CyberRank- Knowledge Elicitation for Risk Assessment of Database Security

Hagit Grushka-Cohen,
Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev
hgrushka@post.bgu.ac.il

Oded Sofer, Ofer Biller
IBM Security Division, IBM Cyber Security Center of Excellence, Beer Sheva
{odedso, ofer.biller}@il.ibm.com

Bracha Shapira, Lior Rokach
Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev
{bshapira,liorrk}@bgu.ac.il

ABSTRACT
Security systems for databases produce numerous alerts about anomalous activities and policy rule violations. Prioritizing these alerts will help security personnel focus their efforts on the most urgent alerts. Currently, this is done manually by security experts that rank the alerts or define static risk scoring rules. Existing solutions are expensive, consume valuable expert time, and do not dynamically adapt to changes in policy.
Adopting a learning approach for ranking alerts is complex due to the efforts required by security experts to initially train such a model. The more features used, the more accurate the model is likely to be, but this will require the collection of a greater amount of user feedback and prolong the calibration process. In this paper, we propose CyberRank, a novel algorithm for automatic preference elicitation that is effective for situations with limited experts' time and outperforms other algorithms for initial training of the system. We generate synthetic examples and annotate them using a model produced by Analytic Hierarchical Processing (AHP) to bootstrap a preference learning algorithm. We evaluate different approaches with a new dataset of expert ranked pairs of database transactions, in terms of their risk to the organization. We evaluated using manual risk assessments of transaction pairs, CyberRank outperforms all other methods for cold start scenario with error reduction of 20%.

Keywords
Risk assessment; Preference elicitation; Ranking; Cold Start; Semi Supervised; Cyber Security

1. INTRODUCTION
Security information and event management (SIEM) systems are widely used by organizations to implement security policies and detect attacks and data abuse. Known security risks include abuse of an organization's sensitive information and violation of personal privacy. Various solutions based on anomaly detection identify risks such as data leakage, data misuse, and attacks in database systems [1,2,3,4,5,6]. These systems produce alerts when policy rules are violated or anomalous activities are performed [3,6]. Each alert demands the attention of a security officer [3] who must decide whether an alert represents a risk which should be investigated or dismissed. Some alerts, such as those that pertain to sensitive data, are more urgent than others. Security officers are inundated with data due to the large volume of alerts, and they could easily miss an urgent alert or only become aware of an important alert after it is too late, while handling alerts which pose less risk to the organization. Moreover, an excessive number of irrelevant alerts can cause the user to lose confidence in the security system and abandon it. This challenge is not unique to cyber security systems and represents a hurdle for the adoption of AI alert systems in other environments such as healthcare [8].

Our objective is to develop a prioritization method that automatically rank alerts by the risk a transaction poses, enabling security experts to focus their time and efforts on the most important alerts. Some studies [1,9] have tried to tackle this problem by elevating the threshold of anomaly detection algorithms, however this has resulted in lower recall thereby missing important alerts.

The determination of the level of risk associated with an alert (or activity) is based on the security officer's (SO) knowledge and understanding of each individual case presented to her. The alert is associated with metadata that is used by the SO to determine the risk. Such metadata features may include: User Group, Database Accessed or User OS Vulnerabilities (has the user installed all latest security patches). The definition of risk varies depending on the domain and organization. Given this, most currently used systems must first learn organizational preferences by interrogating the system officer. For example, an organization may prefer to focus their security efforts on protecting trade secrets in their engineering database while being less concerned with the integrity of their corporate wiki.
The security domain is very dynamic: attacks evolve over time, with more sophisticated attacks being created all of the time, and new rules and regulations are established (e.g., regulations related to maintaining customers' privacy). As a result, the risk associated with an activity may change accordingly.
The state-of-the-art approaches for ranking use learning algorithms such as recommender systems to learn user's preferences [10]. Supervised learning algorithms such as Ranking SVM [11,12] or Learning to Rank [13,14] require datasets with annotated examples in order to learn user preferences. Our problem presents a constant cold start situation as features and preferences vary between organizations, and only domain experts are able to accurately annotate the data; therefore, the creation of such datasets is expensive, a fact which often prevents organizations from adopting such systems.
In addition, even when organizations are willing to invest expert time, most existing methods for ranking alerts generate a long list of customized rules to weigh features, or they use preference elicitation heuristics such as analytic hierarchical processing (AHP) that transform questionnaire answers to features weights. Being

CIKM'16, October 24–28, 2016, Indianapolis, IN, USA.
 ISBN 978-1-4503-4073-1/16/10…$15.00.
DOI: http://dx.doi.org/10.1145/ 2983323.2983896

rule-based these solutions are static and do not improve with users' feedback; thus they cannot effectively handle the dynamic nature of the risks.

In this paper, we present CyberRank, a novel algorithm for bootstrapping cold start ranking with preference elicitation in the domain of the security of database transactions. It is a method for automatic preference elicitation that is effective in situations with little data availability or requiring minimal user feedback. We generate synthetic examples and annotate them using a model produced by the AHP heuristic, and we use this dataset to bootstrap a preference learning algorithm. Over time the model can be retrained by combining the synthetic data with feedback data gathered from the user. This enables the learning model to provide meaningful results with little or no training examples and keep improving as feedback is gathered. CyberRank, the proposed algorithm provides several advantages: (i) it provides useful results out of the box, (ii) it improves over time, (iii) it is explainable as the SVM model provides linear weights that can be explained to the user, and (iv) can be calibrated by users as needed (e.g., upon policy change).

2. RELATED WORK

Estimating the riskiness of objects can be accomplished by mimicking the ways in which experts rank activities or by defining policy rules. Customized rules are static and their creation is time consuming, in contrast to learning algorithms for ranking which can model the expert knowledge dynamically.

2.1 Ranking Algorithms

Ranking algorithms serve as the base of recommender systems which are often used for document retrieval in a variety of settings [10, 15]. The ranking function assigns a score to each of the ranked entities. The ranking order represents the entities' preference with respect to the question asked. There are two main approaches for the task of ranking (or two main types of ranking algorithms): the pairwise approach and the listwise approach.

2.1.1 Pairwise approach

This approach formulizes the problem as a classification problem by collecting pairs of instances, for example, database (DB) activities, and assigning a label representing the relative riskiness of the two activities to the pair. It then trains a supervised classification model such as ranking SVM [11,12] RankBoost, and RankNet [16]. For this, we form the difference of all comparable elements such that our data is transformed into $(x'_k, y'_k) = \left(x_i - x_j, sign(y_i - y_j)\right)$ for all comparable pairs. The task is to separate positive samples from negative ones.

2.1.2 Listwise approach

Instead of using object pairs as instances, this approach uses a list of objects as instances in learning and trains a learning function through the minimization of a listwise loss function defined on the predicted list and the ground truth list. Listwise approaches include RankCosine, ListNet, or ListMLE algorithms [13,14].

2.2 AHP

Analytic hierarchy process (AHP) [17] is a heuristic for quickly eliciting preferences, which requires the expert to answer questions about pairs of features and their values (e.g., *"What is more risky: (a) accessing a sensitive object or (b) a user with system vulnerability?"*). In our setting, the result is a weighted model that can be used to grade DB records based on the given value of each feature. In the case of [18], AHP scores were found to be correlated to user perception of risk.

The AHP method has the advantage that it provides measures of judgement consistency, meaning if value *a* is better than value *b,* and value *b* is better than value *c*, then value *a* should be better than value *c*. This allows estimating how consistent the rankings are.

However, this is a static solution that does not enable ongoing learning and therefore needs to be repeated when additional policies are implemented or new hazards are identified.

2.3 Semi-Supervised Learning for Ranking

In the semi-supervised approach, the lack of annotated training data is addressed by leveraging unannotated examples [15]. Zhang and Ben He [19] created pseudo labels for learning a ranking model in the domain of information retrieval (IR). They chose training examples from query results which were easily divided into two clusters, based on the level of relevance, and labeled the documents in the more relevant cluster as positive.

2.4 Alerting Methods in the Security Domain

Several methods, such as [2,3,4], have been proposed and applied for the detection of data leakage, data misuse, and attacks in database systems. The main approach focuses on using machine learning, mostly unsupervised, to detect anomalous patterns or outliers [3,5]. The focus is on reducing false positive rates due to class imbalance (most activities are legitimate). In this work we are not attempting to detect misuse or anomalous activities but to prioritize the alerts produced by anomaly detection systems.

3. Ranking by Risk

We examined three approaches for prioritization: a baseline approach for preference elicitation using AHP, a supervised pairwise learning approach, and a hybrid approach (described below) which bootstraps the learning algorithm using the preference elicitation model.

3.1 CyberRank Hybrid Approach

Our objective is to develop a system that allows easy initial calibration in order to circumvent the cold start (similar to AHP), while being capable of learning from feedback. We present CyberRank, a method for bootstrapping learning algorithms for ranking based on preference elicitation.

CyberRank consists of two stages: setup and learning (see Figure 1). The setup stage includes preference elicitation with AHP and the generation of synthetic training data annotated with the AHP model. During the learning stage a supervised model is trained from the data, and this model is used for ranking alerts. The user can provide feedback as well, and when feedback is provided, the model is retrained on a training set comprised of both the synthetic and feedback data.

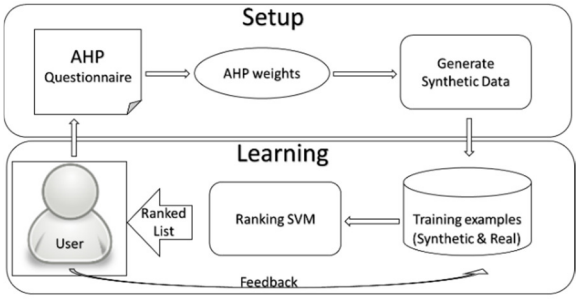

Figure 1 CyberRank Process Breakdown: Setup Stage and Learning Stage

3.1.1 Preference Elicitation with AHP

The SO identifies relevant features for the ranking task. For each pair of features, the SO answers which feature is more risky (see

section 2.2). The answers are then combined using the AHP heuristic to provide a weighted model for producing a score between 0 and 1 for each transaction.

3.1.2 Generating a Synthetic training set

Synthetic data is generated by sampling values for each feature from a uniform distribution. The result is a vector of categorical values for each feature. Each vector is scored using the weighted model produced in 3.1.1. The score represents the transaction risk according to the questionnaire model (a higher score suggests higher risk).

Pairs of synthetic data points are chosen for training pairwise ranking. In order to make the samples more separable (provide the model with clear cut training examples) we choose pairs with AHP scores difference higher than 0.3 (this difference was chosen empirically based on the pair distance distribution).

3.1.3 Training a Ranking SVM Model

Using the synthetic data generated in step 3.1.2 we train a two class linear SVM model for pairs of transactions.

3.1.4 Incorporating New Training Data (user feedback)

As user feedback is gathered, the new data points are added to the synthetic dataset for training, and the model is retrained. To compensate for the imbalance due to initially small amount of data gathered, we are oversampling the real user generated data. Real transactions are given higher weight during training so that the model assigns 50% of the decision according to the synthetic samples and 50% according to the synthetic dataset (when enough user feedback is gathered the synthetic model can be discarded).

4. EXPERIMENTAL SETTING

The experiments are designed to evaluate CyberRank, based on the accuracy of the pairwise ranking as a function of the size of the training sets. We compared CyberRank with other methods in terms of the number of training examples required to reach a satisfactory level of accuracy. The following is a description of the dataset collected, the compared methods, and the results.

4.1 Datasets

Our dataset is made of two types of data: synthetic annotated examples and manually annotated examples provided by security experts.

4.1.1 SO annotated dataset

The user annotation is based on categorized data within each feature: users were better able to understand and consider categorized data rather than numerical values. For example, referring to the user's IP as external or internal is more informative than the IP itself. Therefore, when constructing the examples to be annotated by expert users we categorized all of the values.

We randomly created possible scenarios using combinations of possible categories for each feature. Each vector was scored using the weighted model created with the AHP (based on the questionnaire filled by the SO).

The dataset annotated by the SO was comprised as follows: one third of the pairs had a low difference in AHP score, another third of the pairs were significantly different based on AHP, and the remaining third fell in between in this regard.

We gathered 170 pairs of examples. For each pair, the SO indicated which transaction is riskier (first or second).

4.1.2 Synthetically annotated data set:

We generated 100 pairs with scores with an AHP score difference higher than 0.3 as described in section 3.1.2. The labels for these pairs were determined using the AHP weighted model. Each pair was represented by the deduction of the values of the features of the second example from the values of the features of the first example. The labels were the sign of the deduction of the AHP scores.

4.2 Experiments

We conducted experiments to test the performance of our approach compared to four models, all of which are briefly introduced below:
1) Pure AHP – defined the risk score based on the AHP weights, no training was involved (simply applying to the test dataset).

2) Vanilla Ranking SVM – trained only on the SO annotated samples.

3) CyberRank no oversampling – Ranking SVM trained on both: (i) user samples and (ii) synthetic data produced with the AHP model (from step 1). No oversampling as described in 3.1.4.

4) CyberRank without pair distance criteria – using Ranking SVM to train the model over randomly chosen synthetic pairs. (we do not require a difference in AHP score as described in 3.1.2).

5) CyberRank – trained on a weighted combination of user generated data and synthetic data produced with the AHP model.

We compared the models' performances for different sized SO annotated datasets. We started with 10 examples and increased the number of samples by five each time, up to 80 samples. Given that we have only 12 features, we were able to apply the models over 10 samples annotated by end users. For less than 10 samples we could not apply model 2 as the model failed to converge.

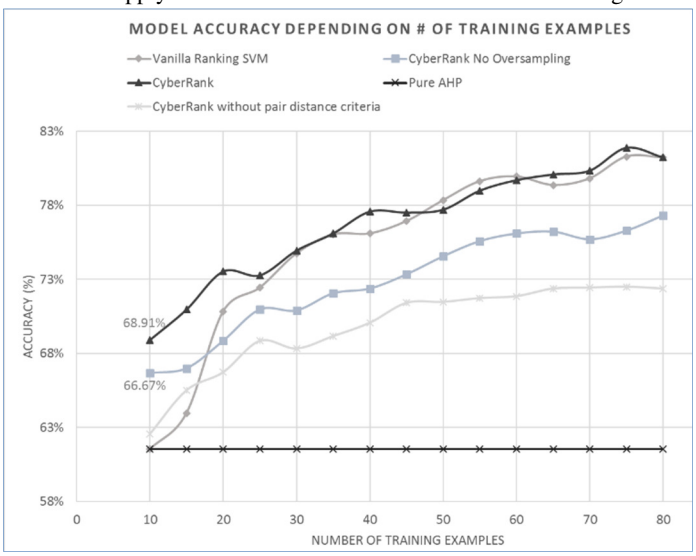

Figure 2. Accuracy of models as a function of the number of training examples

4.3 Evaluation Metric

We conducted 30-fold cross-validation experiments on training datasets, and evaluated the accuracy of our model over the testing dataset, made of 30 percent of the original annotated by SO dataset.

5. Results

5.1 Ranking Accuracy

Figure 2 summarizes the performance of all compared models: AHP, Vanilla Ranking SVM, Ranking SVM with Synthetic Data,

and CyberRank. As demonstrated from the results, CyberRank approach outperforms baseline learning algorithms in the cold start scenario: trained only on synthetic data it achieves 61.5% accuracy and 71% accuracy when trained on 15 samples, significantly surpassing the 64% achieved by Ranking SVM on the same sample size (error reduction of 20%). We performed a paired t-test for sample sizes of 10 and 15 examples and found that the means of the Vanilla Ranking SVM and CyberRank are significantly different at $p < 0.05$.

At no point did the Ranking SVM model outperform CyberRank, even when ample data was available: with 70-80 annotated examples both models achieved ~81% accuracy. For a small number of training examples (10 and 20) there was high standard deviation in the accuracy of the Vanilla learning approach (0.12 and 0.11, see Table 1), suggesting high dependency on the examples sampled. Using CyberRank the standard deviation is much lower, reducing that dependency.

The CyberRank variants: (i) no oversampling, and (ii) no distance criteria for synthetic data, both underperformed compared to CyberRank and Ranking SVM.

6. Conclusions

In this paper, we presented and analyzed CyberRank, a novel algorithm for bootstrapping cold start ranking with preference elicitation in the domain of the security of database transactions.

Our proposed algorithm, CyberRank, enables bootstrapping a pairwise preference learning algorithm, such as Ranking SVM, using synthetic examples annotated with a model created using AHP. CyberRank overcomes the need for a minimal number of samples when using supervised learning with a large number of features. It shortens the time before the model becomes useful in cold start scenarios. Our experiments show that the Vanilla Ranking SVM model does not outperform CyberRank at any point. Moreover, CyberRank may be modified to replace the AHP step by annotating synthetic examples using existing policy weights rules, which is useful if a customized model is already available.

We conclude from the experiment that AHP provides a non-trivial baseline for capturing expert knowledge but not accurately enough alone. For generating synthetic data to train on, AHP performed well when using the distance criteria for choosing pairs – when the score difference in AHP was small the pairs confused the Ranking SVM yielding significantly worse results. During the human tagging phase, the security experts expressed a similar intuition that deciding on similar pairs was harder. Oversampling the human annotated examples provided faster improvement of the model, this is especially important in cold start when gathering examples is the most expensive.

The underlying preference model (AHP) is flexible, and in the case of policy changes or changes in the security landscape the preference questionnaire answers may be changed and new synthetic data introduced to the training for reflecting the change. For example, if it is discovered that there are many cyber-attacks from a specific country, the SO can remake the AHP model and produce synthetic samples according to the new preferences. Our proposed approach is not limited to Ranking SVM – other algorithms may be plugged in (we had similar results using decision trees).

Although CyberRank algorithmic framework was originally designed for the domain of the security of database transactions preference learning, it could be used in solving general cold start ranking problems.

All the data and code discussed in the paper are made available[1].

[1] https://github.com/hagitGC/CyberRank

7. Acknowledgments

This work has been partially supported by the IBM Cyber Security Center of Excellence, Beer Sheva and by IBM Security Division. Many thanks to Dario Kramer and Rosa Miroshnikov of IBM USA and IBM Canada for their fruitful contribution to the experiments process.

8. REFERENCES

[1] Sallam, A., Bertino, E., Hussain, S.R., Landers, D., Lefler, R.M. and Steiner, D., 2015. DBSAFE—An Anomaly Detection System to Protect Databases From Exfiltration Attempts. *IEEE SYSTEMS JOURNAL.*

[2] Kim, G., Lee, S. and Kim, S., 2014. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, 41(4), pp.1690-1700.

[3] Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection: A survey. In *ACM computing surveys* (CSUR), 41(3), p.15.

[4] Veeramachaneni, K. and Arnaldo, I., AI2: Training a big data machine to defend.

[5] Hodge, V.J. and Austin, J., 2004. A survey of outlier detection methodologies.In *Artificial Intelligence Review*, 22(2), pp.85-126.

[6] Niu, Z., Shi, S., Sun, J. and He, X., 2011. A survey of outlier detection methodologies and their applications. In *Artificial intelligence and computational intelligence* (pp. 380-387). Springer Berlin Heidelberg.

[7] Costante, E., Vavilis, S., Etalle, S., den Hartog, J., Petkovic, M. and Zannone, N., 2013, July. Database anomalous activities detection and quantification. In *Security and Cryptography (SECRYPT), 2013 International Conference* (pp. 1-6). IEEE.

[8] Pecht, M. and Jaai, R., 2010. A prognostics and health management roadmap for information and electronics-rich systems. *Microelectronics Reliability,* 50(3), pp.317-323.

[9] Geng, J., Ye, D., Luo, P. and Lv, P., 2015. A Novel Clustering Algorithm for Database Anomaly Detection. In *Security and Privacy in Communication Networks* (pp. 682-696). Springer International Publishing

[10] Ricci, F., Rokach, L. and Shapira, B., 2011. *Introduction to recommender systems handbook* (pp. 1-35). Springer US

[11] Joachims, T., 2002, July. Optimizing search engines using clickthrough data. In *SIGKDD* (pp. 133-142). ACM.

[12] Zhang, Y., Xu, J., Lan, Y., Guo, J., Xie, M., Huang, Y. and Cheng, X., 2015, October. Modeling Parameter Interactions in Ranking SVM. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (pp. 1799-1802). ACM.

[13] Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F. and Li, H., 2007, June. Learning to rank: from pairwise approach to listwise approach. In *ICML* (pp. 129-136). ACM.

[14] Xia, F., Liu, T.Y., Wang, J., Zhang, W. and Li, H., 2008, July. Listwise approach to learning to rank: theory and algorithm. In *ICML* (pp. 1192-1199). ACM.

[15] Liu, T.Y.,2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3), pp.225-331

[16] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N. and Hullender, G., 2005, August. Learning to rank using gradient descent. *ICML* (pp. 89-96). ACM.

[17] Saaty, T.L., 2008. Decision making with the analytic hierarchy process. International journal of services sciences, 1(1), pp.83-98.

[18] Harel, A., Shabtai, A., Rokach, L. and Elovici, Y., 2012. M-score: A misuseability weight measure. *Dependable and Secure Computing, IEEE Transactions* on, 9(3), pp.414-428.

[19] Zhang, X., He, B., Luo, T., Li, D. and Xu, J., 2013, October. Clustering-based transduction for learning a ranking model with limited human labels. CIKM (pp. 1777-1782).