Securing Tag-based recommender systems against profile injection attacks: A comparative study

Georgios Pitsilis
Department of Computer Science
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
georgios.pitsilis@ntnu.no

Heri Ramampiaro
Department of Computer Science
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
heri@ntnu.no

Helge Langseth
Department of Computer Science
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
helge.langseth@ntnu.no

ABSTRACT
This work addresses challenges related to attacks on social tagging systems, which often comes in a form of malicious annotations or profile injection attacks. In particular, we study various countermeasures against two types of threats for such systems, the Overload and the Piggyback attacks. The studied countermeasures include baseline classifiers such as, Naive Bayes filter and Support Vector Machine, as well as a deep learning-based approach. Our evaluation performed over synthetic spam data, generated from del.icio.us, shows that in most cases, the deep learning-based approach provides the best protection against threats.

KEYWORDS
recommender systems; collaborative tagging; attacks; del.icio.us

ACM Reference Format:

1 INTRODUCTION
Recommender Systems (RS) are information filtering mechanisms, aiming at predicting the preference of users to particular resources. Social Tagging systems, facilitate resource recommendations using the users’ assigned annotations to resources, (aka folksonomies), as input to prediction algorithms. The novelty of using tags for annotation has attracted the interest of scientists in RS [1, 2], as well as of attackers. In general, an attack against a tagging system consists of coordinated malicious profiles that correspond to fictitious identities, injected into the system, for biasing the recommendation algorithm towards suggesting inferior products to users.

The issue of security in tag-based RS has so far been mainly approached using anti-spamming techniques such as, Bayesian type filtering [8], or other tag classification methods [5]. In the above works, particular characteristics of the tags used in annotations are exhibited, with the assumption that tags used by legitimate users would coexist with each other. Nevertheless, such filtering becomes ineffective if attackers are aware of the attack filtering policy [5, 8]. Other feature-based countermeasures, employ either the neighbors’ honesty within the group [9], or mix features of tags and users together along with other, derived from the social connectivity [6]. Neural network-based approaches, including deep learning (DL), are known to provide good protection against spamming [3]. However, DL has neither been adequately investigated so far for safeguarding tag-based RS, nor quantifiable results exist to show the effectiveness of the countermeasures on the recommendations.
The research question we address is: How effective are the various classifier schemes against profile injection attacks, that aim to influence the personalized recommendations in a tag-based RS? To answer this question, our main goals are: i) to study various classification schemes, over known types of shilling attacks, and ii) to evaluate the effectiveness of classification into the recommendation process.

Our contribution in this paper is two fold: i) a synthetic set of malicious data to serve testing purposes, and ii) a comparative study of the effects of known attacks and the effectiveness of potential countermeasures against them, in a typical tag-based RS. Those include a properly adapted DL model.

2 SECURING THE FOLKSONOMIES
In this work, we focus on two forms of intrusions known as Overload and Piggyback attacks [7]. The goal of the former is to overload a tag context with a bogus resource to achieve correlation between the tag and that resource. To accomplish this, an attacker associates the bogus resource with a number of popular tags. For the latter, the objective is the bogus resource to ride the success of another highly popular one. To achieve that, an attacker would annotate the bogus resource choosing any popular tags already associated with the popular one, so that they appear similar. Our comparative study includes the following algorithms:

Naive Bayes filtering: It is a quite known classifier for detecting spam emails based on the Bayes theorem, here applied for classifying folksonomies based on the existence of tags in them.

Support Vector Machine (SVM): The input folksonomies were first vectorized and transformed into TF-IDF values to scale down the impact of the most frequent tags. Then, they were classified into legitimate and malicious, using the linear kernel function.

![Figure 1: The deep learning model architecture used](image-url)

Deep Learning (DL): We used a hybrid classification model which employs a Long-Short-Term-Memory (LSTM)-based Recurrent Neural Network and works both with and without sequential data. It
is composed of four layers (see Fig.1): a) an Embedding layer with size relevant to the vocabulary of tags used in the folksonomies (set to 25), b) a hidden layer, fully connected to the input and the subsequent layer, with dimensionality of its outer space set to 250, based on preliminary results, c) a dense layer, used for improving the learning and stabilize the output, of size equal to the input folksonomy, and d) a single neuron output layer provides the classification output in the form of probabilities for the Legitimate class.

3 EXPERIMENTS

To demonstrate the effects of the attacks on the recommendations, we employed Vector Space, a legacy algorithm for personalized recommendations in tag-based RS, which we adapted to our particular case. First, we express every tag in the corpus as word2vec vector by Google’s pre-trained set. Next, every user-posted folksonomy is represented by a single vector, by averaging the word2vec values of all tags in that folksonomy. As such, every user, or resource acquires its own vector representation, by combining together all folksonomy vectors associated with them. Finally, the personalized top-k recommendations for a user is build using the Cosine Similarity of his vector and the resources' vectors.

3.1 Dataset and Evaluation Metrics

For every run in our evaluation we selected randomly subsets of 3k users from del.icio.us dataset, corresponding to 73k folksonomies that form corpuses of 42k different tags. Due to lack of pre-labeled bogus data, we built synthetic bogus folksonomies of size and tag content determined according to guidelines found in the literature [7]. As such, to simulate the Overload attack, the tags of the fake folksonomies were chosen out of the 75 most popular ones used in the legitimate folksonomies, while the max size of the fake ones was limited to 50 tags. The actual size of a fake folksonomy was chosen so that, legitimate and fake ones will follow the same distribution. The popular tags were selected from those used for annotating the most popular resources. The impact of the attacks is demonstrated via a set of approved metrics [7] we adopted, which are: 1) the F-Score for the spam classification, 2) the Avg. rank of the bogus resource in the users’ top-k lists, and 3) the population affected by the attack, (users been recommended a bogus resource). For the Piggyback attack, the last metric refers to users for whom the bogus resource has been ranked higher than the popular one.

3.2 Experimental Setup

For training the classifiers we appended 30% fake synthetic folksonomies onto the set of legitimate ones. For the testing, we chose variable attack size, ranging from 0.1% to 10%, which refers to the ratio of the fake folksonomies over the legitimate ones.

To demonstrate the effectiveness of each algorithm, for each setup, fake folksonomies and legitimate ones were mixed together and supplied into the vector space model to compile the top-k recommendations (k was set to 15) for each user, both before and after applying the countermeasures. For all three filtering algorithms we tested, we performed 10-fold cross validation over the sample data. The DL model was implemented in the Keras toolkit, applied ADAM optimization [4] and selected categorical cross-entropy as the learning objective. Also, we modeled the input folksonomies (Fig. 1) in the form of vectors using word-based frequency vectorization. Finally, the DL model was trained for an optimal number of epochs.

3.3 Results and Discussion

The results are the average values of five runs. DL outperforms the other approaches (see Table 1) in classifying the legitimate and bogus folksonomies, for both attacks. As far as the recommendation service (see Fig. 2), very interestingly, even attacks of small scale are enough to render a significant population of users vulnerable. In fact, the DL approach, in comparison to the other alternatives, provides in general, good resistance to intrusions of bogus resources into the users top-k lists, for both attacks. Also, in terms of the Avg. Rank of the Bogus resource (see Fig. 2, right), DL scales better for large sizes of attacks vs the Bayes classifier, but performs best for small attacks only. For the same metric in the piggyback attack, DL despite being the second best performing, it also does better for large attacks, as opposed to Bayes classifier.

<table>
<thead>
<tr>
<th>Population affected by the attack (%)</th>
<th>Size of overload attack</th>
<th>F-score</th>
<th>SVM</th>
<th>BAYES</th>
<th>DL</th>
<th>Size of piggyback attack</th>
<th>F-score</th>
<th>SVM</th>
<th>BAYES</th>
<th>DL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1%</td>
<td>0.2%</td>
<td>0.5863</td>
<td></td>
<td></td>
<td></td>
<td>0.9014</td>
<td>0.9391</td>
<td>0.7748</td>
<td>0.9082</td>
<td></td>
</tr>
<tr>
<td>0.2%</td>
<td>0.4%</td>
<td>0.8339</td>
<td></td>
<td></td>
<td></td>
<td>0.9909</td>
<td>0.9788</td>
<td>0.9378</td>
<td>0.9818</td>
<td></td>
</tr>
<tr>
<td>1%</td>
<td>2%</td>
<td>0.9570</td>
<td></td>
<td></td>
<td></td>
<td>0.9709</td>
<td>0.9890</td>
<td>0.9638</td>
<td>0.9728</td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>25%</td>
<td>0.9051</td>
<td></td>
<td></td>
<td></td>
<td>0.9501</td>
<td>0.9426</td>
<td>0.9426</td>
<td>0.9426</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: The affected population (small values indicate strong resistance), and the rank of bogus resource (large values indicate strong resistance)

4 CONCLUSIONS AND FUTURE WORK

In this paper, we investigated the impact of spam filtering in tag-based RS. We simulated two known attacks, by generating fake data from original, taken from del.icio.us. Our experiments showed that our deep learning model outperforms all the legacy classifiers in terms of F-score and, in most cases, it can safeguard the user recommendations. Our future work includes experimentation with feature extraction from folksonomy data to feed the neural network, as well as generalizing our results by exploring more datasets.

REFERENCES