model is at explaining the dependent variable. The pseudo R2 score tells how much of the variance the model can predict, in other words how much of the variance between successful and not successful the evidence that we are providing (the dataset), the model can explain.

If we now take a look at table VII, where coefficients of the complete model for all time points is reported, we see that our complete model goes from a pseudo R2 score of 0.02 at t_0, to a pseudo R2 score of 0.24 at t_1. Meaning that when a thread gets posted we are unable to explain the variance in the dependent variable, but one week after the thread is posted we are already able to explain 24% of the dependent variable. We see the pseudo R2 score increase even further for time points t_1 and t_2.

Having a look at the pseudo R2 score of the commenter score in tables XVI-XIX we see that we have a score of 0.00, 0.22, 0.29 and 0.32 for time points t_0, t_1, t_2 and t_3 respectively. This means that except for at time point t_0, the vast majority of the variance gets explained by the commenter score, suggesting that the presence of a prominent commenter in the forum (positively) affects the likelihood of successfulness of the product.

In table VIII we have reported our hypotheses, the independent variables for that hypothesis, the coefficients reported for that hypothesis and whether or not our hypotheses are supported by the logistic regression. For all our hypotheses we have the assumption that features identified by an hypothesis positively effect the successful outcome, hence we consider an hypothesis supported if the coefficient associated to that feature is significant *and* positive. Coefficients can be interpreted as indicating the relative increment in odds of success for when the feature is present, or increased by one unit. For example, the coefficient for `commenter score > 0` in model at t_3 is 2.47 (positive and significant). This indicates that when an

Time point	Unsuccessful			Successful		
	P	R	F1	P	R	F1
t_0	0.72	0.99	0.83	0.00	0.60	0.00
t_1	0.83	0.88	0.85	0.63	0.53	0.58
t_2	0.85	0.86	0.85	0.64	0.63	0.63
t_3	0.87	0.86	0.86	0.64	0.66	0.65

TABLE IX
PRECISION, RECALL AND F-SCORE FOR THE LOGISTIC REGRESSION WITH SUCCESS WHEN P >0.5

important commenter comments on a product, the chances of the product being successful are 2.47 times higher than for products where an important commenter does *not* comment. Similarly, the coefficient -0.11 for `log(account age)` indicates that for a unit increase in the log of account age, there is an 11% decrease in the odds of the product being successful. As we take the natural logarithm of account age, this means that, roughly, a 1% increase in account age corresponds to an 11% decrease in the odds of success.

For hypothesis one we have that the account age is negative and significant and opposite of the direction that is initially hypothesized and therefore is not supported. The relevant threads start out as being significant negative at t_0, but become significant and positive for the other time points and is therefore supported. The relevant comments is significant and positive and is therefore supported. For hypothesis two, we have that relevant success and thread success score are not significant and therefore not supported, but thread success score is significantly negative and therefore not supported. For hypothesis three the relevant interest is significant and negative and therefore not supported. And finally for hypothesis four the commenter score after t_0 is significantly positive and therefore supported.

Finally we take a look at using the logistic regression as a predictor. For this we take as predictor that when the logistic regression gives a probability of greater than 0.5 the thread becomes successful we get the precision, recall and f-scores as seen in table IX. We clearly see that the prediction of successful products is not possible when the thread is posted, however it goes up significantly after one week. This means that after one week we are able to predict of the selling threads which threads will become successful with an accuracy of about 50%. However as the logistic regression gives a probability of the likelihood of success, it can also be used as a mechanism to prioritize threads.

This shows us that using the important comments on a selling thread gives us the most information about the successfulness of that thread. When studying the dataset we often found threads that were posted by non-important authors. The way these products would gain trust is by an established user commenting on the thread. This is now also reflected in our findings.

VII. DISCUSSION

We used logistic regression to test how well all of our hypotheses are able to explain the successfulness of a thread separately and finally we used the combination of all hypothesis together to test how well our complete model is able to explain the successfulness. To test how well the models perform over time we have four different time points for which we selected all the features from. We saw that at the moment the thread was posted the model was not able to explain any of the variance. However one week after the thread was posted, we already get a lot of information. After two weeks we get even more information however this growth is less than between t_0 and t_1. After three weeks we still get more information, however the growth rate seems to significantly slow down, therefore it seems to be a logarithmic growth. It turns out that studying the commenters is the best predictor by far for the successfulness of a product.

One question remains, namely how could this research be used in the field. If we have a forum that is being monitored by a security researcher for successful products being sold, we can first of all cut down significantly on the number of threads by just looking at the selling threads as classified by thread type. As we can see in our dataset of the 42587 total threads, only 8854 were classified as being selling, so that means a reduction of 79% of threads that need monitoring. After that the researcher can use the logistic regression to predict the successful threads and only focus on those threads, further decreasing the number of threads that need monitoring. However as the model has lower statics for prediction we would advise to use the logistic regression probability to prioritize threads instead. After doing this for a certain amount of time the security researcher could manually find a probability threshold after which there shouldn't be successful threads anymore, further cutting down on the number of threads that should be manually checked.

The methodology that we have proposed in this paper should generalize quite well to other forums, as the basic structure of these kinds of forums remains the same. However as has been noted in previous research [6], every forum is basically it's own little domain and cross forum classification will not work reliably. This means that for every forum the thread type classification and comment sentiment has to be trained. Furthermore on this forum the market place and rest of the forum are divided into subboards with specific categories. This means that for the thread category we did not have to train a classifier. For forums were this is not the case, or where there are a lot of threads posted to the wrong category, a classifier has to be trained to predict the thread category.

A. Limitations

There are a number of limitations that we were not able to overcome during this research. First of all our methodology was only tested on a single forum. Therefore we were not able to test how well the methodology works on other forums. Neither were we able to test if combining features from different forums can help with identifying successful products.

As is the case with our analyzed forum, to gain access you need to have reputation on other forums. It would be interesting to combine features retrieved across forums, to see how different forum react to the same product. We do not know the exact account age when a thread is posted as we do only have access to public data. As we saw in the results section this means that for a lot of threads that get posted the account age is still at zero days old. Similarly, the provided data may contain inaccuracies or parsing errors; for example, upon inspection of the data it is revealed that there are indeed some threads where comments are posted at the exact same time as the thread itself, this is probably a mistake by the web scraper.

Secondly we do not have a native understanding of the Russian language, which means we had to rely on translation software. This means that for the manual labeling of the thread type and especially the comment sentiment we are dependent on the quality of the translation.

Thirdly the process of manual labeling by one person has the risk of being biased towards a certain class. The normal procedure is that multiple people label everything and only the labels with agreement are actually used. We tried to mitigate this by having a second person verify a subset of the labels and discuss why it was labeled as such and see if there was agreement. While doing this process we did not find any bias towards a certain class.

Fourthly the sentiment classification for negative comments is of sub par quality. Which hinders not only the classification of the sentiment, but also effects all the following steps of the methodology and even the labeling of the ground truth.

Finally the graphs keep growing while the forum remains active. This means that for every thread that needs to be checked, calculating all the SNA features becomes slower over time. For this it might be useful to take time into account and remove certain threads and comments when they get too old as they might no longer be relevant.

VIII. Related work

In this section we look at work that is directly related to our current topic. These are works that try and classify or predict certain features based on data from these underground forums.

After Frei et al. [13] showed that there is a gap between exploit time and patch time, Bezorgi et al. [15] in 2010 started building a model using data mining and machine learning to try and classify vulnerabilities. Using this model they try and predict whether and how soon a vulnerability be exploited. Their model is trained on data retrieved from Open Source Vulnerability Database (OSVDB) (OSVDB no longer exists) [17] and the MITRE Common Vulnerabilities and Exposures (CVE) [16] database. In their research they argue that their approach can help prioritize the development of patches. However as already stated in the conclusion of the background work section, we know that relying only on data from CVEs is not reliable.

A year later in 2012, Holt [1] furthers the analysis of the social structures in these underground (forum based) markets. He takes a look at how the price, customer service and trust influences relationships between actors in these underground forums. Furthermore he draws parallels with real-world markets for illicit goods. This research is based on data gained from scraping these underground forums, this way only ”publicly” accessible data is gained, unlike the work of Motoyama et al. [7], where also the private messages where used in their analysis. They show that price, good customer service and positive feedback play an important role in the successfulness of a seller.

In 2015 Edkrantz et al. [21] build a framework to predict exploit likelihood and time frame for unseen security vulnerabilities. To build this framework they use historical data from NVD and EDB. On this data they used common machine learning techniques. Again they note that the data from NVD is of poor quality for statistical analysis.

In 2017 Portnoff et al. [23] proposed an automated top down approach for analyzing underground forums. They showed how their approach achieved an 80% accuracy in detecting post category, product and pricing. To reach this accuracy they used a hybrid system that uses support vector machine (SVM) to predict to post type and the product being sold and they use a combination of regex and SVM for the product pricing. Showing us how we can use SVM to help classify the type and category of a post.

At the same time Portnoff et al. [24] showed that classic natural language processing techniques have problems with out of domain data and that every forum is basically it’s own fine grained domain. This means we cannot use standard NLP tools to try and extract information from underground forum data. They also present a new dataset with manually annotated product references.

The following year in 2018, Caines et al. [25] present an annotation schema in their paper to label forum posts for three properties, namely post type, author intent and addressee. Their schema uses natural language processing and machine learning to accomplish this goal. For post type and category they use a logical model based on regular expressions, which they found does perform well except it has a low recall. To fix this they added a hybrid approach that uses machine learning to complement the logical model. This schema works best for post type and author intent, however only fairly well for addressee. They state that they too have found that standard NLP tools do not work well with underground forum data, as the structure of sentences and the jargon used are different from normal well written text.

Pastrana et al. [27] also released a dataset named CrimeBB to help other security researchers. The advantages of this dataset is that it is a huge dataset spanning a decade worth of data, that is kept up to date and spans four separate forums. To collect this data they developed a crawler named CrimeBot. This dataset is useful to verify that the methodology described in this paper also works on different datasets.

Another important aspect that underground markets provide is in the supply chain for cyber crime. In 2019 Bhalerao et al. [12] were the first to take a look at this. In their paper they

studied two forums. In these forums they classified product categories and reply categories using supervised classification. Having the product and reply categories they build an interaction graph. Using this interaction graph they can then build a supply chain that performs better than their baseline. Again showing an approach for classifying post and replies on underground forums and building a graph showing social interactions between users.

Chen et al. [28] came with a novel idea to use Twitter to try and predict when an exploit that is associated with a CVE be exploited. For this they build a multi-layered graph of author, tweet and CVE. Then they used an adapted Hawkes process to estimate the number of tweets/retweets related to a CVE. And finally they build two forecast models to predict when the CVE be exploited.

So these related works showed us how different approaches to classify post type, post category, comment type and comment intent. It showed us how we can can create graphs of the the data (forum, Twitter) and how to use social network analyses to retrieve features from this. Furthermore it show us how to use machine learning to predict certain outcomes based on features extracted by previous methods.

IX. Conclusions

Underground forums play an important role in the creation and distribution of attack vectors online. These forums have to be manually monitored, which is a labor intensive task. In this paper we propose a methodology that can help with predicting the successful products. We implemented the methodology on a test set using different NLP and SNA techniques. We did all this taking into account that we do not use data from the future for any of the features used in the prediction of successfulness. Finally we used logistic regression model to see how well our hypotheses are at explaining the variance between successfulness and our chosen features. We saw that when a new thread gets posted we are unable to explain any of the variance, however after one week we are already able to explain the majority of the variance. We found that using the commenters on a thread is by far the best prediction of a successful product.

Our analysis of the successful products is a first step into greater understanding what makes products successful on these underground forums, however more research is needed to be able to fully predict all successful products.

References

[1] T.J. Holt, "Examining the Forces Shaping Cybercrime Markets Online" in Social Science Computer Review, 2012 pp.165-177.

[2] C. Grier, L. Ballard, J. Caballero, N. Chachra, C. J. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, A. Nappa, A. Pitsillidis, N. Provos, M. Zubair Rafique, M. Abu Rajab, C. Rossow, K. Thomas, V. Paxson, S. Savage, and G. M. Voelker "Manufacturing Compromise: The Emergence of Exploit-as-a-Service", in Proceedings of the 19th ACM Conference on Computer and Communications Security, pp. 821-832, 2012.

[3] A.K. Sood, and R.J. Enbody. "Crimeware-as-a-service - A survey of commoditized crimeware in the underground market." In Critical Infrastructure. Protect., vol. 6, no. 1, pp. 2838, 2013.

[4] N. Kshetri. "Diffusion and effects of cyber-crime in developing economies." in Third World Quarterly, 31:7, pp. 10571079, 2010.

[5] Group IB. "State and Trends of the Russian digital crime market." TechnicalReport, 2011.

[6] R.S. Portnoff, S. Afroz, G. Durrett, J.K. Kummerfeld, T. Berg-Kirkpatrick, D. McCoy, K. Levchenko, and V. Paxson. "Tools for Automated Analysis of Cybercriminal Markets." In Proceedings of the 26th International Conference on World Wide Web, 2017.

[7] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G.M. Voelker. "An analysis of underground forums." In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, 2011.

[8] S. Samtani, R. Chinn and H. Chen, "Exploring hacker assets in underground forums", in IEEE International Conference on Intelligence and Security Informatics, pp. 31-36, 2015.

[9] S. Pastrana, A. Hutchings, A. Caines and P. Buttery. "Characterizing Eve: Analysing Cybercrime Actors in a Large Underground Forum" in 21st International Symposium, RAID 2018.

[10] F. Pendlebury, F. Pierazzi, and R. Jordaney, "TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time", in the Proceedings of the 28th USENIX Security Symposium, 2019.

[11] L. Allodi, "Economic Factors of Vulnerability Trade and Exploitation." in CCS, pp. 1483-1499, 2017.

[12] R. Bhalerao, M. Aliapoulios, I. Shumailov, S. Afroz and D. McCoy. "Mapping the Underground: Towards Automatic Discovery of Cybercrime Supply Chains", 2018.

[13] S. Frei, M. May, U. Fiedler and B. Plattner. "Large-scale vulnerability analysis." In Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense, 2006.

[14] S. Frei, D. Schatzmann, B. Plattner and B. Trammell. "Modeling the Security Ecosystem" In The Dynamics of (In)Security, 2010.

[15] M. Bozorgi, L.K. Saul, S. Savage and G.M. Voelker. "Beyond heuristics: learning to classify vulnerabilities and predict exploits." In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010.

[16] CVE Editorial Board. Common Vulnerabilities andExposures: The Standard for Information SecurityVulnerability Names. https://cve.mitre.org/https://cve.mitre.org/

[17] OSVDB. The Open Source Vulnerability Database. https://blog.osvdb.org/https://blog.osvdb.org/

[18] C. Herley and D. Florencio. "Nobody Sells Gold for the Price of Silver: Dishonesty, Uncertainty and the Underground Economy." In Economics of Information Security and Privacy, pp. 33-53, 2010.

[19] L. Allodi and F. Massacci. "A preliminary analysis of vulnerability scores for attacks in wild" In Proceedings of the Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, pp. 17-24, 2012.

[20] L. Allodi, W. Shim and F. Massacci, "Quantitative Assessment of Risk Reduction with Cybercrime Black Market Monitoring" In IEEE Security and Privacy Workshops, pp. 165-172, 2013.

[21] M. Edkrantz and S. Alan. "Predicting Cyber Vulnerability Exploits with Machine Learning." IN SCAI, 2015.

[22] L. Allodi, M. Corradin and F. Massacci. "Then and now: on the maturity of the cybercrime markets the lesson that black-hat marketeers learned" In IEEE Transactions on Emerging Topics in Computing, vol. 4, no. 1, 7044581, pp. 35-46, 2016.

[23] R.S. Portnoff, S. Afroz, G. Durrett, J.K. Kummerfeld, T. Berg-Kirkpatrick, D. McCoy, K. Levchenko, and V. Paxson. "Tools for Automated Analysis of Cybercriminal Markets." In Proceedings of the 26th International Conference on World Wide Web, pp. 657666, 2017.

[24] G. Durrett, J. Kummerfeld, T. Berg-Kirkpatrick,R. Portnoff, S. Afroz, D. Mccoy, K. Levchenko and V. Paxson, Vern. "Identifying Products in Online Cybercrime Marketplaces: A Dataset for Fine-grained Domain Adaptation." 2017.

[25] A. Caines, S. Pastrana, A. Hutchings and P. Buttery. "Automatically identifying the function and intent of posts in underground forums." In Crime Science. 7, 2018.

[26] S. Pastrana, A. Hutchings, A. Caines and P. Buttery. "Characterizing Eve: Analysing Cybercrime Actors in a Large Underground Forum." In 21st International Symposium, RAID, 2018.

[27] S. Pastrana, D. Thomas, A. Hutchings and R. Clayton. "CrimeBB: Enabling Cybercrime Research on Underground Forums at Scale.", 2018.

[28] H. Chen, R. Liu, N. Park, and V.S. Subrahmanian. "Using Twitter to Predict When Vulnerabilities be Exploited." In Proceedings of the 25th

ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.
[29] G.A. Akerlof. "The Market for Lemons: Quality Uncertainty and the Market Mechanism", 1970
[30] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov. "Bag of Tricks for Efficient Text Classification" In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp 427-431, 2017
[31] M. Lui and T. Baldwin "langid.py: An Off-the-shelf Language Identification Tool". In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, 2012
[32] E. Loper and S. Bird. "NLTK: the Natural Language Toolkit". In ETMTNLP '02: Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1, pp. 63-70, 2002
[33] M. Honnibal and M. Johnson, "An Improved Non-monotonic Transition System for Dependency Parsing" In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1373-1378, 2015
[34] P. Qi, Y. Zhang, Y. Zhang, J. Bolton and C.D. Manning. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". 2020
[35] T. Chen and C. Guestrin."XGBoost: A Scalable Tree Boosting System." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp 785-794, 2016.
[36] P. Peixoto. "The graph-tool python library", 2014.
[37] M. Yip, N. Shadbolt and C. Webber. "Why forums? An empirical analysis into the facilitating factors of carding forums" In proceedings of the 5th Annual ACM Web Science Conference, pp. 453-462, 2013.

APPENDIX

For fastText we use the sanitized text directly in the classifier as fastText is a text classification algorithm that works by obtaining vector representations for words. For all other classifiers, we apply the following techniques to extract features from the thread posts. First we convert the thread posts to a document-term matrix. A document-term matrix is a matrix that describes the frequency of all words in the documents (in our case the thread posts) by putting all words found in the documents as columns, the documents as rows and the frequencies as values. Having build this document-term matrix we drop all words that have a frequency of one or less. This document-term matrix is then converted using term frequency, inverse document frequency (TF-IDF). We use TF-IDF as a weighting factor. In TF-IDF a word value increases proportionally to frequency in the document and is offset by the frequency of all documents containing that word. We choose to also use unigram and bigram words, what this means is that we not only look at the words, but also at the occurrence of two words following each other.

Table/Field	Description
Members	Table containing all members of the forum
id	Unique identification number of the member
name	Username of the member
join_date	join date of the member based on either first thread post or first comment made by the member
Resource	Table containing all different resources (the two different forums)
id	Unique identifier of the resource
name	Name of the resource
Section	Table containing the (sub)sections of the forum
id	Unique identification number of the section
resource_id	The id of the resource this section belongs to
name	The name of the resource
Thread	Table containing all the threads
id	Unique identification number of the thread
section_id	The section where the thread is posted
name	Title of the thread
content	Contents of the thread post
posted	Post date of the thread
author_id	The author of the thread
Post	Table containing the posts (comments)
id	Unique identification number of the post
thread_id	The thread where the comment was posted under
content	The contents of the comment
posted	Post date of the comment
author_id	The author of the comment
sequence_number	The sequence number of the comment on the thread

TABLE X
STRUCTURE OF THE DATASET AS OBTAINED BY MR. MARTIN POZDENA, AUXILIUM, WITH SHORT DESCRIPTION OF WHAT EVERY FIELD IS USED FOR.

Model	Sell			Other		
	P	R	F1	P	R	F1
regex	0.54	0.91	0.68	0.97	0.78	0.87
SVM	0.82	0.83	0.83	0.90	0.90	0.90
SVM meta	0.76	0.82	0.79	0.90	0.87	0.88
SGD	0.90	0.75	0.82	0.82	0.94	0.88
SGD meta	0.80	0.79	0.80	0.88	0.88	0.88
Logistic regression	0.83	0.83	0.83	0.90	0.90	0.90
Logistic regression meta	0.62	0.73	0.67	0.87	0.80	0.83
XGBoost	0.66	0.89	0.76	0.95	0.83	0.89
XGBoost meta	0.66	0.89	0.76	0.95	0.83	0.89
fastText	0.73	0.91	0.81	0.96	0.86	0.90

TABLE XI
PRECISION, RECALL AND F1-SCORES FOR RUSSIAN (WITHOUT `iso9`) THREAD TYPE CLASSIFICATION

Model	Positive			Negative			Other		
	P	R	F1	P	R	F1	P	R	F1
SVM	0.79	0.83	0.81	0.24	0.55	0.33	0.88	0.76	0.82
SVM*	0.80	0.80	0.80	0.25	0.54	0.34	0.86	0.77	0.81
SGD	0.80	0.79	0.79	0.29	0.48	0.36	0.83	0.77	0.8
SGD*	0.79	0.76	0.77	0.27	0.47	0.34	0.82	0.77	0.79
LogReg	0.81	0.82	0.81	0.35	0.49	0.40	0.84	0.78	0.81
LogReg*	0.82	0.74	0.78	0.30	0.48	0.37	0.79	0.78	0.79
XGBoost	0.58	0.78	0.67	0.13	0.52	0.19	0.91	0.69	0.78
XGBoost*	0.59	0.79	0.67	0.14	0.51	0.20	0.91	0.69	0.78
fastText	0.74	0.84	0.78	0.09	0.67	0.16	0.93	0.73	0.82

TABLE XII
PRECISION, RECALL AND F1-SCORES FOR SENTIMENT CLASSIFICATION OF RUSSIAN (WITHOUT `iso9`) COMMENTS. MODEL* IS THE CLASSIFICATION WITH METADATA

Board	Subboard	Category
About	About	Other
About	Hacking Materials (Articles, Videos)	Hacking
About	Jabber Server	Other
About	Vulnerability Testing Service	Other
Hacking	Anonymity and Privacy	Other
Hacking	Cryptography	Crypto
Hacking	Hacking and Security	Hacking
Hacking	Hacking and Security - Bugtrack	Hacking
Hacking	IM Messengers	Account
Hacking	Malware	Malware
Hacking	Malware - Samples	Malware
Hacking	Money	Financial
Hacking	Money - Articles	Financial
Hacking	Money - Casino	Financial
Hacking	Social Engineering	Social Engineering
Hacking	Spamming	Spam
Hacking	Wardriving and Bluejacking	Other
Marketplace	Auction	Other
Marketplace	Black List - Arbitration	Other
Marketplace	Black List - Black List	Other
Marketplace	Buy/Sell - Accesses, FTP, Shells, SQLi, etc	Hacked server
Marketplace	Buy/Sell - Financial, Bank Accounts	Financial
Marketplace	Buy/Sell - Labor market	Hack-for-hire
Marketplace	Buy/Sell - Malware, Exploits, EK, Crypter, A3	Malware
Marketplace	Buy/Sell - Mobile Connections, SMS, Call reception, Exploitation	Mobile
Marketplace	Buy/Sell - Others	Other
Marketplace	Buy/Sell - Payment Systems	Financial
Marketplace	Buy/Sell - Rules, Verification, Garant	Other
Marketplace	Buy/Sell - Servers, Hosting, Domains, Proxy	Hosting
Marketplace	Buy/Sell - Social Networks, Access, Spamming	Account
Marketplace	Buy/Sell - Spamming	Spam
Marketplace	Buy/Sell - Traffic	Traffic
Offtopic	Flame	Other
Offtopic	Flame - Games, Music, Movies	Entertainment
Offtopic	Flame - Jokes	Entertainment
Offtopic	Legal Business	Other
Offtopic	Legal Questions	Other
SW and HW	Cryptocurrencies	Crypto
SW and HW	HW	Other
SW and HW	Mobile	Mobile
SW and HW	Operating Systems	OS
SW and HW	SW	Other
Tools	DB Dumps, Leaks	Dumps
Tools	SW	Tools
Web	Development	Web
Web	Programming	Web
Web	Programming - Scripts	Web
Web	SEO	Web

TABLE XIII
MAPPING BETWEEN BOARDS/SUBBOARDS AND CATEGORIES

Topic	Description	Threads
Account	Access to and trading of social network accounts, mail accounts, blog accounts	3%
Auction	Auction of all kinds of assets, malware, database dump, etc.	3%
Black list	Black list and arbitration of members	5%
Crypto	Cryptography and cryptocurrencies	1%
Dumps	Dumps of databases and other data	1%
Entertainment	All entertainment discussions, movies, games, etc.	<1%
Financial	Trading and discussion on money, bank accounts and payment systems	16%
Hacked server	Hacked server for sale	5%
Hacking	General hacking related activities	5%
Hack for hire	Labor market for looking and offering hacking related jobs	9%
Hosting	Hosting	2%
Malware	General malware related trading and discussions	11%
Mobile	Mobile related asset trading and discussions	2%
OS	Operating system related asset trading and discussions	1%
Other	All other topics	14%
Social engineering	Social engineering	1%
Spam	Spam trading and infrastructure	10%
Tools	General tools trading and discussions	1%
Traffic	Trade of traffic	6%
Web	Web related trading and discussions	3%

TABLE XIV
TOPICS USED ON THE FORUM AND A SHORT DESCRIPTION OF WHAT THE TOPIC ENTAILS WITH THE PERCENTAGE OF THREADS POSTED IN THAT TOPIC

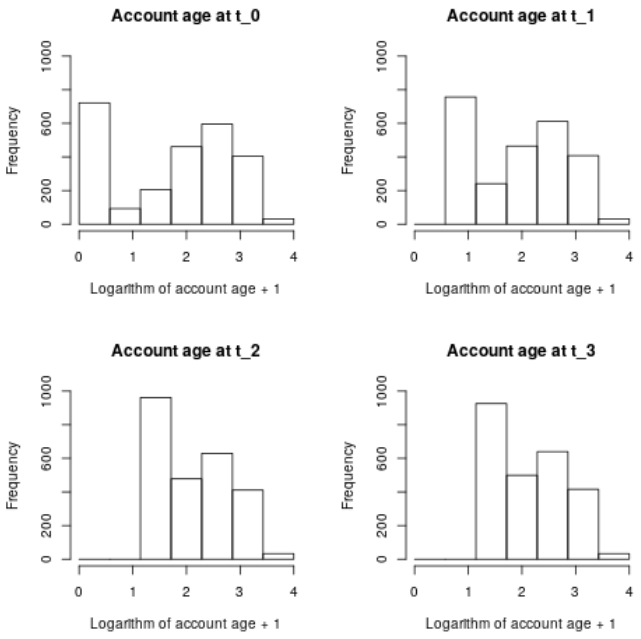

Fig. 7. Distribution of account age at t_0 to t_3

Category	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
Account	0.00	7.69	20.0	12.5	0.00	0.0	0.00	2.63	1.67	2.91	3.35	3.30	3.30	2.69
Auction	0.00	0.00	0.0	0.0	0.00	0.0	0.00	0.00	0.00	0.00	0.00	4.59	8.99	21.16
Crypto	0.00	0.00	10.0	0.0	0.00	4.0	3.23	2.63	0.00	0.97	0.42	0.81	0.36	0.32
Dumps	0.00	0.00	0.0	0.0	0.00	4.0	0.00	0.00	0.00	0.00	0.00	0.96	1.49	1.60
Financial	0.00	0.00	0.0	0.0	6.25	0.0	3.23	7.89	3.33	4.85	17.57	15.78	15.88	16.84
Hacked server	0.00	0.00	10.0	12.5	6.25	0.0	6.45	15.79	1.67	8.74	9.62	9.42	9.48	7.23
Hacking	33.33	15.38	0.0	12.5	12.50	8.0	3.23	0.00	5.00	6.80	2.09	5.93	4.56	3.46
Hack for hire	0.00	0.00	0.0	0.0	0.00	4.0	19.35	5.26	11.67	3.88	6.28	3.59	3.53	3.75
Hosting	0.00	0.00	10.0	25.0	18.75	20.0	12.90	15.79	20.00	11.65	7.53	2.97	3.33	2.37
Malware	8.33	23.08	10.0	12.5	6.25	8.0	12.90	10.53	18.33	22.33	15.06	13.30	10.90	8.83
Mobile	25.00	7.69	0.0	0.0	0.00	4.0	3.23	5.26	5.00	3.88	5.02	4.83	3.27	2.08
OS	8.33	7.69	0.0	0.0	6.25	0.0	0.00	0.00	1.67	0.00	1.67	1.00	0.29	0.35
Other	16.67	7.69	10.0	0.0	25.00	32.0	9.68	5.26	13.33	17.48	10.46	10.47	11.55	11.24
Social engineering	0.00	0.00	0.0	0.0	0.00	8.0	0.00	0.00	0.00	0.97	2.09	0.91	1.00	0.22
Spam	0.00	0.00	30.0	0.0	18.75	0.0	6.45	18.42	10.00	7.77	10.04	14.63	13.68	9.80
Tools	0.00	7.69	0.0	0.0	0.00	0.0	0.00	0.00	0.00	0.97	0.84	0.67	1.23	0.93
Traffic	0.00	0.00	0.0	0.0	0.00	4.0	16.13	10.53	5.00	6.80	6.69	4.78	5.69	5.79
Web	8.33	23.08	0.0	25.0	0.00	4.0	3.23	0.00	3.33	0.00	1.26	2.06	1.46	1.3

TABLE XV
THE PERCENTAGE OF SELLING THREADS POSTED PER CATEGORY OVER THE YEARS

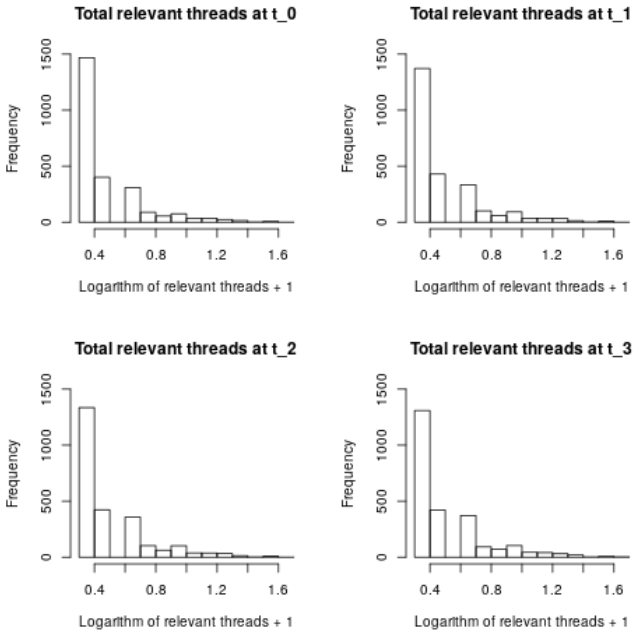

Fig. 8. Distribution of relevant threads at t_0 to t_3

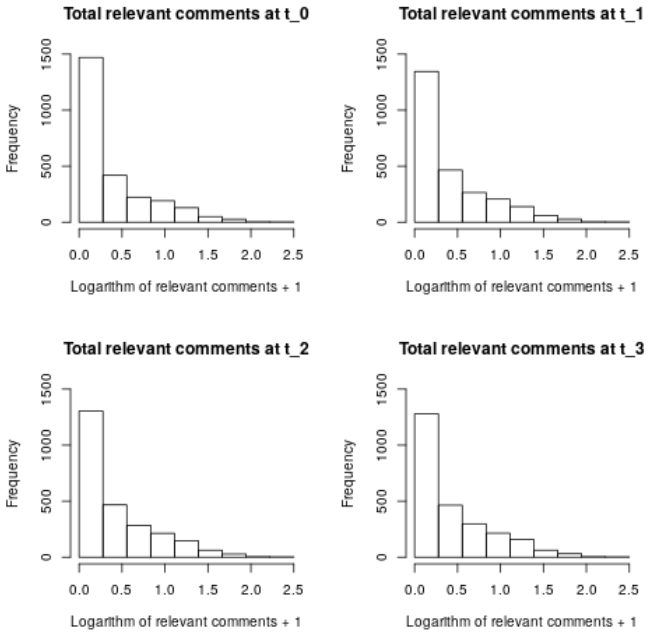

Fig. 9. Distribution of relevant comments at t_0 to t_3

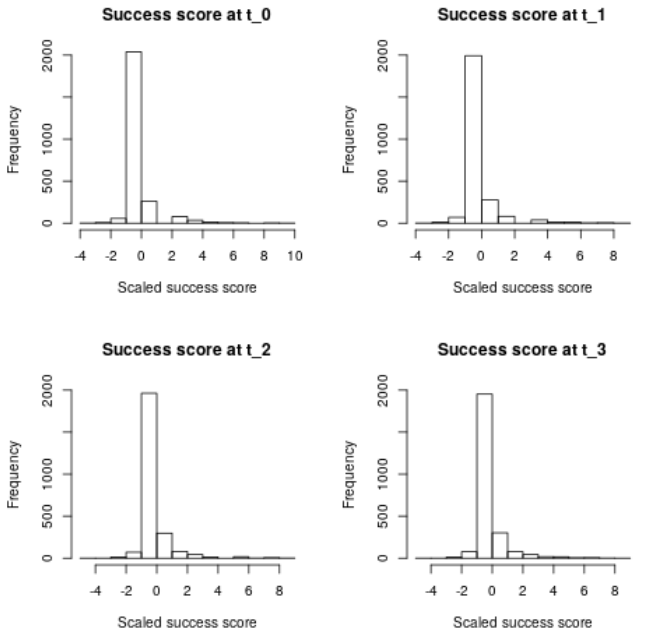

Fig. 10. Distribution of success score at t_0 to t_3

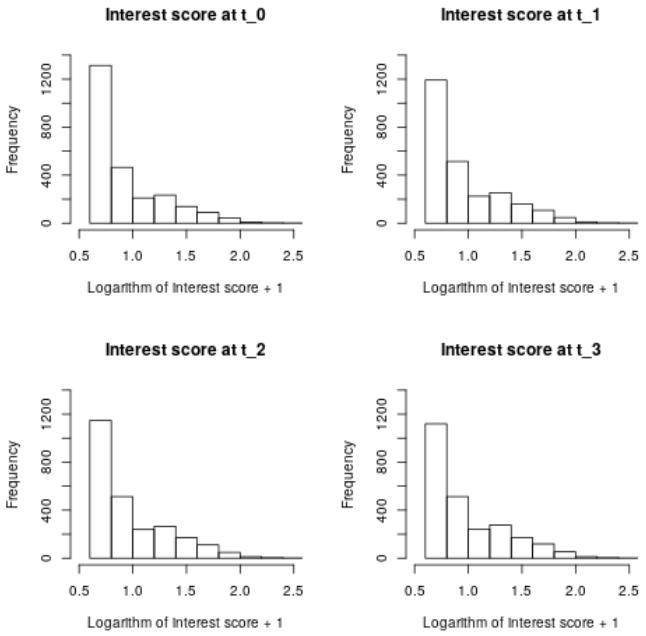

Fig. 12. Distribution of interest score at t_0 to t_3

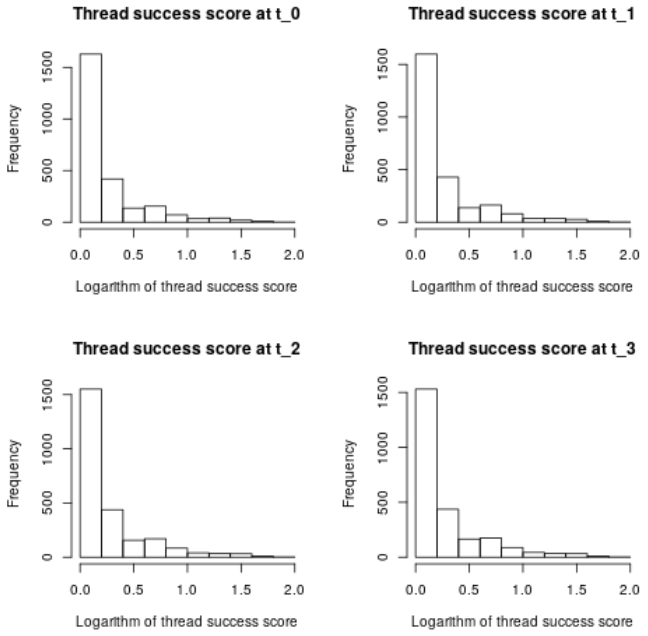

Fig. 11. Distribution of thread success score at t_0 to t_3

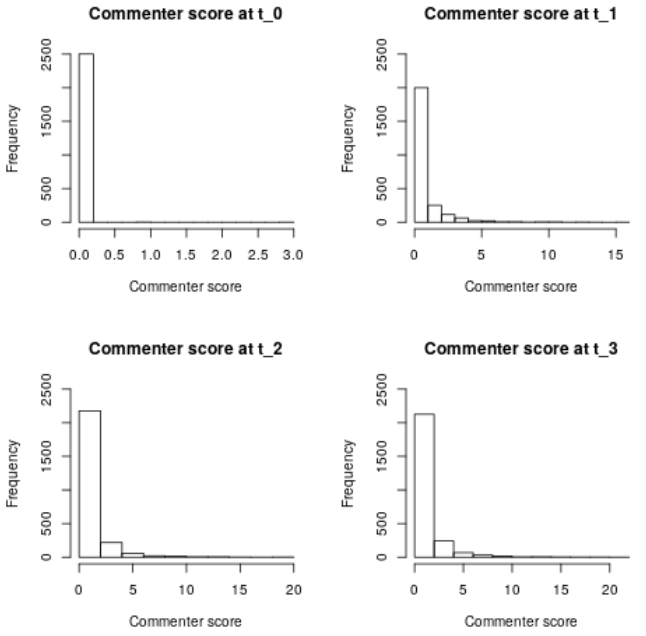

Fig. 13. Distribution of commenter score at t_0 to t_3

	Model 1	Model 2	Model 3	Model 4	Model 5
(Intercept)	-0.51^{***}	-0.91^{***}	-0.93^{***}	-0.93^{***}	-0.51^{***}
	(0.04)	(0.03)	(0.03)	(0.02)	(0.04)
log(account age)	-0.11^{***}				-0.12^{***}
	(0.01)				(0.01)
log(relevant thread)	-0.37				-0.72^{*}
	(0.22)				(0.36)
log(relevant comments)	0.18^{***}				0.11
	(0.04)				(0.06)
scale(relevant success)		0.02			0.01
		(0.02)			(0.02)
log(thread success score)		-0.04			0.07^{*}
		(0.03)			(0.04)
log(relevant interest)			-0.04		0.47
			(0.18)		(0.39)
commenter score > 0				13.50	13.40
				(162.37)	(161.26)
AIC	10421.25	10552.59	10552.48	10542.42	10414.23
BIC	10449.61	10573.86	10566.66	10556.60	10470.94
Log Likelihood	-5206.63	-5273.30	-5274.24	-5269.21	-5199.12
Deviance	10413.25	10546.59	10548.48	10538.42	10398.23
Num. obs.	8854	8854	8854	8854	8854
Pseudo R2	0.02	0.00	0.00	0.00	0.02

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

TABLE XVI

STATISTICAL MODELS FOR LOGISTIC REGRESSION AT t_0

	Model 1	Model 2	Model 3	Model 4	Model 5
(Intercept)	-0.51^{***}	-0.90^{***}	-0.93^{***}	-1.74^{***}	-1.20^{***}
	(0.04)	(0.03)	(0.03)	(0.03)	(0.09)
log(account age)	-0.11^{***}				-0.11^{***}
	(0.01)				(0.02)
log(relevant thread)	-0.37				0.81^{**}
	(0.22)				(0.28)
log(relevant comments)	0.18^{***}				0.20^{**}
	(0.04)				(0.06)
scale(relevant success)		0.02			0.04
		(0.02)			(0.02)
log(thread success score)		-0.06			-0.04
		(0.03)			(0.04)
log(relevant interest)			0.00		-0.86^{*}
			(0.17)		(0.36)
commenter score > 0				2.34^{***}	2.32^{***}
				(0.06)	(0.06)
AIC	10421.25	10550.45	10552.53	8563.44	8514.14
BIC	10449.61	10571.72	10566.71	8577.62	8570.85
Log Likelihood	-5206.63	-5272.23	-5274.26	-4279.72	-4249.07
Deviance	10413.25	10544.45	10548.53	8559.44	8498.14
Num. obs.	8854	8854	8854	8854	8854
Pseudo R2	0.02	0.00	0.00	0.29	0.30

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

TABLE XVIII

STATISTICAL MODELS FOR LOGISTIC REGRESSION AT t_2

	Model 1	Model 2	Model 3	Model 4	Model 5
(Intercept)	-0.51^{***}	-0.90^{***}	-0.93^{***}	-1.54^{***}	-1.01^{***}
	(0.04)	(0.03)	(0.03)	(0.03)	(0.07)
log(account age)	-0.11^{***}				-0.12^{***}
	(0.01)				(0.02)
log(relevant thread)	-0.37				0.72^{**}
	(0.22)				(0.28)
log(relevant comments)	0.18^{***}				0.21^{***}
	(0.04)				(0.06)
scale(relevant success)		0.02			0.03
		(0.02)			(0.02)
log(thread success score)		-0.06^{*}			-0.05
		(0.03)			(0.04)
log(relevant interest)			-0.01		-0.78^{*}
			(0.17)		(0.35)
commenter score > 0				2.09^{***}	2.07^{***}
				(0.06)	(0.06)
AIC	10421.25	10549.63	10552.52	9046.13	8976.83
BIC	10449.61	10570.90	10566.70	9060.30	9033.53
Log Likelihood	-5206.63	-5271.82	-5274.26	-4521.06	-4480.41
Deviance	10413.25	10543.63	10548.52	9042.13	8960.83
Num. obs.	8854	8854	8854	8854	8854
Pseudo R2	0.02	0.00	0.00	0.22	0.24

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

TABLE XVII

STATISTICAL MODELS FOR LOGISTIC REGRESSION AT t_1

	Model 1	Model 2	Model 3	Model 4	Model 5
(Intercept)	-0.51^{***}	-0.90^{***}	-0.93^{***}	-1.85^{***}	-1.31^{***}
	(0.04)	(0.03)	(0.03)	(0.04)	(0.10)
log(account age)	-0.11^{***}				-0.11^{***}
	(0.01)				(0.02)
log(relevant thread)	-0.37				0.91^{**}
	(0.22)				(0.28)
log(relevant comments)	0.18^{***}				0.22^{***}
	(0.04)				(0.06)
scale(relevant success)		0.02			0.04
		(0.02)			(0.03)
log(thread success other)		-0.06			-0.05
		(0.03)			(0.04)
log(relevant interest)			-0.03		-1.04^{**}
			(0.17)		(0.37)
commenter score > 0				2.49^{***}	2.47^{***}
				(0.06)	(0.06)
AIC	10421.25	10550.21	10552.50	8282.24	8239.00
BIC	10449.61	10571.47	10566.68	8296.42	8295.71
Log Likelihood	-5206.63	-5272.10	-5274.25	-4139.12	-4111.50
Deviance	10413.25	10544.21	10548.50	8278.24	8223.00
Num. obs.	8854	8854	8854	8854	8854
Pseudo R2	0.02	0.00	0.00	0.32	0.33

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

TABLE XIX

STATISTICAL MODELS FOR LOGISTIC REGRESSION AT t_3