Achieving Active Cybersecurity through Agent-Based Cognitive Models for Detection and Defense

Robert Thomson[1][0000-0001-9298-2870] , Edward A. Cranford[2][0000-0002-6081-4570], and Christian Lebiere[2][0000-0003-4865-3062]

[1] United States Military Academy, West Point, NY 10996, USA
[2] Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, 15213, USA
robert.thomson@westpoint.edu; {cranford, cl}@cmu.edu

Abstract. We propose a methodology for the development of autonomous intelligent cyber-defense agents based on cognitive models. Those cognitive models inherit both mechanism and limitations from cognitive architectures implementing unified theories of human cognition. The mechanisms endow the models with powerful characteristics of human cognition, including robustness, generalization and adaptivity. The limitations enable the models to predict the cognitive biases of human teammates and adversaries, allowing them to augment the former and exploit the latter. This paper provides an introduction to the cognitive mechanisms used, in particular the subsymbolic activation mechanisms underlying symbolic knowledge representation enabling human-like learning and adaptivity. We illustrate the approach with a number of applications, including models of sensemaking in geospatial intelligence, deceptive signaling for cyber defense, and malware and intrusion detection systems.

Keywords: Intrusion Detection Systems, Cognitive Modeling.

1 Introduction

This article presents an overview describing how cognitive models have been applied for both understanding and augmenting human analyst performance and as autonomous agents in cybersecurity applications. Using the ACT-R cognitive architecture as a unifying theory, we present a framework for understanding how to leverage human-inspired heuristic reasoning to process large amounts of information, well beyond human capacity and without the limitations of human cognitive biases (Thomson, Lebiere, Anderson, & Staszewski, 2014). ACT-R accurately models human cognition in a variety of decision-making (Lebiere, Gonzalez, & Martin, 2007; Erev et al, 2010) and general intelligences tasks (Lebiere, Gonzalez, & Warwick 2009), as well as in complex domains such as intelligence analysis (Lebiere, et al., 2013). These models have also performed well on reasoning tasks where historical knowledge is sparse, limited, or dissimilar to the current context. To scale to complex tasks involving substantial human expertise (e.g., Sanner et al, 2000), models can abstract from the high-fidelity framework aspects of the task that cannot be constrained by data and are not directly related to performance (Reitter & Lebiere, 2010).

ACT-R provides an experimentally validated general model for human cognition as a hybrid symbolic/sub-symbolic architecture (Laird, Lebiere & Rosenbloom, 2017),

which provides a clearer distinction between automatic and implicit (System 1) processes from deliberative and explicit (System 2) processes (Kahneman, 2011). While traditional dichotomies argue that System 1 processes are based on *procedural knowledge* and System 2 processes on *declarative knowledge,* ACT-R instead shows that operations over sub-symbolic elements (based on activation dynamics) can be seen as System 1, while operations over symbolic elements can be seen as System 2 (Thomson et al., 2014). Instance-based learning (*IBL*; Gonzalez, Lerch, & Lebiere, 2003) is a commonly used modeling approach, which argues that expertise is gained through the accumulation and recognition of experienced events (i.e., *instances*), which form the primary basis for decision-making. The dynamics of an instance's sub-symbolic activations determine which instances are likely to be most relevant for a given situation and can also explain why they were selected and what factors came into play. Supporting this structure, Lebiere, Gonzalez, and Warwick (2009) have shown how recognition-primed decision-making and IBL have similar mechanisms and predictions in naturalistic decision-making.

In the present review we wish to focus on two features of cognitive models: (1) how cognitive models can be used to understand and augment human analysts, including the source and remediation of biases as well as an in-the-loop aid for personalized modeling; and (2) as autonomous agents in the context of cybersecurity tasks including malware identification and instruction detection. In embedded applications, autonomous agents may need to interact with the adversary. We argue that cognitive models are ideally suited for this kind of interaction as they are able to simulate the underlying cognitive processes of the adversary (even if the adversary is an algorithm; Somers et al., 2020) as well as those of a human teammate.

2 What is a Cognitive Model?

A cognitive model is an introspectable abstraction of human reasoning processes. When embodied in a *cognitive architecture* (such as ACT-R), it provides an empirically validated and falsifiable algorithm for understanding human decision-making. ACT-R is a hybrid architecture using both symbolic information and sub-symbolic processes operating over these symbolic elements. Symbolic information structures provide the knowledge representation and reasoning characteristic of expert human performance, while sub-symbolic statistical processes provide the adaptivity and robustness of everyday common sense activities. ACT-R is formally broken up into two main modules: *declarative* knowledge and *procedural* knowledge. Declarative knowledge is knowledge of facts and what is traditionally thought-of as human memory, while procedural knowledge is knowledge of the skills and abilities that generally utilize declarative facts to make decisions. For the purpose of this paper, we focus on declarative knowledge, but for a fuller discussion see Thomson et al. (2014).

Declarative knowledge is represented formally in terms of chunks. Chunks have an explicit type and consist of an ordered list of slot-value pairs of information. Chunks are retrieved from declarative memory by an activation process:

$$P_i = (e^{A_i/s}) \,/\, (\Sigma_i e^{A_j/s}) \quad (1)$$

where P_i is the probability that chunk i will be recalled, A_i is the activation strength of chunk i, $\sum A_j$ is the activation strength of all of eligible chunks j, and s is momentary noise inducing stochasticity by simulating background neural activation. The activation of a given chunk i (A_i) is governed by its summed base-level activation (B_i) reflecting its recency and frequency of occurrence, partial matching score (P_i) reflecting the degree to which the chunk matches the retrieval request, and finally a noise value ($\mathcal{E}_i$) including both transient and permanent noise:

$$A_i = B_i + P_i + \epsilon_i \tag{2}$$

Sub-symbolic activations approximate Bayesian inference by framing activation as log-likelihoods, with base-level activation (B_i) as the prior, the sum of spreading activation and partial matching (P_i) as the likelihood adjustment factor(s), and the final chunk activation (A_i) as the posterior.

A chunk's base-level activation is computed by summing across the number of presentations n for chunk i the log of the time t_j since the jth presentation discounted by the decay rate d, with an optional constant β_i added to this value:

$$B_i = ln\left(\Sigma_{j=1}^{n} t_i^{-d}\right) + \beta_i \tag{3}$$

Base-level activation corresponds to the Bayesian prior of a chunk's activation. A benefit of base-level activation is that it provides an automated procedure for frequency-based strengthening as well as temporal discounting. Chunks are also compared to the desired retrieval pattern using a partial matching mechanism (P_i) that subtracts from the activation of a chunk i its degree of mismatch M_{ki} to the desired pattern k, additively for each component and chunk value:

$$P_i = \Sigma_k P M_{ki} \tag{4}$$

While the most active chunk is usually retrieved, a blending process (i.e., a *blended retrieval;* see Lebiere, 1999; Wallach & Lebiere, 2003) can also be applied that returns a derived output V reflecting the similarity S_{ij} between the values of the content of all chunks i and compromise value j, weighted by their retrieval probabilities P_i reflecting their activations and similarity scores:

$$V = argmin\, \Sigma_i\, P_i\left(1 - S_{ji}\right) \tag{5}$$

This process enables the generation of continuous values (e.g., probabilities) in a process akin to weighted interpolation.

Cognitive architectures can be used for AI purposes by leveraging basic cognitive mechanisms while not necessarily respecting all their constraints. 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. In that case, it is arguably better to specifically state which aspects of the model are not constrained by data, and rather than mockup 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.

2.1 Tripartite Explanation of How Biases Emerge

An essential feature in being able to explain how a model performs decision-making is to examine not only the sources of generating expertise, but also to examine both *where* heuristics come from and *how* they are applied; and how they lead to biased behavior. In our account, biases arise from three sources: (1) the mechanisms and limitations of the human cognitive architecture; (2) the information structure of the task environment; and (3) the use of heuristics and strategies to adapt performance to the dual constraints of cognition and environment. The first level of description entails an understanding of the constraints imposed by the mechanisms and limitations of the cognitive architecture. These include an understanding of the impact of recency and frequency on the likelihood of an instance being retrieved. Other sources of constraint include the serial nature of human reasoning, system, memory limitations, and matching human time-course of responses. The second level of description entails an understanding of the constraints imposed by the task environment. An understanding of the statistical and quantifiable regularities within the task environment drives the overall ability and rate of learning, and the nature of environmental feedback provides further evidence. Using an example from Simon (1990); if you want to study the movement of an ant across the beach you need look no further than the hills and valleys in the sand to determine its path. This level of description is generally captured in ACT-R by specific goals that drive a set of productions that represent the cognitive skills required for solving the task. The third level of description entails an understanding of how the joint constraints of architecture and task environment influence the kinds of heuristics and strategies available to the model. This is the explanation of the selection and sequence of production rules firing. In strategy selection, even simple heuristic structures can greatly influence the output of the model, which in turn could overly constrain decision-making while also making complex problems tractable. In other words, the detection of affordances provided by the task environment influences the kinds of information that the model can accumulate and the actions that the model may perform.

We can also use *model tracing* to measure the genesis and mitigation of biases. Model tracing is a technique where a model is forced to respond with some or all of the same values as a human participant, accumulating the same experiences. The internal states of the model can then be examined to determine the influence of these experiences on future decisions (Somers et al., 2020; Cranford et al., 2020). By examining commonalities between the model's internal states and human behavior, modelers are potentially able to make causal claims about the nature of mental processes; that is, to explain how human performance (a bias) is produced by various cognitive mechanisms and their interaction. We describe several examples below.

3 Using Cognitive Models to Understand Human Performance

3.1 Cognitive Models of Sensemaking

As an instance of capturing human decision making patterns, we will focus on a cognitive model of sensemaking in the context of six geospatial intelligence analysis tasks for the IARPA Integrating Cognitive-Neuroscience Architectures for

Understanding Sensemaking (ICArUS) Program (see Lebiere et al., 2013 for a complete description of the tasks and quantitative fits). Sensemaking is a concept that has been used to define a class of activities and tasks in which there is an active seeking and processing of information to achieve understanding about some state of affairs in the world. Our sensemaking model is composed of several related components. The first learns statistical patterns of events and generates probability distributions of category membership based on the spatial location and frequency of these events. The second applies probabilistic decision rules in order to generate and revise probability distributions of category membership, and then makes decisions about the allocation of resources based on the probabilities of the causes of perceived events.

The first task involved watching a series of attacks on a map from several groups and predicting the center of each group's area. When group centers were generated directly from a retrieval of events represented in memory, the blended retrieval process in ACT-R reflected a disproportionate influence of the most recent events given their higher activation. A strategy to combat this recency bias consisted of generating a final response by performing a blended retrieval over all the group centers (both current and past) stored in memory, thereby giving more weight to earlier events by compounding the influence of earlier centers over the subsequent blended retrievals. This effectively implements an anchoring-and-adjustment process where each new estimate is a combination of the previous ones together with the new evidence. Moreover, because there are an equal number of centroid-of-centroids chunks, there is no effect of base-rate on the model's later probability judgments, even though the base-rate for each category is implicitly available in the model based on the number of recallable events.

To leverage an instance-based learning (IBLT) approach for probability adjustment and resource allocation, the ACT-R model's memory was seeded with a range of instances consisting of triplets: an initial probability, an adjustment factor, and the resulting probability. The factor is set by the explicit rules of the task. When the model is asked to estimate a probability, it simply performs a blended retrieval specifying prior and factor and then outputs the posterior probability. When provided with linear similarities between probabilities (and factors), the primary effect is an underestimation of the adjusted probability for much of the initial probability range (i.e., an anchoring bias), with an overestimation on the lower end of the range (i.e., confirmation bias). The model then generates a resource allocation distribution by focusing on the leading category and determining how many resources to allocate to that category. By priming the model with the winner-take-all and probability matching strategies, it is possible for the model to learn any strategy in between them. Instance-based learning can thus be seen in this instance as a highly flexible metacognitive strategy with the effects themselves *a priori* predictions of the architecture.

In more general terms, when evidence is received, it is matched against various hypotheses and the best matching one is retrieved, leading to a boost in activation. If contradictory evidence starts accumulating, two biases will emerge. First, new evidence will sometimes be misinterpreted because the dominant hypothesis is most active and can overcome some degree of mismatch. Second, even if the evidence is correctly interpreted and the correct hypothesis reinforced, for the new hypothesis to attain primacy it will take some time to sufficiently build activation and for the activation of the previously dominant hypothesis to sufficiently decay. This process has been given a number of names, from anchoring bias to persistence of discredited evidence.

3.2 Cognitive Models to Aid Autonomous Defenses

In recent research in cyber-deception for defense, we have developed methods for using cognitive models of adversaries to drive personalized, adaptive defense algorithms (Cranford et al., 2020). Many current cyber-security algorithms are developed using traditional game-theory methods that often assume perfectly rational adversaries. Meanwhile, cognitive architectures such ACT-R aim to accurately represent the cognitive processes that give rise to boundedly rational human behavior and emergent cognitive biases. For example, Cranford et al. (2020) created an IBL model in ACT-R that accurately predicts attacker decisions in an experimental game that simulates an insider attack scenario, dubbed the Insider Attack Game (IAG). An automated defense algorithm optimizes the allocation of limited defenders across targets in a network, and the rate at which it sends (possibly deceptive) signals to attackers in an effort to deter attacks. The goal of the defense algorithm is to optimize the rate of deceptive signals so that the attacker maintains belief in the signal and withdraws in its presence. The results showed how the combination of task structure and cognitive biases, such as confirmation bias, lead to far more attacks than predicted of a perfectly rational adversary. According to IBLT, a confirmation bias arose due to effects of frequency and recency of instances in memory. Because the task was structured in such a way that all targets have positive expected values (i.e., it is more likely than not that a target is undefended), it was more likely to experience a positive reward than a negative penalty on any given trial. Given enough positive reinforcement early on, the attacker begins to generate expectations of positive rewards that are self-reinforcing, and it is unlikely for them to experience enough negative outcomes in sequence to introduce negative expectations. As a result, the confirmation bias emerges naturally from the availability of positive information in memory that influences future retrievals.

One reason the defense algorithm performed less well than expected was that it failed to account for human biases. Another shortcoming of traditional security algorithms is that they are static and tailored to a population, or average, user. In reality, individuals display vastly different behavior. As Cranford et al. (2020) showed, the cognitive model not only predicted the mean human behavior throughout the game, it was able to accurately predict the full range of human behavior, solely as a result of stochasticity in retrieval leading to different trajectories of experience, thereby influencing future decisions. As such, some model runs do not exhibit strong confirmation bias, just like some humans do not. Given our model can account for the full range of human behavior, we leveraged model-tracing techniques developed from research on cognitive tutors, to adapt a model run to a specific individual (Corbett and Anderson, 1995).

As an adversary interacts with the system, we use model-tracing to align the model's memory with the observed (and sometimes inferred) experiences of the human. With more experience, the model is able to more accurately predict an attacker's decisions in real time. Using this method, the model can drive an adaptive defense algorithm. Cranford et al. (2020) attempted to alleviate the observed confirmation bias in the IAG by providing blocks of only truthful signals to those participants that are expected to continue to attack despite the presence of a monitor signal. The idea, according to IBLT, is that these attackers will only have negative attacking experiences given a monitor signal and, with enough of such experiences, will start withdrawing. The technique proved effective, but only for a portion of participants. Further examination showed

that a portion of participants that continued to attack despite the intervention also reported completely ignoring a signal feature. Our own model that ignored the signal in its decisions accurately reflected this behavior. These results show that any efforts to adjust the rate of signals for such attackers is in vain, but also that these techniques could be used instead to shift coverage toward the more desirable targets or to craft more desirable targets (e.g., honeypots).

4 Cognitively-Inspired Models for Detection

As an extension of the IARPA ICArUS project, we modeled a malware dataset identified by Mandiant (2013) using dynamic sandbox output from ANUBIS (2014). From the ANUBIS data, a total of 1740 malware attributes were identified. We studied all families where there were at least 5 samples successfully processed by ANUBIS, which provided 15 families (including BISCUIT, NEWSREELS, GREENCAT, and COOKIEBAG) and 137 samples. Based on malware family description, we associated a set of tasks with each malware family (that each malware in that family was designed to perform). In total, 30 malware tasks were identified for the given malware instances. On average, each family performed 9 tasks.

The cognitive model operates as follows: 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 model primarily leverages the activation calculus underlying retrieval from 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 long-term memory of the chunks representing instances of those families. To assign intent in a second pass the model uses a similar process to generate likely intents from a representation linking each malware family to known intents.

The model uses 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 and represents the set of attributes together with the family identification. The activation of each chunk is learned by to the mechanism described in Section 2. The power law decay makes it sensitive to the recency of presentation, allowing both for old malware instances to quickly decay away as well as for new ones to rapidly reach prominence. If the same instance (i.e., same attributes and family) is presented multiple times, the activation will also reflect the frequency of presentation. The effect of context, as represented by the set of attributes of the current malware, will be reflected through the partial matching mechanism. The match score of a chunk to the current context will reflect the similarity between the attribute sets of the current malware sample and each instance in memory, as measured by the dot product between the respective attribute vectors. The retrieval process then extracts from the chunk its family identification. The blending mechanism computes a probability distribution over all family values, reflecting the activation of each instance.

The instances learn to associate the probability distribution over families computed for the given malware with its actual intents. Given a new malware instance, a retrieval process matches its family probability distribution against those of previous instances

and extracts the probability of each intent using the same blending process used for generating the family probabilities. Intents reaching the 50% threshold are again selected. The key aspect of this process is that it is now sensitive to the entire probability distribution over families rather than simply a sum of its values. Table 1 indicates the F1 score model as well as several standard machine learning techniques for LOOCV.

Table 1. Performance Comparison of ACT-R vs Other Methods

Method	DT	NB	LOG-REG	SVM	**ACT-R IBL**
LOOCV F1 Score	.80	.71	.82	.89	**.93**

To see how the model generalizes to unseen malware families, we performed a leave-one-family-out comparison where we test against one previously unseen malware family. The instance-based model significantly outperformed the other approaches in terms of precision, recall and F1. We also tested generalizability using the GVDG and Metasploit malware generation tools, with the model still consistently outperforming other techniques, including when only receiving a sparse 10% of the training.

4.1 Intrusion Detection

The vast majority of state-of-the-art techniques for intrusion detection are ML-based, with ANN, SVM, Decision Tree, Bayesian, k-means, k-NN, and Fuzzy Logic being the most used across all IDS techniques reviewed by Hindy et al. (2020). ML techniques usually require large amounts of data to learn from and build a knowledge representation that can be used to inform the decision classifier (Hamed, Ernst, & Kremer, 2018). While ML techniques have proven useful for many classification tasks across many domains, particularly in image classification, the cybersecurity domain presents unique challenges in that the patterns of incoming data are constantly changing in real time (Hodo et al., 2017). Therefore, adaptive techniques are warranted that can learn quickly with new and sparse data.

IBL techniques grounded in cognitive architectures may prove a useful technique then because such models need few training instances (and no separate learning stage) to make highly accurate predictions and can quickly adapt to changing environments as new instances are added to the knowledge base. Some recent IDS techniques have shown promise in overcoming limitations stemming from novel attacks. For example, Al-Hawawreh, Moustafa, and Sitnikova (2018) use a deep auto-encoder and deep feedforward neural network to learn with new incoming information. Similarly, Salo, Nassif, and Essex (2019) used an instance-based algorithm to aid classification in their ensemble approach to IDS, which alleviated limitations of classifying novel instances.

In their review of IDS techniques and datasets, Hindy et al. (2020) point out that one limitation of current ML techniques for IDS are that datasets lack real-life characteristics of recent network traffic and therefore fail to generalize under real-world deployment, cannot adapt to changes in network topology, and perform poorly against novel attacks. According to their review, the KDD-99 dataset is the most prominently used dataset, in over 50% of IDS techniques reviewed, but is outdated. Therefore, Moustafa and Slay (2016) more recently created a dataset to combat the lack of modern low-footprint attack styles and modern normal traffic scenarios seen in prior datasets and included a different distribution of training and testing sets. Their dataset, the

UNSW-NB15, contains 9 attack types, including Fuzzers, Analysis, Backdoors, DoS, Exploits, Reconnaissance, Shellcode, Worms, and other "generic" attacks, and defines 49 features of the network traffic, providing for robust coverage of current attack profiles. Moustafa compared the performance of several IDS techniques between the KDD-99 and UNSW-NB15 datasets. The results showed that detection rates were generally higher for the KDD-99 datasets compared to the UNSW-NB15, highlighting the greater complexity and realism of the newer dataset. The present research therefore uses the UNSW-NB15 dataset to evaluate the performance of our IBL cognitive modeling approach to intrusion detection.

4.2 Cognitive Model description and IDS methodology

The cognitive model is a version of ACT-R models of categorization (e.g., Lebiere, 2005). For each attack the model associates a set of features describing the attack with its type. The current set of features was the same chosen in existing models (cite) to make our approach more comparable to existing ML models. A more elaborate approach would have been to use the reinforcement learning-like ACT-R mechanism that learns production utilities from external rewards to select the most effective feature set (Martin et al., 2018). Since the partial matching mechanism is key to generalize across instances, similarities were set between the feature data types. A linear similarity function was used for real values with a scaling factor of 0.1. For integer values a log-ratio similarity function was used with a base of 2. For symbolic values, the maximum dissimilarity value was left at the default value of -1.0.

Even though the cognitive model doesn't require a separate training phase, to make our results comparable to ML algorithms we used the separate training and testing sets provided. Rather than use all the examples provided in the training set (over 175 thousand) we created 10 instances for each attack type (roughly the log of the number of examples of that type), and 90 instances of normal traffic (same number of examples as the total number of intrusion instances) for a total of 180 instances, or about $1/1000^{th}$ of the total size of the training data set.

Results. We tested a range of values of the mismatch penalty parameter from 0.5 to 2.5 as well as both low and moderate noise values (0.01 and 0.1, respectively). Performance of the model was largely insensitive to parameters. The percentage of correct attack type identification was greater than 50% (as opposed to a change probability of 10%) with a high of 58%. Performance rises to 81% correct when only detecting intrusion as opposed to diagnosing attack type.

5 Conclusions

We introduced our methodology for autonomous intelligent cyber-defense agents based on cognitive models of human intelligence. Those models capture powerful characteristics of human cognition such as efficient learning, robust generalization, and continuous adaptivity. They can also reflect the limitations of human teammates and adversaries in order to compensate or exploit them. This approach aims to leverage

decades of research and applications of cognitive architectures (Laird, Lebiere & Rosenbloom, 2017) to serve as the basis for a broad, validated AICA framework.

References

1. Corbett, A. T. & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction,* 4, 253-278.
2. Cranford, E. A., Gonzalez, C., Aggarwal, P., Cooney, S., Tambe, M., & Lebiere, C. (2020). Adaptive cyber deception: Cognitively-informed signaling for cyber defense. In *Proceedings of the 53rd Hawaii International Conference on System Sciences* (pp. 1885-1894). Maui, HI.
3. Erev, I., Ert, E., Roth, A. E., Haruvy, E., Herzog, S., Hau, R., Hertwig, R., Stewart, T., West, R., Lebiere, C. (2010). A choice prediction competition, for choices from experience and from description. *Journal of Behavioral Decision Making 23(1)*: 15-47.
4. Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R., & Bellekens, X. (2020). A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems. *IEEE Access*.
5. Kahneman, D. (2011). Thinking, fast and slow. Macmillan.
6. Laird, J. E., Lebiere, C. & Rosenbloom, P. S. (2017). A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. *AI Magazine 38(4)*. https://doi.org/10.1609/aimag.v38i4.2744
7. Lebiere, C. (2005). Constrained functionality: Application of the ACT-R cognitive architecture to the AMBR modeling comparison. In Gluck, K, & Pew, R. (Eds.) *Modeling Human Behavior with Integrated Cognitive Architectures*. Mahwah, NJ: Erlbaum.
8. 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 2nd Conference on Artificial General Intelligence*. Atlantis Press.
9. Lebiere, C., Pirolli, P., Thomson, R., Paik, J., Rutledge-Taylor, M., Staszewski, J., & Anderson, J. (2014). A Functional Model of Sensemaking in a Neurocognitive Architecture. *Computational Intelligence and Neuroscience*.
10. Lebiere, C., Gonzalez, C., & Martin, M. (2007) Instance-Based Decision-Making Model of Repeated Binary Choice. *International Conference on Cognitive Modeling*.
11. Martin, M., Lebiere, C., Fields, M, & Lennon, C. (2018). Learning Features While Learning to Classify: A Cognitive Model of Classification and Feature Selection for Autonomous Systems. *Computational and Mathematical Organization Theory*, 21(3),1-32.
12. Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective,* 25(1-3), 18-31.
13. 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*. Philadelphia, PA.
14. Somers, S., Mitsopoulos, C., Lebiere, C., & Thomson, R. (2019). CogXAI: Cognitive-Level Salience for Explainable Artificial Intelligence. In *Proceedings of ICCM*.
15. Sanner, S., Anderson, J., Lebiere, C., & Lovett, M. (2000). Achieving efficient and cognitively plausible learning in backgammon. *Proceedings of ICML, 823-830.*
16. Thomson, R. Lebiere, C., & Bennati, S. (2014). Human, Model, and Machine: A Complementary Approach to Big Data. In A*ssociation for Computing Machinery Proceedings of the IARPA Workshop on Human Centered Big Data Research*. Raleigh, NC.
17. Thomson, R. Lebiere, C., & Bennati, S. (2014). A General Instance-Based Model of Sensemaking in a Functional Architecture. In *Proceedings of the 23rd Annual Behavior Representation in Modeling and Simulation Conference*. Washington, DC.