Functional Cognitive Models of Malware Identification

Christian Lebiere, Stefano Bennati, Robert Thomson ({cl, sbennati, thomsonr}@andrew.cmu.edu)
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213 USA

Paulo Shakarian, Eric Nunes ({shak, eric.nunes}@asu.edu)
Arizona State University
699 S. Mill Avenue, Tempe, AZ 85281 USA

Abstract

An important source of constraints on unified theories of cognition is their ability to perform complex tasks that are challenging for humans. Malware reverse-engineering is an important type of analysis in the domain of cyber-security. Rapidly identifying the tasks that a piece of malware is designed to perform is an important part of reverse engineering that is manually performed in practice as it relies heavily on human intuition. We present an automated approach to malware task identification using two different approaches using ACT-R cognitive models. Against a real-world malware dataset, these cognitive models significantly out-perform baseline approaches while demonstrating key cognitive characteristics such as the ability to generalize to new categories and to quickly adapt to a change of environment. Finally, we discuss the implications of our approach for applying cognitive models to complex tasks.

Keywords: functional cognitive models, ACT-R, Bayesian models, decision trees, malware detection.

Introduction

Malware reverse-engineering is an important type of analysis in the domain of cyber-security. Rapidly identifying the tasks that a piece of malware is designed to perform is an important part of reverse engineering and is manually performed in practice as it relies heavily on human intuition (Sikorski & Honig, 2012). The difficulty of this task increases substantially when historically studied malware samples are significantly different (i.e. members of a different malware family). Cognitive architectures such as ACT-R (Anderson, Bothell, Byrne, et al., 2004) have previously been shown to effectively model human cognition on a variety of decision-making (Lebiere, Gonzalez, & Martin, 2007) and general intelligences tasks (Lebiere, Gonzalez, & Warwick 2009), including complex domains such as intelligence analysis (Lebiere, Pirolli, Thomson, et al., 2013). Further, they have been shown to perform well on reasoning tasks where historical knowledge is sparse, limited, or dissimilar to the current context (Taatgen, Lebiere, & Anderson, 2006). However, models have occasionally had to abstract from some of the details of the high-fidelity framework that cannot be constrained by data in order to scale to complex tasks involving substantial human expertise (e.g., Sanner et al, 2000). Our work fits into that approach by selectively using some features of the cognitive architecture while temporarily ignoring others.

Malware Identification

In this paper, we leverage such models to identify the tasks associated with a piece of malware. Using a real-world malware dataset (Mandiant Corp, 2013), these cognitive models identify sets of tasks with an unbiased F1 measure of 0.94 – significantly out-performing baseline approaches. Even when trained on historical datasets of malware samples from different families, our ACT-R cognitive models still maintain the precision of baseline methods while providing a significant improvement to recall by identifying over 60% of malware tasks.

Existing work on malware classification falls into two general categories: (1) determining if a given binary is malicious (Tamersoy, Roundy & Horng 2014; Firdausi, Lim Erwin, & Nugroho, 2010) and (2) classifying malware by family (Bayer, Comparette, Hlauschek, et al., 2011; Kinable & Kostakis, 2011; Kong & Yan, 2013). The problem of identifying whether a binary is malware is complementary to this effort (a "first step") – as an analyst must first identify malware before then determining what it does. Our work substantially differs from malware family classification as we look to directly infer the tasks that a malware was created to perform whereas malware family classification is mainly used to help guide an analyst into identifying tasks by first identifying a family. It is noteworthy that we were able to train our classifiers on data of malware of *different* families than the malware we are attempting to classify and were still able to obtain a set of tasks with over 60% recall on the best-performing cognitive models. Further, malware family classification has suffered from two primary draw-backs: (1) disagreement about malware family "ground truth" as different analysts (i.e. Symantec and MacAfee) cluster malware into families differently; and (2) previous work has shown that some of these approaches mainly succeed in "easy to classify" samples (Perdisci, 2012; Li, Liu, Gai & Reiter, 2010) – where an "easy to classify" family is a family that is typically agreed upon by multiple malware analysis firms. By inferring malware tasks directly, we avoid both of these pitfalls. Further, as a side-effect, we create a probability distribution over malware families as part of an intermediate step – though the ultimate inference of malware tasks is independent of *how* the historical malware families are classified by family.

ACT-R Models

The models are built using the mechanisms of the ACT-R cognitive architecture and learn to recognize malware samples based upon a limited training schedule similar to the actual experience of a human analyst. Given a malware sample, the model generates a probability distribution over a set of malware families then infers a set of likely malware intents based upon that distribution. The models primarily leverage the subsymbolic (statistical) mechanisms of the ACT-R architecture, especially the activation calculus underlying retrieval from long-term declarative memory. Each sample is represented by its set of static and dynamic attributes. The model operates in two stages: first by family, then by intent. To assign family, the model generates a probability distribution over the set of possible malware families from the activation in declarative memory of the chunks representing those families. To assign intents in a second pass, the model combines the probability distribution over families with a representation linking each malware family to known intents. Two distinct models were created that leveraged separate parts of the activation calculus.

ACT-R Rule-Based Model

This ACT-R model is based on the Bayesian components of the activation calculus, specifically the base-level and spreading activation components. Given a malware training sample with its set of attributes, along with the ground truth family, we derive a pair of conditional probabilities $p(a/f)$ and $p(a/\neg f)$ for attribute a belonging (or not) to family f. Those probabilities are used to set the strengths of association from each attribute a to each family chunk f. Similarly, Bayesian priors $p(f)$ are used to set the base-level of each family. Given the attributes of the current malware held in the goal buffer context, a retrieval for family chunks (the "rules", not to be confused with production rules) computes their activation and sets the probability of each family according to the Boltzmann (softmax) equation. Intents are then determined by summing up the probability of the families associated with a given intent, with an appropriately set threshold (50%).

ACT-R instance-Based Model

This model follows the instance-based learning theory (IBL; Gonzalez, Lerch, and Lebiere, 2003) that is particularly relevant to modeling naturalistic decision making in complex dynamic situations. The instance-based approach is an iterative learning method that reflects the cognitive process of accumulating experiences and using them to make decisions. In this case a chunk is created for each malware instance associating the set of attributes of that malware with its family. When a new malware is encountered, a retrieval for past chunk instances is triggered with the purpose of inferring their family. The retrieval primarily uses the base-level and partial matching components of the activation equation. The base-level reflects the recency and frequency of each instance according to the power law of learning and decay, while the similarity measure used in partial matching is computed as the overlap (dot product) between the attribute vector of the current malware and each sample in memory. A probability distribution over families is generated by the blending mechanism that sums up the evidence supporting each family from the individual instance chunks (Lebiere, 1999; Wallach & Lebiere, 2003). The same process is used for generating intent judgments, this time partial matching the family probability distribution of this malware instance against those of past instances. The intent chunks that reach the activation threshold are given as answers.

Experiment

We created a dataset from 132 malware samples used by the APT1 cyber espionage group as identified by the popular report by Mandiant Inc (Mandiant, 2013). Dynamic malware analysis was performed using the ANUBIS sandbox which generates an XML-formatted report for each malware. From the XML data, a total of 1740 malware attributes were identified (see Table 1).

Table 1: Sample attributes from Anubis malware sandbox

ATTRIBUTES	INTUITION
hasDynAttrib	Malware has a generic attribute determined in the analysis
usesDll(***X***)	Malware uses a library ***X***
regAct	Malware conducts an activity in the registry
fileAct	Malware conducts an activity on a certain file
proAct	Malware initiates or terminates a process

Each malware sample belonged to one of 15 families (e.g., BISCUIT). Based on malware family description, we associated a set of tasks with each family that each malware in that family was designed to perform. In total, 30 malware tasks were identified for the given malwares (see Table 2). On average, each family performed 9 tasks.

Table 2. Sample of malware tasks.

TASK	INTUITION
beacon	Beacons back to the adversary's system
enumFiles	Designed to enumerate files on the target
ServieManip	Manipulates services running on the target
takeScreenShots	Takes screen shots
upload	Designed to upload files from the target

Decision Tree

We implemented a decision tree as a baseline approach. This hierarchical algorithm is widely used for classification problems (Alpaydin, 2007). We used information gain to find the best split at a node. The gain was calculated using malware attributes. In order to avoid over-fitting, the

terminating criteria was set to less than 5% of total samples. Note that labels are not used for terminating the tree, hence the leaf nodes may or may not be pure, generating a probability distribution over the malware families.

Results

We compared the decision tree (DT) approach to implementations of the rule-based and instance-based ACT-R models (ACTR-R and ACTR-IB respectively). Precision, recall and F1 values were computed for the inferred adversarial tasks. On average, each sample was associated with 9 tasks out of 30 different tasks in total. DT predicted 9 tasks per sample, ACTR-R 9 tasks, and ACTR-IB 10 tasks.

Leave One Out Cross-Validation (LOOCV)

In leave one out cross validation, for N malware samples, we train on N-1 samples and test on the remaining one. This procedure was repeated procedure for all samples and the results were averaged (see Figure 1).

ACTR-IB outperformed both the DT and ACTR-R models; average F1 = 0.94 vs .81 ($t(132) = 5.77$, $p = 5e\text{-}8$), and .82 ($t(132) = 5.35$, $p=3.83\text{-}7$) respectively. The Bayesian nature of the ACT-R model dominates because it is trained on a stable, almost complete set of statistics. The IBL model is superior because it uses the full pattern of the probability distribution over families rather than just a sum.

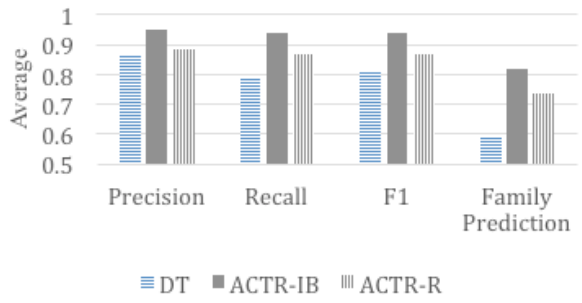


Figure 1. Average Precision, Recall, F1 and Family prediction comparisons for DT, ACTR-IB and ACTR-R.

These three approaches were also evaluated with respect to predicting the correct family (before the tasks were determined). Both the ACTR-IB and ACTR-R cognitive models outperform DT to predict the correct malware family. ACTR-IB has an average family prediction accuracy of 0.82, outperforming the DT model's accuracy of 0.6, $t(132) = 5.35$, $p = 3.8e\text{-}7$. ACTR-R also outperformed with prediction accuracy of 0.72 vs 0.6, $t(132) = 3.23$, $p = 1e\text{-}3$. Figure 2 (below) shows family-wise performance for LOOCV. This gives an unbiased estimation regarding predictions for different malware families, giving insight as to which families are difficult to predict.

ACTR-IB outperforms DT in 9 out of 15 malware families with an average F1 difference of 0.3 with at least 99% confidence, $t(132) = 4$, $p = 0.01$. DT performs qualitatively better than ACTR-IB in 4 out of 15 malware families with an average F1 difference of 0.05, but this difference is not statistically significant, $t(132) = 0.76$, $p = 0.49$. Similarly, ACTR-R outperforms DT in 7 out of 15 malware families with an average F1 difference of 0.27, while DT does not perform significantly better than ACTR-R, $t(132) = 0.28$, $p = 0.786$.

Among the cognitive models ACTR-IB performs qualitatively better than ACTR-R in 12 out of 15 families with average F1 difference of 0.08, but this difference is not statically significant, $t(132)= 1.58$, $p = 0.19$.

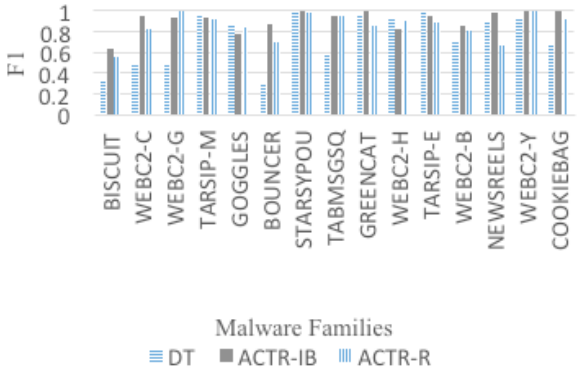
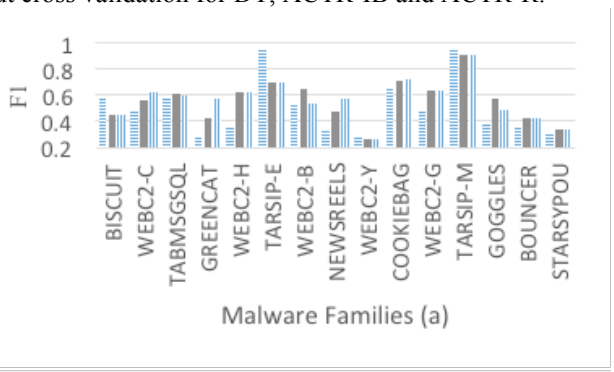


Figure 2. F1 measure by malware families for leave one out cross validation for DT, ACTR-IB and ACTR-R.



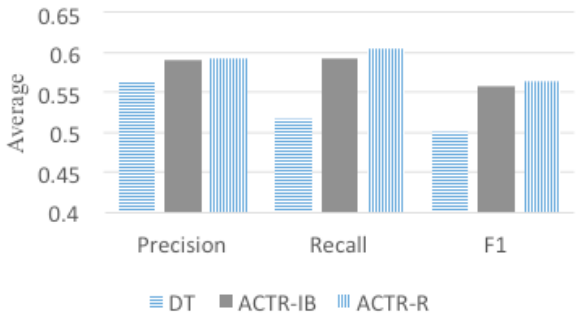


Figure 3. Average F1 values for 15 malware families (a) and the average precision, recall and F1 across all families (above) for DT, ACTR-IB and ACTR-R.

Leave One Family Out Cross-Validation

To see how the models generalize to unseen malware families, we performed a leave-one-family-out comparison,

where we test the models against one previously unseen malware family. Both the ACTR models significantly outperform the decision tree in terms of precision, recall and F1 (see Figure 3). This is primarily due to the statistical nature of the classification performed by the cognitive models over the logical classification of the decision tree.

Despite the overall higher performance for the cognitive models over the decision-tree, there were certain families where the decision tree performed particularly well when compared to the cognitive model. In the F1 comparison the decision tree peaks for TARSIP-Eclipse and TARSIP-Moon. Both these families are variants of the same malware profile. TARSIP-Eclipse performs 12 tasks, while TARSIP-Moon performs 13. They have 12 tasks in common hence during testing one family gets incorrectly predicted as the other while still getting almost all their tasks correct.

90/10 Training/Testing

Finally, we randomly divided the data into 90% training and 10% testing. This measure was then divided into 10 phases, where in the first phase the models were trained with 10% of the total training data (which was 90% of the dataset) and then an additional 10% of the training data was added for each subsequent phase. Note that each phase gets tested on the same test data. This allows us to observe the performance of the decision tree and cognitive models for incremental learning across the training data. In the real world, humans need to be able to learn from small partial samples and adapt quickly to changes in the underlying distribution. As shown in Figure 4, it is clear that the ACT-R models outperform the decision tree in precision, recall and F1 measures. This is particularly true of the instance-based model, which uses the dynamic nature of the blending mechanism to generalize over the entire space from just a few instances.

An important point to note is that the cognitive models achieve the best performance against the decision tree with only 40% of the training data. T-tests were computed for each fraction of training data comparing each of the ACT-IB, ACTR-R, and DT models against each other. The results of both ACTR models statistically outperformed decision tree (all $p < .001$) except for when 30% of the training data was used ($p = 0.46$). We hypothesized that our random sample for the 30% training data phase may have underrepresented the population of malware samples where the decision tree performs poorly. By examining the family-wise performance for leave one out cross validation (see Figure 2) we determined that decision tree has difficulty predicting malware tasks from families BISCUIT, WEBC2-CSON, WEBC2-GREENCAT, TABMSGSQL, COOKIEBAG and NEWSREELS (difference in F1 measure is greater than 0.3 as when compared to ACTR-IB or ACTR-R). The overall fraction of malware samples belonging to these families is 0.36 and in all phases, except it is 0.31 in the 30% phase, thus relatively increasing the performance of decision tree at that point.

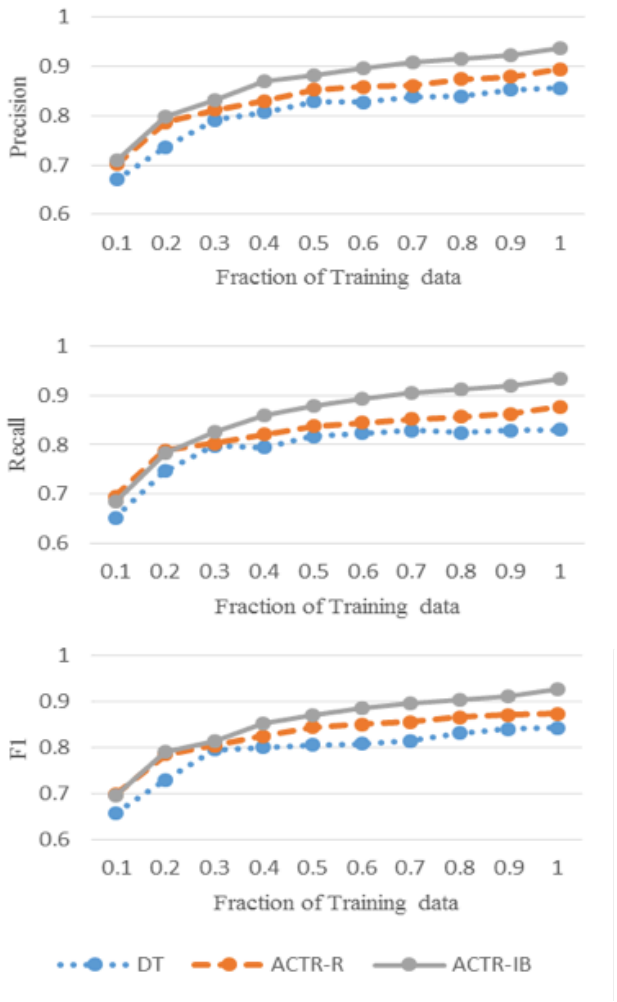


Figure 4. Average Precision, Recall and F1 for fraction of training data of 200 trials for DT, ACTR-IB and ACTR-R.

Discussion

These ACT-R models are not full-fledged high-fidelity models in that, while they make sole use of cognitive mechanisms, they do not use all aspects of the architecture, nor are they directly compared to human data. The primary reasons for this approach are three-fold: (1) because of the challenging nature of the task, we decided to focus on the functional aspects of the model; (2) we did not believe that the unmodeled aspects of the task would significantly impact the performance of the model; (3) we did not have human performance data with which to assess the model.

Regarding (1), we believe that there is a valid use of cognitive architectures for artificial intelligence that makes use of basic cognitive mechanisms while not necessarily making use of all constraints of the architecture. In that case, the model has to be evaluated on functional grounds, which is the approach that we took. However, we also discuss in the concluding section which aspects of the model were currently not cognitively plausible, such as the lack of working memory constraints, and how they could be remedied, perhaps by improving current deficiencies of the

architecture. In general, artificial intelligence constraints, such as high performance on complex tasks, can serve a valuable purpose in driving the development of cognitive architectures. Conversely, constraints on unified theories of cognition can be used to design more useful benchmark tests of artificial intelligence (Lebiere et al., 2015).

Regarding (2), Reitter & Lebiere (2010) introduced a modeling methodology called accountable modeling that recognizes that not every aspect of a cognitive model is reflected in measurable performance, and thus that human performance data cannot constrain all aspects of a model. In that case, it is arguably better to specifically state which aspects of the model are not constrained by data, and rather than mock up those aspects in plausible but impossible to validate manner, simply treat them as unmodeled processes. This approach results in simpler models with a clear link between mechanisms used and results accounted for, rather than being obscured by complex but irrelevant machinery.

Regarding (3), we are exploring obtaining human data through empirical studies using expert malware analysts to provide the kind of data that can be directly compared against performance provided by a full ACT-R model.

Conclusion

We present two cognitive models of malware intent classification. Those models are both based on the ACT-R cognitive architecture but leverage separate mechanisms and have distinct advantages. The rule-based model leverages the Bayesian memory activation mechanisms. The representation is more compact, with a single memory chunk for each family whose associations abstract the various instances belonging to that category, but those associations need to be computed and do not involve time discounting and other adaptive features (Thomson & Lebiere, 2013). The instance-based model is based on a more direct, incremental learning that accumulates malware instances in long-term memory and leverages neurally plausible pattern matching processes such as partial matching and blending (Lebiere et al., 2013) but is less parsimonious with storage and thus has potential scalability issues for large data sets.

A number of further model developments can address those and other issues. The first computational efficiency issue is the size of the feature set, which can easily number in the hundreds for a given malware. It is also an issue of cognitive plausibility since feature set size is associated to working memory, usually assumed in humans to be about seven or so (Miller, 1956). Reducing feature set size could also potentially improve generalization by removing features that are only misleadingly associated with specific intents and focusing on those that are causally related to malware function. One potential approach is to choose among features those that most contribute to correct performance. This can be implemented in ACT-R by relying on the production utility reinforcement learning mechanism to sequentially select specific features (Rutledge-Taylor et al., 2011). Given the limited working memory constraint in the form of a fixed spreading activation parameter, this can benefit performance both in eliminating spurious features and focusing limited attentional resources on the most diagnostic features.

Another approach to reducing feature set size is to build higher order features with which to represent malware instances. This process is similar to the concept of chunking in expertise-driven domains such as chess playing (Chase & Simon, 1973). When those higher-order features are known, they can be directly incorporated in the model and have been shown to improve learning performance by orders of magnitude (Sanner et al., 2000). Alternatively, a deep learning algorithm could be used to infer those features in a manner similar to past efforts combining ACT-R with neural learning mechanisms (e.g., Jilk et al., 2008, Vinokurov et al, 2011), illustrating the benefits of combining symbolic and neural architectures.

A second model development to improve computational efficiency for the instance-based learning model would be to reduce the size of the instance set in long-term memory. One possibility is to reinforce the most similar malware chunk(s) already in memory instead of creating a new one, which has already been shown to preserve generalization while sharply reducing memory requirements and retrieval process demands (Sanner et al, 2000). This approach results in the emergence of prototypes that can be seen as a middle ground between the single-chunk representation of categories in the rule-based model and the pure instance-based approach of the IBL mode.

Another approach to reducing the size of the feature set would be to include an ontology of malware functionality (e.g., Mateos et al., 2012) that would allow the model to reason over the association of features and intents and prune the representation (e.g., Oltramari et al., 2014). This process of combining symbolic reasoning to guide statistical learning is one of the main advantages of integrated cognitive architectures.

Acknowledgements

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of the Interior (DOI) contract number D10PC20021. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained hereon are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI, or the U.S. Government.

References

Alpaydin, E. 2007. Introduction to Machine Learning. *Massachusetts Institute of Technology*.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. 2004. An integrated theory of mind. Psychological Review, 11(4), 1036-1060.

Bayer, U., Comparetti, P.M., Hlauschek, C., Krügel, C., Kirda, E. 2009. Scalable, behavior- based malware clustering. In *NDSS*.

Chase, W. G., & Simon, H. A. 1973. Perception in Chess. *Cognitive Psychology*, 4, 55-61.

Firdausi, I; Lim, C.; Erwin, A; & Nugroho, AS. 2010. Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection. *In Proceedings of Second Annual Conference of Advances in Computing, Control and Telecommunication Technologies*. 201-203.

Gonzalez, C., Lerch, J. F., & Lebiere, C. 2003. Instance-based learning in dynamic decision making. *Cognitive Science, 27*, 591-635.

Jilk, D. J., Lebiere, C., O'Reilly, R. C., & Anderson, J. R. 2008. SAL: An explicitly pluralistic cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 20(3), 197-218.

Kinable, J., Kostakis, O. 2011. Malware classification based on call graph clustering. *J. Comput. Virol*. 7(4), 233–245. DOI 10.1007/s11416-011-0151-y.

Kong, D., & Yan, G. 2013. Discriminant malware distance learning on structural information for automated malware classification. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining,* 1357–1365. ACM, New York, NY, USA. DOI 10.1145/2487575.2488219.

Lebiere, C. 1999. The dynamics of cognition: An ACT-R model of cognitive arithmetic. *Kognitionswissenschaft.*, 8 (1), 5-19.

Lebiere, C., Gonzalez, C., & Martin, M. 2007. Instance-based decision making model of repeated binary choice. In *Proceedings of the 8th International Conference on Cognitive Modeling*. Ann Arbor, Michigan, USA.

Lebiere, C., Gonzalez, C., & Warwick, W. 2009. A Comparative Approach to Understanding General Intelligence: Predicting Cognitive Performance in an Open-ended Dynamic Task. In Proceedings of the Second Artificial General Intelligence Conference (AGI-09). Amsterdam-Paris: Atlantis Press.

Lebiere, C., Pirolli, P., Thomson, R., Paik, J., Rutledge-Taylor, M., Staszewski, J., & Anderson, J. R. 2013. A functional model of sensemaking in a neurocognitive architecture. *Computational Intelligence & Neuroscience.*

Lebiere, C, Bothell, D., Morrison, D., Oltramari, A., Martin, M., Romero, O., Thomson, R., & Vinokurov, J. (2015). Strong Cogsci: Guidance from cognitive science on the design of a test of Artificial Intelligence. Proceedings of the *Beyond the Turing Test Workshop, AAAI-2015*.

Li, P., Liu, L., Gao, D., & Reiter, M. K. 2010. On Challenges in Evaluating Malware Clustering. *Proceedings of the 13th International Symposium on Recent Advances in Intrusion Detection*, Ottawa, Canada.

Mandiant. 2013. APT1: Exposing One of China's Cyber Espionage Units. Mandiant Corp. URL: http://intelreport.mandiant.com/ retrieved 1/21/2014.

Mateos, V., Villagrá, V. A., Romero, F., & Berrocal, J. 2012. Definition of response metrics for an ontology-based Automated Intrusion Response Systems. *Computers & Electrical Engineering,* 38(5), 1102-1114.

Miller, G. A. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review,* 63 (2), 81–97.

Oltramari, A., Vinokurov, Y., Lebiere, C., Oh, J., & Stentz, A. 2014. Ontology-Based Cognitive System for Contextual Reasoning in Robot Architectures. *In 2014 AAAI Spring Symposium Series.*

Perdisci, P., & ManChon, U. 2012. VAMO: Towards a Fully Automated Malware Clustering Validity Analysis. In *Proceedings of the 28th Annual Computer Security Applications Conference.*

Reitter, D., & Lebiere, C. (2010). Accountable Modeling in ACT-UP, a Scalable, Rapid-Prototyping ACT-R Implementation. In *Proceedings of the 2010 International Conference on Cognitive Modeling.*

Rutledge-Taylor, M., Lebiere, C., Vinokurov, Y., Staszewski, J., & Anderson, J. R. 2011. Bridging the gap: A neurally plausible functional model of sensemaking. In Proceedings of *Biologically Inspired Cognitive Architectures,* 331-340.

Sanner, S., Anderson, J. R., Lebiere, C., & Lovett, M. 2000. Achieving efficient and cognitively plausible learning in backgammon. *In Proceedings of the Seventeenth International Conference on Machine Learning,* 823-830. San Francisco: Morgan Kaufmann.

Sikorski, M., Honig, A. 2012. Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software, 1st edn. No Starch Press, San Francisco, CA, USA.

Taatgen, N., Lebiere, C. & Anderson, J.R. 2006. Modeling paradigms in ACT-R. In Sun, R. (Ed) *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation.* NY, NY: Cambridge University Press.

Tamersoy, A., Roundy, K. A., & Horng, D. P. 2014. Guilt By Association: Large Scale Malware Detection by Mining File-Relation Graphs". In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2014.* New York City, NY.

Thomson, R. & Lebiere, C. 2013. Constraining Bayesian Inference with Cognitive Architectures: An Updated Associative Learning Mechanism in ACT-R. In *Proceedings of the 35th Annual Conference of the Cognitive Science Society*. Berlin, Germany.

Vinokurov, Y., Lebiere, C., Herd, S. A., & O'Reilly, R. C. 2011. A Metacognitive Classifier Using a Hybrid ACT-R/Leabra Architecture. *Lifelong Learning AAAI Workshop*.

Wallach, D. & Lebiere, C. 2003. Conscious and unconscious knowledge: Mapping to the symbolic and subsymbolic levels of a hybrid architecture. In Jimenez, L. (Ed*.) Attention and Implicit Learning.* Amsterdam, Netherlands: John Benjamins Publishing Company.