Predicting User Roles from Computer Logs using Recurrent Neural Networks

Aaron Tuor[*] and **Samuel Kaplan** and **Brian Hutchinson**
Western Washington University
Bellingham, WA

Nicole Nichols and **Sean Robinson**
Pacific Northwest National Laboratory
Seattle, WA

Introduction

Network and other computer administrators typically have access to a rich set of logs tracking actions by users. However, they often lack metadata such as user role, age, and gender that can provide valuable context for users' actions. Inferring user attributes automatically has wide ranging implications; among others, for customization (anticipating user needs and priorities), for managing resources (anticipating demand) and for security (interpreting anomalous behavior).

User attribute profiling has been most widely explored in the context of social media, using lexical and social graph data. Ikeda et al. (2013) use a hybrid system which predicts user demographic information by a clustering of the user's followers/followees in conjunction with a support vector machine which analyzes the tweet history of a user. Oentaryo et al. (2016) use a generalization of logistic regression with social network relationship data in order to predict user attributes such as age, sex, and religion.

Other researchers have considered user attribute profiling in network and cybersecurity domains. Iglesias (2012) presents an unsupervised rule based classifier that profiles users into learned and adaptive categories based on sequences of Unix shell commands. Meng et al. (2008) learn a mixture model over behavior templates for sequences of network activity. Panda and Patra (2008) explore the effectiveness of several decision tree classifiers and Naive Bayes at classifying a network intrusion from sequences of network events. Udoeyop (2010) uses kmeans and kernel density estimation to detect normal and abnormal user profiles from fine-grained user network activities, including processes, process times, and file reads, writes and opens.

In the ongoing work described in this abstract, we predict the role of users from computer logs, although our method trivially generalizes to predict other categorical attributes. We use recurrent neural networks to make increasingly refined predictions over time, achieving a 30% relative reduction in role classification error over our baseline.

[*]Email: tuora@wwu.edu. Phone: 360-650-6134. Address: 516 High Street, Bellingham, WA 98229.

Model

Our model takes as input a series of T feature vectors $\mathbf{x}_1^u, \mathbf{x}_2^u, \ldots, \mathbf{x}_T^u$ (one vector per day) for a user u and produces as output a series of T probability vectors $\mathbf{p}_1^u, \mathbf{p}_2^u, \ldots, \mathbf{p}_T^u$, giving the distribution over roles for this user. Because we use recurrent neural networks to produce the probabilities, $\mathbf{p}_t^u$ is a function of all feature vectors up to and including time t. In contrast to a model where $\mathbf{p}_t^u$ depends only on $\mathbf{x}_t^u$, this approach offers two major advantages: first, it allows us to capture any temporal patterns that may exist in user behavior across days, and second, it allows us to build confidence in our predictions having accumulated evidence over a series of days.

Specifically, our model is a stacked LSTM (Hochreiter and Schmidhuber 1997). Let $\mathbf{h}_t^u$ and $\mathbf{c}_t^u$ denote the hidden state and (long-term memory) cell state, respectively. For an LSTM with C classes and a single hidden layer, $\mathbf{p}_t^u$ is calculated as follows:

$$p_{t,k}^u = \frac{e^{h_{t,k}^u}}{\sum_{j=1}^{C} e^{h_{t,j}^u}}, \text{ where} \quad (1)$$

$$\mathbf{h}_t^u = \mathbf{o}_t^u \odot \tanh(\mathbf{c}_t^u) \quad (2)$$

$$\mathbf{c}_t^u = \mathbf{f}_t^u \odot \mathbf{c}_{t-1}^u + \mathbf{i}_t^u \odot \mathbf{g}_t^u, \text{ and} \quad (3)$$

$$\mathbf{f}_t^u = \sigma\left(\mathbf{W}_{f,x}\mathbf{x}_t^u + \mathbf{W}_{f,h}\mathbf{h}_{t-1}^u + \mathbf{b}_f\right) \quad (4)$$

$$\mathbf{i}_t^u = \sigma\left(\mathbf{W}_{i,x}\mathbf{x}_t^u + \mathbf{W}_{i,h}\mathbf{h}_{t-1}^u + \mathbf{b}_i\right) \quad (5)$$

$$\mathbf{o}_t^u = \sigma\left(\mathbf{W}_{o,x}\mathbf{x}_t^u + \mathbf{W}_{o,h}\mathbf{h}_{t-1}^u + \mathbf{b}_o\right) \quad (6)$$

$$\mathbf{g}_t^u = \tanh\left(\mathbf{W}_{g,x}\mathbf{x}_t^u + \mathbf{W}_{g,h}\mathbf{h}_{t-1}^u + \mathbf{b}_g\right) \quad (7)$$

Here $\odot$ denotes element-wise multiplication, and σ denotes the logistic sigmoid function. The model parameters are the eight weight matrices $\mathbf{W}$ and the four bias vectors $\mathbf{b}$, and are shared among all users. For a stacked LSTM with more than one hidden layer between $\mathbf{x}_t^u$ and $\mathbf{p}_t^u$, the inputs to the ℓth layer are the hidden states of the $(\ell-1)$th layer.

In order to accommodate the streaming nature of the network user role classification task, we train on multiple user action sequences concurrently. This is accomplished by saving LSTM hidden and cell states for a fixed number of time steps for each user of the network.

Experimental Setup

To assess our model's effectiveness we conducted a set of role classification experiments on the CERT Insider Threat v6.2 dataset (Glasser and Lindauer 2013). The data consists of log lines collected from a sophisticated simulation of user behavior in an organization's computer network. Log lines are produced from five different sources: user logon/logoff behavior, web traffic, file reads/writes/accesses, email traffic, and external device usage. 135,117,169 log lines in total are generated from this simulated activity of 4000 users over the course of 516 days. Under the assumption that 90 days is a sufficient duration to identify user roles, we use only the first 90 days worth of data. For each user for each day, we derive a 408-dimensional feature vector capturing the user's behavior with respect to these five sources (e.g. one feature is the # emails with attachments the user sent between 12am-6am). User role is provided in the dataset and serves as our prediction label. Roles with very few representatives (under 10) are merged into a single catch-all role, leaving 33 distinct user roles including the catch-all. We split the data by user into training (80%), development (10%) and test (10%) sets.

Random search is used to tune several hyper-parameters on the development set, including the number of layers in the stacked LSTM, the dimensionality of the hidden and cell state vectors and the learning rate. Our model is implemented using the Tensorflow toolkit.

Results

We evaluate our model in terms of cross entropy, which is the negative log probability that our model assigns to the true role. Fig. 1 shows our prediction cross entropy as a function of time. As expected, performance starts out poor (high cross entropy) but steadily improves as more observations are fed into the model. Predictions continue to improve until approximately day 40, after which performance levels out. After 90 days of observations, our model achieves an accuracy of 38%, significantly better than the 11% baseline of predicting the majority class role.

Conclusions

User attribute profiling in a streaming network environment is an important and challenging problem; in this abstract we propose a deep learning approach. Our preliminary results are encouraging, finding that the model shows steady improvements over time, ultimately achieving an accuracy of 38% on a 33-way role classification task.

There are several ways that we plan to extend this ongoing work. First, we would like to explore a richer set of features that may be more discriminative for user role. We also plan to address the role imbalance by resampling/replay learning and improve generalization and/or accelerate training using techniques such as dropout, and layer normalization. Finally, we plan to evaluate our model on additional datasets.

Acknowledgments. The research described in this paper was conducted in part under the Laboratory Directed Research and Development Program at PNNL, a multi-program national laboratory operated by Battelle for the U.S. Department of Energy, and supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Visiting Faculty Program (VFP).

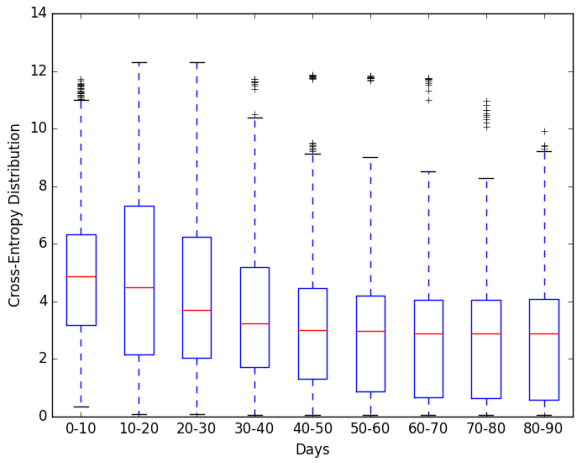

Figure 1: Performance as a function of time.

References

[Glasser and Lindauer 2013] Glasser, J., and Lindauer, B. 2013. Bridging the gap: A pragmatic approach to generating insider threat data. In *Proc. SPW*.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

[Iglesias et al. 2012] Iglesias, J. A.; Angelov, P.; Ledezma, A.; and Sanchis, A. 2012. Creating evolving user behavior profiles automatically. *IEEE Trans. KDE* 24(5):854–867.

[Ikeda et al. 2013] Ikeda, K.; Hattori, G.; Ono, C.; Asoh, H.; and Higashino, T. 2013. Twitter user profiling based on text and community mining for market analysis. *Knowledge-Based Systems* 51:35–47.

[Meng et al. 2008] Meng, X.; Jiang, G.; Zhang, H.; Chen, H.; and Yoshihira, K. 2008. Automatic profiling of network event sequences: algorithm and applications. In *Proc. INFOCOM*. IEEE.

[Oentaryo et al. 2016] Oentaryo, R. J.; Lim, E.-P.; Chua, F. C. T.; Low, J.-W.; and Lo, D. 2016. Collective semi-supervised learning for user profiling in social media. *arXiv preprint arXiv:1606.07707*.

[Panda and Patra 2008] Panda, M., and Patra, M. R. 2008. A comparative study of data mining algorithms for network intrusion detection. In *Proc. Int. Conf. on Emerging Trends in Engineering and Tech.*, 504–507.

[Udoeyop 2010] Udoeyop, A. W. 2010. Cyber profiling for insider threat detection. Master's thesis, University of Tennessee.