A Collaboration Model to Recommend Network Security Alerts Based on the Mixed Hybrid Approach

Arthur de Moura Del Esposte[1], Rodrigo Campiolo[1,2], Fabio Kon[1], Daniel Batista[1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (IME-USP)
Rua do Matão, 1010 – 05508-090 – Cidade Universitária – São Paulo – SP – Brasil

[2]Universidade Tecnológica Federal do Paraná (UTFPR) - Campo Mourão - PR - Brasil

{esposte, kon, batista}@ime.usp.br, rcampiolo@utfpr.edu.br

Abstract. *A high number of cyber security alerts are shared every second in different medias like forums, mail lists, and online social networks. The flood of alerts complicates the network administrator's job, since not all cyber security alerts are important for his/her specific environment. Thus, recommender system techniques could be properly used to filter cyber security alerts based on network administrator ratings and preferences. This paper presents a collaboration model to recommend cyber security alerts for network administrators, helping them to focus on the relevant alerts. To evaluate the model, an offline experiment is executed. Partial results showed that our model can be used to recommend cyber security alerts.*

1. Introduction

The Internet has grown significantly in recent years, especially in number of users, offered services, and transferred data. Thereby, most human activity fields, such as education, health, business, economy, and social interactions have evolved with the expansion of the Internet. More and more people, enterprises, governments, and critical infrastructures depend on information and communication technologies (ICT) and the infrastructure that make up cyberspace.

To address the challenges related to ICT dependence, such as the confidentiality, availability and integrity, it is necessary to increase the cyber security in all layers related to it, from physical components to organizational procedures. However, the risk to information and computer assets comes from a broad spectrum of threats, that grows as new vulnerabilities emerge [GCHQ and Cert-UK 2015a].

Formally, a vulnerability can be defined as an instance of failure in the specification, development or software configuration so that its execution can violate security polices, implicitly or explicitly [Krsul 1998]. Vulnerabilities can be maliciously exploited to allow unauthorized access, privileges changes and denial of service. Although the ICT is composed of many physical, logical or organizational components, most of the exploitable vulnerabilities is present in software. According to the Internet Security Threat Report [GCHQ and Cert-UK 2015b], in 2014 76% of websites contained vulnerabilities.

The more new cyber security issues arise, the more network administrators must update and improve their knowledge about these issues to implement preventive actions on computer networks and services. As long as many new attacks occur and new vulnerabilities are discovered, many alerts and news are released in all sorts of media. Specialized

Websites and email lists are examples of traditional medias used to warn about security threats, though they are not always effective in quickly publishing recent threats [Frei et al. 2006]. As discussed by Santos and collaborators [Santos et al. 2013], alternative decentralized medias, such as social networks, provide rich and heterogeneous data sources that can be used as means of spreading security notifications. This speech is reinforced by Surjandari and colleagues [Surjandari et al. 2015] who claim that social networks are valuable platforms for tracking and analyzing information, since people post real time messages about their opinions on a variety of topics and disseminate information quickly and collaboratively.

Despite the large number of cyber security news and warnings generated over the Internet every day, a network administrator not always gets relevant information. The large number of data sources can become a problem since it is a hard task to find or filter relevant information of interest. Historically, a natural way to accomplish this is relying on recommendations from people of similar interest profile or on expert advice. Ekstrand et al. [Ekstrand et al. 2011] expose that computer-based systems provide the opportunity to expand the set of people from whom users can obtain recommendations. They also enable us to mine users' history and stated preferences for patterns that neither them nor their acquaintances identify, potentially providing a more finely-tuned selection experience. The set of computer systems that provide those services are called *recommender systems*.

In view of what has been mentioned, this paper focuses on the design and evaluation of a recommender model for cyber security alerts extracted from external unstructured data, based on a *mixed hybrid* approach, to properly improve the hard task to get valuable cyber security alerts and news. Two main recommendation techniques are used in the designed model: Content-based and Collaborative Filtering. We also present a prototype implementation of this model, which is used to evaluate it through an offline experiment.

The remainder of this paper is structured as follows: Section 2 presents the main concepts related to recommender systems. In Section 3 we review and discuss related works. Section 4 describes the design and how we defined the recommender model, followed by the explanation of the method used to evaluate it. We show the experimental results in Section 5. In Section 6, we present the conclusion and future directions.

2. Theoretical Background

This section summarizes the main concepts related to this work. First, we describe the Top-N recommendation problem. After, we describe what are recommender systems and the two categories of algorithms that were used to design the hybrid recommender model presented in Section 4.

2.1. The Top-N Problem

The problem we aim to solve is to recommend a set of news and alerts that can be useful for a given network administrator. This problem is characterized as the Top-N recommendation problem which aims to identify a set of *N* items that will be of interest to certain user [Karypis 2001]. To avoid problems with efficiency or accuracy, *N* should be chosen carefully. If *N* is too large, an excessive amount of memory will be required to store the neighborhood lists and predicting ratings will be slow. On the other hand, selecting a too small value for N may reduce the coverage of the recommender method, which causes some items to be never recommended [Ricci et al. 2011].

2.2. Recommender Systems

A recommender system, as defined by Ricci and colleagues in their book *Recommender Systems Handbook* [Ricci et al. 2011], is a set of tools and techniques that provide suggestions for items of interest to a user. The problem of recommendation can be seen as the problem of estimating the user rating for items not yet rated by the user. Items predicted with high rating for a given user can be offered by the system as a recommendation [Glauber et al. 2013].

In this work, we are interested in the two main approaches for recommendation: Collaborative Filtering and Content-based Filtering. Hereafter, we discuss both approaches and related problems in more detail. Last, we describe hybrid recommender approaches, highlighting the mixed approach that is used in our proposed model.

Collaborative Filtering: recommends to a given user the items that other users with similar tastes liked in the past. The similarity in taste of two users is calculated based on the similarity in the rating history of the users [Feldman and Sanger 2007]. This approach is the most popular and widely implemented technique in recommender systems [Ricci et al. 2011]. An important point of this approach is that the recommendations are based on the quality of items as evaluated by peers. However, Collaborative Filtering suffers from the *cold-start* problem of handling new items or new users in the system, once there is no ratings related to neither new items or new users. Another consequent difficulty with this approach is the sparsity of the ratings space, meaning that most entries are unknown.

Content-based Filtering: the general principle of this approach is to identify the common characteristics of items that have received a favorable rating from a user u, and then recommend to u new items that share these characteristics [Feldman and Sanger 2007]. The Content-based approach suffers of limited content analysis if the system has only a limited amount of information on its users or the content of its items. This approach does not consider the quality of the items. Another important issue related to the Content-based approach is the problem to recommend items that are different but still interesting to the user.

Hybrid Approach: combines two or more recommender filtering to provide more sophisticated and complete recommendations. Burke [Burke 2007] explores seven strategies for hybrid recommendations, which were identified in his earlier survey of hybrids [Burke 2002], where he identifies possible combinations of recommendation approaches. We are mainly interested in the *mixed hybrid* approach, which presents recommendations of its different components side-by-side in a combined list. The main challenge in this type of recommender is to choose how to rank the recommended items from different approaches.

[Ricci et al. 2011] defines three main elements to be specified in a recommender model:

Items: item is an entity that is recommended in a recommender system. An item has the main attributes, in which a user is interested, and other structures that are used to rate or value them. In our problem, an item is defined as a security alert and its content elements are the main attributes.

Users: a user may have very diverse goals and characteristics whose information are used by recommendation algorithms to make new item recommendations. In our domain, the users are network administrators who are technically interested in cyber security.

Transactions: a transaction refers to a potential interaction between a user and the recommender system. The set of transactions define how items can be rated. All recommender algorithms use data collected from some transactions to perform a new recommendation.

3. Related Works

Several works have researched methods to obtain relevant security alerts to support the prevention of cyber attacks. [Apel et al. 2009] and [Flegel et al. 2010] suggested collaborative approaches based on the exchange of information among organizations to prevent and detect security threats in advance. [Grobauer et al. 2006] proposed an organizational framework to share, correlate, and analyze cyber security alerts extracted from multiple organizations. Unlike those three previous works, our research aims to evaluate a model that can be directly used by network administrators, instead of only support organizational environments.

[Bonchi et al. 2011] and [Surjandari et al. 2015] identified business opportunities in the rich content generated through collaboration between users in social networks. Santos et al. [Santos et al. 2013] also explored the potential for collaboration in social networks to extract cyber security alerts from *Twitter* that can lead network administrators to apply security policies faster. Even though Santos et al. have used heuristic techniques and unsupervised learning to properly filter cyber security messages, they evidenced the difficulty of dealing with the semantics of those messages. By applying recommender systems techniques, our work continues exploring the collaboration phenomena between users and the progress made by [Santos et al. 2013].

[Karypis 2001] discussed and compared some algorithms to solve the problem of top-*N* recommendations by conducting several offline experiments. [Ekstrand et al. 2011] and [Font et al. 2013] used and evaluated several different recommender approaches, whereas [Sarwar et al. 2001] and [Zheng and Li 2010] adapted traditional recommender methods based on collaborative filtering and content-based approaches to different practical applications. [Burke 2002] exposed some hybrid approaches to build recommender systems using more than one technique, while [Glauber et al. 2013] explored the *mixed* hybrid approach for given names, similar to the hybrid approach we adopt in our recommender model.

Our work advances the state of the art in cyber security by proposing a new model for gathering relevant information on cyber security alerts based on recommender system methods. Thus, the proposed recommender model aims to address the open issues arising from the progress made in [Santos et al. 2013] by applying collaborative techniques to properly classify cyber security alerts.

4. Recommender Model Design

This section presents the proposed recommender model for cyber security alerts and the methodology used for its design. We followed the process proposed by Ricci et al. [Ricci et al. 2011], which addresses the recommendation problem in three dimensions:

- Users: who are the users, what are their goals?
- Data: what are the characteristics of the data on which recommendations are based?
- Application: what is the application the recommender is part of?

The following subsections describe how we adopt this process. First, Section 4.1 describes how we requested and analyzed the requirements for the recommender model. After gathering enough information, we defined the data model specification, selected the proper recommender techniques and composed the mixed hybrid solution, as detailed in Section 4.2. Finally, Section 4.3 exposes the design of the running offline experiment.

4.1. Requirements Inception and Analysis

The main objective of the present work is to build a recommender model to recommend cyber security alerts, collected from external data sources. The recommender model is aimed at general users with network administrator profile, technically interested in cyber security. We intend to ensure that the model always recommend some items even if the user has never informed their preferences and even at the first interaction of the user with a system implementing the model.

Following the main definitions of entities related to a recommender system [Ricci et al. 2011], the first step to design the recommendation model is to identify some initial information about the users who will compose the model, understanding their key identifying characteristics, their skill levels, and their prior experience with similar systems. It is also important to identify the main characteristics about cyber security alerts which will match the model items and the possibilities of interactions between users and items.

For this purpose, we applied a qualitative survey for dozens of Brazilian network administrators professionals, getting 20 responses, through ESurv[1], an online web-based free tool for create surveys. This survey was designed with four parts:

- *Profile Survey*: to identify some general characteristics of potential users.
- *Cyber Security Survey*: to identify how much the users know about cyber security issues and whether they use any communication channel to be up to date about those issues.
- *Interest Assessment*: to identify requirements that must be considered into the recommender model conception. It also helps to get what are the main elements that must compose the recommender model item by asking the users what are the most important elements of a cyber security alert. Finally, this topic also provides questions that will be used to establish some interactions between users and items.
- *Use Cases*: this section was designed to simulate some use cases of an hypothetical recommender system to evidence in more detail potential users' interactions.

The survey is available online[2] as well as the results[3].

[1] http://esurv.org/ - Accessed at December 22, 2015

[2] http://bit.ly/1Rk0nR8 - Accessed at December 22, 2015

[3] https://gitlab.com/konsilo/survey/blob/master/results.pdf - Accessed at December 22, 2015

4.2. Model Specification

To build the proposed recommender model, we need to properly define the User and the Item elements. Both elements can be defined as a set of data structures that represent the real world. For both definitions, we used the applied survey described in Section 4.1.

About the User data structure, on average, the respondent network administrators have about 8 years of work experience. Most of them have a good knowledge about cyber security and often read cyber security news. However, only half of them interact with other network administrators, mainly through mailing lists and chat rooms.

A positive result we had in the survey is that all the 20 respondents are interested in using a collaboration system to get and share cyber security alerts and news. However, most of them would never provide some information as input for a recommender system like *personal name* or the *institution they work on*, as can be seen in Figure 1, which shows the percentages of respondents that selected each information they would never provide.

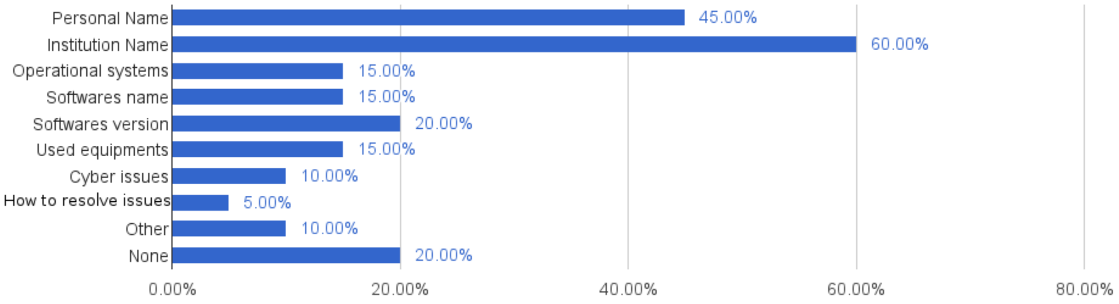

Figure 1. Information that users would never provide

Through the survey results it was also possible to notice that some types of information can be used to determine user preferences, such as operating system and type of attack related to an alert.

The recommender model items are the cyber security alerts, which are mostly available as unstructured data extracted from heterogeneous data sources. As evidenced in the survey results, the following attributes are the most important to be included into our item data structure, from the most important to the less important: (1) title, (2) information source, (3) level of critically, (4) related software/hardware, (5) keywords (tag).

Other collected data from the applied survey demonstrated how users would rate some items, as shown in Figure 2.

As can be seen in Figure 2, the most popular rating mechanism is to mark an alert as urgent or critical. Another popular rating mechanism is vote, which can be positive or negative and can be influenced by different motivations. Open comments also appears as a popular feedback input but we did not add any transaction related to it at first, mainly because it provides more subjective than objective data, tough it could be used in a real implementation of the proposed model. We also added the Categorize rating mechanism

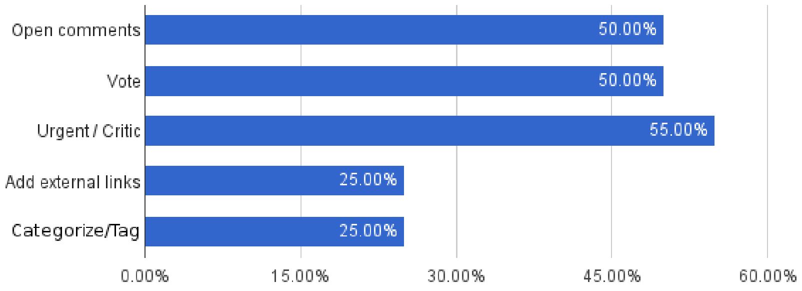

Figure 2. Rating mechanisms

to our recommender model because it can be useful to support the application of content-based filter approach, provided that it is used to define cyber security alerts' features. Therefore, the transactions of the proposed recommender model are shown in Figure 3.

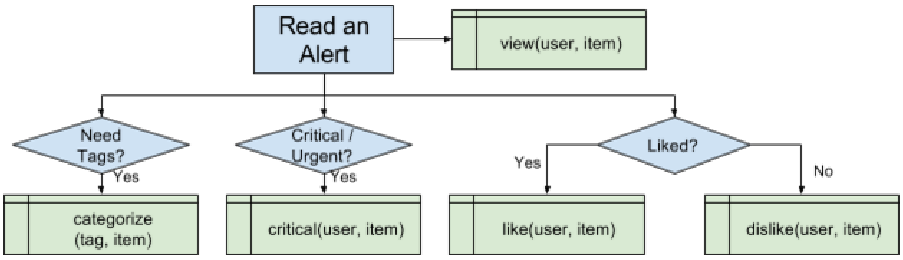

Figure 3. Recommender model transactions

Figure 3 shows the five transactions of our recommender model. They are described below:

- **view**: registers an item visualization by an user. In our model, it must be used for two purposes:
  - Estimate the most popular alerts so that it can be used in general recommendations of cyber security alerts.
  - Whenever a given user reads a security alert, the alert should not be recommended for this user again. Thus, the *view* transaction is important to remove read alerts of new recommendations.
- **categorize**: registers a new category to an alert. As many cyber security alerts are extracted from unstructured data sources, we introduced the *categorize* transaction to supply enough information about cyber security alerts in order to support the Content-based approach. For this purpose, collaborative tagging has emerged as a plausible alternative, despite the related challenges such as tag scarcity or ambiguous labelling [Font et al. 2013]. Alerts categorization can also be supported by text mining techniques by retrieving useful information from alerts that can correctly categorize an item.
- **critical**: registers a user opinion about the urgency of an alert. Although it was the most popular rating mechanism in our survey, it is only considered for general

ranking. On the other hand, it must have a higher weight than *views* and *likes* transactions for general ranking purposes.

- **like**: registers a positive relation between an user and an item. This can be interpreted as if the user showed interest or approval of the rated cyber security alert. It is important for the following purposes:
  - Estimate the most liked alerts so that it can be used in general recommendations of cyber security alerts.
  - Recommend new cyber security alerts using Collaborative Filtering, since it can be used to infer the similarity between users.
  - Recommend new cyber security alerts using Content-based approach, since it defines the main interests of an user through alerts' categories.
- **dislike**: registers a negative relation between a user and an item. This can be interpreted as if the user did not approve the rated cyber security alert content. In our model, it is important to find false or incorrect alerts to remove them. Therefore, the *dislike* transaction is neither used for the Collaborative Filtering nor the Content-based approaches, but it is useful to clean bad items from the database.

Figure 4 shows the recommender model flow, displaying the user's (*blue*) and system's (*red*) actions and how the different recommender approaches are combined to recommend cyber security alerts for network administrators.

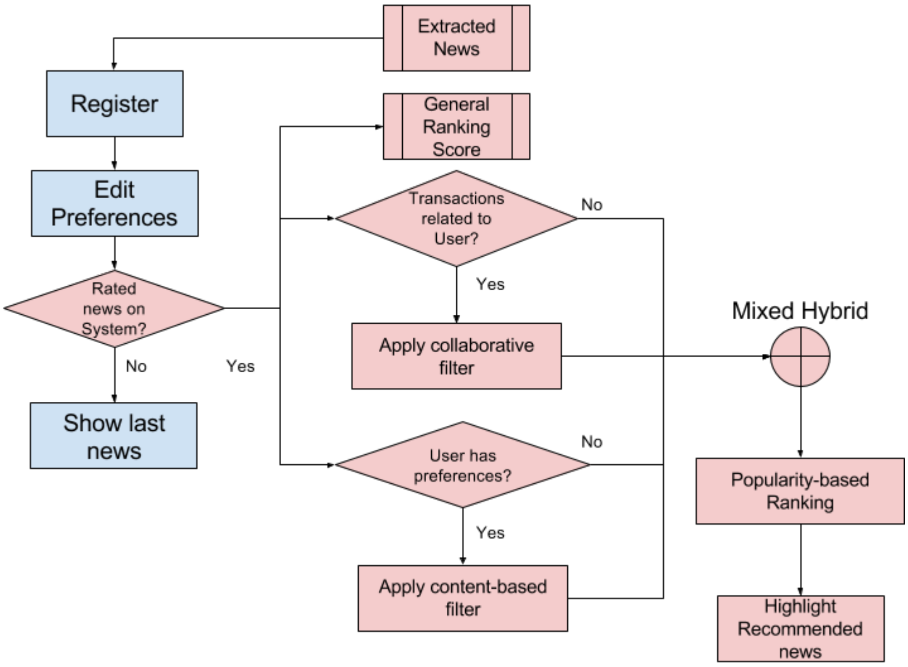

Figure 4. Recommender model main flow

As can be observed in Figure 4, since we have rated items, we must perform three computations separately, which may be done periodically: compute the general ranking score of all items, compute the collaborative filtering and the content-based approaches' recommendations. We compute the ranking score based on the Bayesian Average, as defined in Equation 1, which incorporates existing information related to our data set to minimize the impact of large deviations. The ranking score aims to define a general score to rank items, since it only considers items and ignore any data of specific users. The greater is the $ranking\_score(i)$, the better ranked an item *i* will be.

$$ranking\_score(i) = \frac{(\bar{v} \cdot \bar{r}) + (v_i \cdot r_i)}{\bar{v} + |r_i|} \tag{1}$$

where $\bar{v}$ is the average number of *votes* (likes, dislikes or critical) across all alerts; $\bar{r}$ is the average rating of all alerts; v_i is the total votes of item *i*; and r_i is the rating of the item *i*, which is defined in the Equation 2.

$$r_i = \alpha c_i + \beta l_i - \phi d_i \tag{2}$$

Equation 2 computes the rating score of an item *i* adding up the positive votes and subtracting the negative votes, where c_i is the total number of votes that classify *i* as *critical*, l_i is the total number of *likes*, and d_i is the total number of *dislikes*. It is also important to give different weights for the elements of Equation 2, as denoted by α, β and ϕ, which may vary according to the implementation of the model. Consistent with our survey results described earlier in this Subsection, we suggest the following weights showed in Equation 3, where *critical* votes have the greatest weight, followed by *dislikes* which is weighted twice larger than *likes*, as we want to remove bad alerts.

$$r_i = 4c_i + l_i - 2d_i \tag{3}$$

When there is no rated item, for instance at the initial state of the system, the model shows cyber security alerts ranked by the most recent. Otherwise, we apply the Content-based and Collaborative Filtering separately to generate two lists of recommended items.

To generate the first list with the top-*N* recommendations, we apply the Collaborative Filtering user-based approach, which is the most successful technology for building recommender systems, and is extensively used in many commercial recommender systems [Karypis 2001]. This approach aims to determine the similarity between users and recommended items to a given user that his/her fellows liked. Consequently, items frequently liked by the various members of the group can be used to form the basis of the recommended items. Thus, it is necessary to perform two steps to compute the list of top-N recommendations for a given user *u*.

First, we compute the similarity between users to find the *neighbors*, the set of users that have a history of agreeing with the target user *u*. To compute the similarity value of two users *u* and *v*, we use the Jaccard similarity coefficient defined in Equation 4, which is suitable for binary votes.

$$J(u,v) = \frac{|\ U_l \cap V_l\ |}{|\ U_l \cup V_l\ |} = \frac{|\ U_l \cap V_l\ |}{|\ U_l\ | + |\ V_l\ | - |\ U_l \cap V_l\ |} \tag{4}$$

where U_l is the set of items liked by user *u* and V_l is the set of items liked by user *v*. The $J(u,v)$ can take values between 0 and 1.

Equation 4 is defined as the ratio between the amount of items liked by both two users, and the size of the union of the items liked by each user. The greater is the Jaccard similarity coefficient for two users *u* and *v*, the greater the similarity of these two users.

Once the set of *k neighbors* of user *u*, denoted as $Neighbor(u)$, is formed, the last step is to combine the preferences of neighbors to recommend the top-*N* items for the target user *u*. For this purpose, we measure whether user *u* may like an item *r* by computing Equation 5, as Zheng & Li have done in a previous work [Zheng and Li 2010]. Since we are interested in recommending new items for a target user *u*, we do not compute Equation 5 for items that had already been viewed by *u*.

$$score(u,r) = \frac{\sum_{v \in Neighbor_{(u)}} R_{v,r} \times J(u,v)}{\mid \sum_{v \in Neighbor_{(u)}} J(u,v) \mid} \tag{5}$$

where $R_{v,r}$ is the rate of user v over the item r.

In our model, the second list is generated independently with the Content-based approach through the match of weighted tags represented by the user's interests and item's categories. Whenever a user likes an item, each one of that item's categories tags is added to the user's interests and its weight is set to 1, if it is a new tag, or incremented by 1 otherwise.

Equation 6 is an adaptation of the method presented by Zheng and Li [Zheng and Li 2010], as we added weight to tags based on user liked items, to compute content-based recommendations. It expresses our tag-weight strategy to measure how much a user *u* is interested in a given item *i*.

$$w_{tag}(u,i) = \frac{\sum\limits_{t_i \in tag(i)} weight(u,t_i)}{\sum\limits_{t_j \in tag(u)} weight(u,t_j)} \tag{6}$$

where *tag(i)* represents the set of tags which categorizes an item *i*; *tag(u)* represents the set of tags of interest of a given user *u*; $weight(u,t_i)$ denotes the weight of the tag t_i for the user u, and while $weight(u,t_j)$ denotes the weight of the tag t_j for the user u.

Equation 6 always takes a real number between 0 and 1, and the higher the weight is, the more interest a user has in the item *i*. Note that unlike our Collaborative Filtering approach, our Content-based method does not depend on other users' information to generate a top-*N* list of recommended items.

After performing the Collaborative filtering and the Content-based algorithms, we are ready to apply the mixed hybrid approach to select the top-*N* items to recommend for a target user. Note in Figure 4 that we also mix the two custom generated lists with the popularity-based ranking defined in Equation 1 to always include items considered important by the user community. Since we want to find the top-*N* items to recommend to target user *u*, we select the top ranked *N* items not viewed by *u* and compute the $mixed\_score$ of the item *i* for the user *u*, by applying Equation 7.

$$mixed\_score(u,i) = 2^{\gamma} \cdot ranking\_score(i) \tag{7}$$

where γ is two if i is recommended in both recommender approaches, one if i is only recommended by one of the approaches, and zero otherwise.

Finally, we can get the final recommendation list of items selecting the N items with highest *mixed_score*.

4.3. Evaluation

As described in [Ricci et al. 2011], to evaluate a recommender model we must identify the set of properties that may influence the success of a recommender system or model in the context of a specific application. Then, we can evaluate how the system performs on these relevant properties. In the proposal model, the *offline experiment* approach was used to evaluate the properties of the model we built.

Since there are no available data sets with the necessary cyber security alerts information, to evaluate the proposed recommender model, we used experimental data from MovieLens[4], a recommender system research website created to recommend movies based on user ratings. Although the set of rated items in this data set is not related to cyber security alerts, we can use it since it has rates and categories related to items. As our model is composed of two different recommender approaches, Collaborative Filtering and Content-based, we evaluate them separately. We started our experiment by first dividing the data set into two groups:

- **Training data**: the set of data to simulate the user behavior in our model by recording historical user data. These data corresponds to 80% of the entire data set and are used for computing new recommendations for users in *testing data*.
- **Testing data**: the set of data that is hidden from the model to be compared with the results. These data corresponds to 20% of the entire data set. Since we hide a user *u* rating for an item *i*, we gather the output of our recommender model for *u* and *i*, which can be compared through the defined metrics.

To compare different configurations of the algorithms, we varied the number of items *N* to be recommended, where $N \in \{3, 5, 10, 15\}$.

We can classify each recommendation as: *true positive* - *TP* (when an interesting item is recommended); *true negative* - *TN* (when an uninteresting item is not recommended); *false negative* - *FN* (when an interesting item is not recommended); *false positive* - *FP* (when an uninteresting item is recommended) [Cremonesi et al. 2008]. Since both the approaches Collaborative Filtering and Content-based are used to generate a list of *N* recommendations, we evaluate the *precision* and *recall* of them, as explained below:

- **Precision**: measure the ability to recommend *only* what is relevant to the user. The precision is computed as expressed in Equation 8.

$$precision = \frac{TP}{TP + FP} \tag{8}$$

- **Recall**: measure the ability to recommend *everything* that is relevant to the user. The recall is computed as expressed in Equation 9.

$$recall = \frac{TP}{TP + FN} \tag{9}$$

[4] http://grouplens.org/datasets/movielens/ - Accessed at December 22, 2015

Both metrics take values between 0.0 and 1.0, where the higher the value, the better the algorithm's performance in the measured property.

5. Results Analysis

The experiment was lead on a data set with 207 users, 1287 items and a total of 11088 training ratings. To run the experiment, we developed Konsilo, a web application that implements the proposed recommender model to recommend cyber security alerts extracted from social medias and specialized sites. Konsilo's code is available under the *AGPL V3* license in *https://gitlab.com/konsilo/konsilo* .

Figures 5 and 6 show the comparison between the collaborative filtering and content-based approaches. As mentioned before, we evaluate the precision and recall over different configurations of the *top-N* problem.

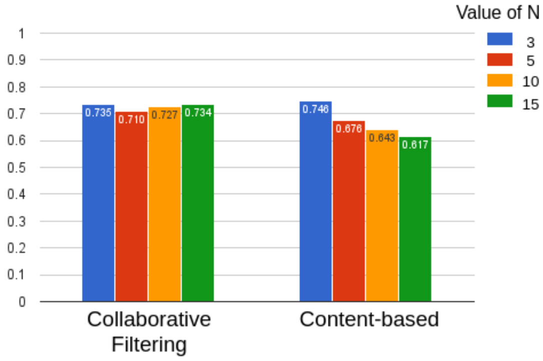

Figure 5. Precision comparison

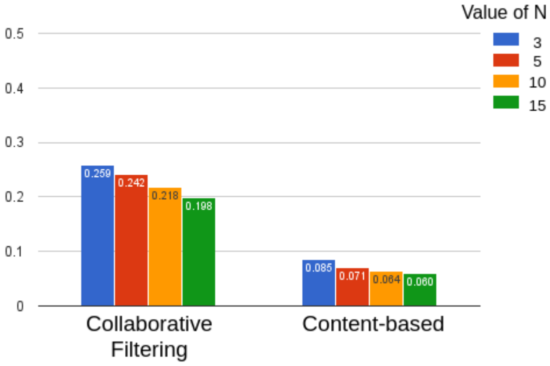

Figure 6. Recall comparison

As can be seen in Figure 5, for different values of *N*, the collaborative filtering had a good precision with little variance. On the other hand, for the content-based approach, the larger the value of *N*, the lower the precision. The best performance was obtained with N equal to 3, but it is important to notice that there is not a large drop in precision for higher values of *N*.

Figure 6 shows the comparison of the recall value between the two recommender approaches. We also measure the recall over different values of *N*. For all configurations, both approaches had low performance related to recall. This may be due the large amount of available items that could be recommended in our experiment, since similar results were found in other *top-N* recommendation experiments [Cremonesi et al. 2008] [Herlocker et al. 2004]. However, it is observable the superiority of collaborative filtering over content-based approach. It is also important to notice that the recall value decreases when increasing the value of *N*.

6. Conclusion and Future Directions

This work advances the state of the art in cyber security by proposing a new model for gathering relevant information on cyber security alerts. To the best of our knowledge, there are no similar works proposed on the literature. This work also shows that recommender system methods and techniques can be explored in the cyber security field to improve access to relevant information for network administrators.

The proposed recommender model is mainly based on two recommender approaches which generate separated lists of recommendations that can be mixed to produce better results and reduce the common problems inherent in them. The model also includes elements of ranking score and other features considered important for network administrators.

It is important to notice that the proposed model is not coupled to technologies, so it can be implemented by any system focused on cyber security alerts. In this work, we developed the web application Konsilo as the first implementation case of the proposed recommender model. Furthermore, Konsilo was used to support the offline experiment to evaluate the proposed collaborative filtering and content-based recommender approaches. In general terms, both approaches demonstrated satisfactory precision for the *top-N* recommendation problem, but poor performance to recall all the interesting items for users.

Future work can be performed to overcome the existing limitations in the present work. We plan to use the GT-EWS[5] project to enable the recommendation of alerts and filtering false positives. We also want to release Konsilo in a production environment as a technological contribution, which also will allow us to carry out new experiments over the recommender model through the behavior of actual users and better evaluate the proposed model as a whole. Both Konsilo and GT-EWS can be used to gather enough information about cyber security alerts and their ratings so that it can be used to build a new data set for future works.

7. Acknowledgments

We thank Fundacão Araucária, Secretaria do Estado da Ciência, Tecnologia e Ensino Superior (SETI-PR), Governo do Estado do Paraná and Rede Nacional de Pesquisa (RNP) for supporting this research and our participation in the event.

References

Apel, M., Biskup, J., Flegel, U., and Meier, M. (2009). Towards early warning systems - challenges, technologies and architecture. In *InCRITIS*, pages 151–164.

Bonchi, F., Castillo, C., Gionis, A., and Jaimes, A. (2011). Social network analysis and mining for business applications. 2:22:1–22:37.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. In *UMUAI 12*, pages 331–370.

Burke, R. (2007). Hybrid web recommender systems. In *The Adaptive Web*, pages 377–408.

Cremonesi, P., Turrin, R., Lentini, E., and Matteucci, M. (2008). An evaluation methodology for collaborative recommender systems. In *Proceedings of the International Conference on Automated solutions for Cross Media Content and Multi-channel Distribution*, pages 224–231.

Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). Collaborative filtering recommender systems. In *Foundations and Trends in Human–Computer Interaction*, volume 4, pages 81–173.

[5] http://gtews.ime.usp.br - Accessed at December 22, 2015

Feldman, R. and Sanger, J. (2007). *he Text Mining Handbook: Advances Approaches in Analyzing Unstructured Data*. Cambridge University Press.

Flegel, U., Hoffmann, J., and Meier, M. (2010). Cooperation enablement for centralisticearly warning systems. In *InProceedings of the ACM SAC*, pages 2001–2008.

Font, F., Serra, J., and Serra, X. (2013). Folksonomy-based tag recommendation for collaborative tagging systems. In *International Journal on Semantic Web and Information Systems*, volume 9, pages 1–30.

Frei, S., May, M., Fiedler, U., and Plattner, B. (2006). Large-scale vulnerability analysis. In *SIGCOMM Workshop on LSAD*, volume 6, pages 131–138.

GCHQ and Cert-UK (2015a). Common cyber attacks: Reducing the impact. Technical report, The Information Security Arm of GCHQ & Cert-UK.

GCHQ and Cert-UK (2015b). Internet security threat report - 2014 trends. Technical report, Symantec.

Glauber, R., Loula, A., and Rocha-Junior, J. B. (2013). A mixed hybrid recommender system for given names. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 25–36.

Grobauer, B., Mehlau, J. I., and Sander, J. (2006). Carmentis: A co-operative approachtowards situation awareness and early warning for the internet. In *InIMF*, pages 55–66.

Herlocker, J., Konstan, J., Terveen, L., and Riedl, J. (2004). Evaluating collaborative filtering recommender systems. In *ACM Transactions on Information Systems (TOIS)*, pages 5–53.

Karypis, G. (2001). Evaluation of item-based top-n recommendation algorithms. In *10th Conference of Information and Knowledge Management*, pages 247–254.

Krsul, I. V. (1998). *Software Vulnerability Analysis*. PhD thesis, Purdue University, West Lafayette.

Ricci, F., Rokach, L., Shapira, B., and Kanto, P. B. (2011). *Recommender Systems Handbook*. Springer.

Santos, L. A. F., R., C., Gerosa, M. A., and Batista, D. M. (2013). Detecção de alertas de segurança em redes de computadores usando redes sociais. In *31.o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 791–803.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *10th international conference on World Wide Web*, pages 285–295.

Surjandari, I., Naffisah, M. S., and Prawiradinata, M. I. (2015). Text mining of twitter data for public sentiment analysis of staple foods price changes. In *Journal of Industrial and Intelligent Information*, volume 3, pages 253–257.

Zheng, N. and Li, Q. (2010). A recommender system based on tag and time information for social tagging systems. In *Expert Systems with Applications*, pages 4575–4587.