A Novel Approach to Intrusion Detection Using a Cognitively-Inspired Algorithm

Robert Thomson
Army Cyber Institute, USMA
robert.thomson@westpoint.edu

Edward Cranford, Sterling Somers, & Christian Lebiere
Carnegie Mellon University
{cranford, psomers, cl}@cmu.edu

Abstract

We propose a novel algorithm for white-box intrusion detection using a cognitive model consistent with the principles of instance-based learning theory. 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. Expanding upon previous research in malware identification and personalized deceptive signaling, the present paper presents a cognitive model able to achieve over 70% accuracy identifying anomalous (vs normal) traffic on the UNSW-NB15 dataset with only 8 features and using only one sample from each attack and 9 normal samples. Accuracy linearly increases to over 85% using up to 100x more samples. A cognitively-inspired salience algorithm then shows the relative impact of each feature driving correct vs incorrect classifications. Implications for integrating this algorithm with human operators are discussed.

Keywords: Cognitive Model, Intrusion Detection, Salience, Cybersecurity

1. Introduction

Traditional network intrusion detection systems (NIDS) shield users from malicious traffic using a set of rules to separate anomalous traffic from *normal* behavior on the network. These rules tend to generate a lot of false positive alerts that a human analyst is required to sift through. NIDS are prone to miss novel attacks known as *zero-day* attacks as they circumvent existing rulesets. These rulesets thus require constant updating as new attacks are detected.

One of the primary roles of a cyber analyst triaging these alerts is to quickly *white-list* these false alarms and send suspicious activity for further forensic evaluation. This is a demanding and repetitive task, which can lead to attentional and other workload-based cognitive biases (Greenlee, Funke, Warm, et al., 2016).

To address the dual-challenge of the brittleness of rule-based NIDS and analyst workload, substantial research has focused on automating the most repetitive tasks using artificial intelligence (AI) techniques. Many of these techniques require substantial preprocessing steps and training data (Strubel, Ganesh, & McCallum, 2020), are hard to re-train (Yang, Liu, Chen, & Tong, 2019), and fall prey to adversarial attacks (Madry, Makelow, Schmidt, et al., 2017). In this paper, we describe an alternative approach to intrusion detection using a cognitively-inspired algorithm previously successful for malware identification (Nunes, Buto, Shakarian, et al., 2015) that does not require substantial training, learns online, does not require the same degree of feature engineering, and is natively explainable.

In this paper we present an overview of cognitive modeling and describe two previous use cases in cybersecurity: 1) a model of malware identification resilient against traditional adversarial techniques, and 2) an adaptive model of personalized deceptive signaling for phishing defense. We then describe our algorithm and highlight its performance on the UNSW-NB15 dataset (Moustafa & Slay, 2016) with limited training. Finally, we present an introspective *cognitive salience* technique that highlights the importance of each feature to the model's classification decision.

1.1. 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 (Anderson & Lebiere, 1998; Anderson et al., 2004, 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 (for recent reviews see Ritter, Tehranchi, & Oury, 2019 and Thomson et al., 2015). Symbolic information structures provide representational and reasoning characteristics reflective of expert human performance, while sub-symbolic statistical processes

URI: https://hdl.handle.net/10125/106493
978-0-9981331-7-1

provide adaptivity and robustness. For the purpose of this paper, we focus on declarative knowledge.

Declarative knowledge is represented formally in terms of chunks. Chunks are data structures that consist of an ordered list of attribute-value pairs. Chunks are retrieved from declarative memory by an adaptive activation process that attempts to estimate the probability of needing a piece of information, following a softmax distribution:

$$P_i = \frac{\left(e^{A_i/s}\right)}{\left(\sum_j e^{A_j/s}\right)} \quad (1)$$

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 (MP_i) reflecting the degree to which the chunk matches the retrieval request, and finally a noise value ($\mathcal{E}_i$) reflecting stochasticity:

$$A_i = B_i + SA_i + MP_i + \epsilon_i \quad (2)$$

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 (SA_i) and partial matching (MP_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:

$$B_i = ln\left(\sum_{j=1}^{n} t_i^{-d}\right) \quad (3)$$

Base-level activation corresponds to the Bayesian prior of a chunk's activation and provides an automated procedure for frequency-based strengthening as well as temporal discounting following the power laws of practice and forgetting. Chunks are also compared to the desired retrieval pattern using a partial matching mechanism (MP_i) that subtracts from the activation of a chunk i its degree of mismatch M_{ki} to the desired pattern k, scaled by a mismatch parameter factor MP, additively for each component and chunk value:

$$MP_i = \sum_k MP \cdot M_{ki} \quad (4)$$

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 squared similarity $Sim(V_t, V_{it})^2$ between the values of the content of all candidate chunks V_{it} and compromise value V_t, weighted by their retrieval probabilities P_i reflecting their activation and degree of match:

$$V = \underset{V_t}{argmin} \sum_{i=1}^{n} P_i \cdot Sim(V_t, V_{it})^2 \quad (5)$$

This process enables generating continuous values (i.e., probabilities) akin to weighted interpolation. In the simplest case, where the values are numerical and the similarity function is linear, the process simplifies to a weighted average by the probability of retrieval.

1.2. Cognitive Salience for Feature Importance

To better understand why a given decision was made, one introspection technique that can be beneficial is to understand the most important feature(s) that influenced the retrieval. In ML, feature importance serves a dual role: (1) in preprocessing steps to reduce the overall dimensionality and noise within the dataset, and (2) as an explainability measure to understand why a given algorithm has made the decision that it did (Wojtas & Chen, 2020). While measures of model-based feature importance are prevalent in ML packages (e.g., the fitted attribute *feature_importances_* in several ML packages and function *permutation_importance* in the *sklearn* package), there are fewer off-the-shelf techniques for class-based feature importance. A workaround is to use a random forest algorithm and train a *single class vs. all* for each class in the dataset.

In recent work on explainable AI, Somers et al. (2019) developed a method for introspecting upon the decisions of a cognitive model by computing a decision-specific feature importance called *cognitive salience*. Cognitive salience has been applied to understand the decisions of deep reinforcement learners paired with a cognitive model using both the Starcraft 2 and OpenAI gym environments (Mitsopoulos et al., 2021). It is a method to introspect individual decisions (i.e., classifications) by computing the derivative of the blending process to determine the degree of influence each feature has on a given decision. This adds a facet of interpretability to the cognitive model, allowing the modeler to introspect upon individual decisions, rather than statistically averaging over all situations.

The match score is computed by subtracting from the base-level activation (B_i) of chunk i the partial matching (Equation 4) term reflecting MP mismatch penalty and the degree of match for features k spanning the l matching slots for which a value is provided:

$$M_i = B_i + \sum_{k=1}^{l} MP \cdot Sim(F_k, v_{ik}) \qquad (6)$$

The derivative of the match score with respect to feature F_k can be readily computed as the derivative of the blending process (Equation 5):

$$S(V_t, F_k) = \frac{MP}{t} \sum_{i=1}^{n} P_i \cdot \left(\frac{\partial Sim(F_k, v_{ik})}{\partial F_k} - \sum_{j=1}^{n} P_j \cdot \frac{\partial Sim(F_k, v_{jk})}{\partial F_k} \right) \qquad (7)$$

This provides a closed form of the gradient-based salience of its representational features on its decision.

Cognitive salience is sensitive to the decision context and can thus be used to determine not only which features are most influential for making accurate classifications overall, but also class-specific decisions. Such information provides interpretability to not only reduce the dimensionality in the dataset, but also provide explainability for why a classification was made in each context. Cognitive salience thus offers a thorough method for evaluating feature importance compared to ML approaches that focus on model- or class-based methods separately.

2. Cognitive Models in Cybersecurity

2.1. Malware Identification

Extending from the IARPA ICArUS project focused on modeling cognitive biases in intelligence analysis, we took inspiration from the cognitive model of the analysts (see Lebiere et al., 2013) and applied it to identify malware by treating dynamic sandbox output as a single instance in memory (see Figure 1).

Cognitive models inspired by instance-based learning theory (IBL; Gonzalez, Lerch, & Lebiere, 2003) treat expertise as the accumulation of *instances*, where decisions are based on a match to prior instances. The cognitive model operates by generating a probability distribution over a set of malware families (e.g., COOKIEBAG, BISCUIT), then inferring a set of likely malware tasks (e.g., *beacon*, *upload*, *takeScreenShots*) 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 generated by sandbox software (e.g., *usesDll(X)*; that is, the malware uses a library *X*). Attributes are essentially binary features associated with a piece of malware that can be observed using dynamic and static analysis, while the tasks tell us the higher-level purpose of the malware.

INPUT: New malware sample i, historical malware corpus $\mathcal{M}$.
OUTPUT: Set of tasks associated with sample i.
for query malware sample i **do**
  for all j in $\mathcal{M}$ **do**
    $B_j = \beta_j$
    $P_j = mp \times \frac{|attribs(i) \cap attribs(j)|}{\sqrt{|attribs(i)| \times |attribs(j)|}}$
    for $a \in attribs(i)$ **do**
      if $a \in attribs(j)$ **then**
        $s_{ij} \mathrel{+}= log(\frac{|\mathcal{M}|}{|fan(a)|})$
      else
        $s_{ij} \mathrel{+}= log(\frac{1}{|\mathcal{M}|})$
      end if
    end for
    $S_j = \sum_j \frac{s_{ij}}{|attribs(i)|}$
    Calculate A_j as per Equation 1
  end for
  Calculate p_j as per Equation 2
  $p_f = \sum_{j \in f s.t. A_j \geq \tau} p_j$
  $t_p = \{t \in T | p_f \geq 0.5\}$
end for

Figure 1. Algorithmic Representation of the Cognitive Model from Nunes et al., (2020).

The model uses an iterative instance-based 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 the mechanisms described in Section 1.1. 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, meaning that the detector is sensitive to base rate. 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 a single most likely family.

We validated the model against the Mandiant malware dataset using dynamic sandbox output from ANUBIS and Cuckoo (Nunes et al., 2015; 2020). From the ANUBIS data, a total of 1740 malware attributes were identified. Families with at least five samples successfully processed by ANUBIS were included in the dataset, which provided 15 families and 137 samples. Based on the 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.

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 ML-based approaches including a commercial offering based on DARPA's Cyber Genome project; with an unbiased F1-score of .97 (vs an average of .9 for the other techniques). We also tested generalizability using the GVDG and Metasploit malware generation tools, with the model consistently outperforming other techniques, including when only receiving a sparse 10% of the training.

2.2. Autonomous Deceptive Signaling

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., 2021). Many current cyber-security algorithms are developed using traditional game-theoretic methods that often assume perfectly rational adversaries. Meanwhile, cognitive models aim to accurately represent the cognitive processes that give rise to boundedly rational human behavior and emergent cognitive biases. Cranford et al. (2020a; 2020b, 2021) developed an IBL model that accurately predicted attacker decisions in a simulated insider attack scenario. The task, dubbed the Insider Attack Game (IAG), consists of an automated defense algorithm that 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. Results (see Figure 2) showed attackers attack far more often than predicted by a rational adversary.

The cognitive model implied that a confirmation bias arose due to effect of frequency and recency on the attackers' memory. Because the task was structured in such a way that all targets have positive expected values (only two defenders cover six targets), the attacker 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. Also, given the higher probability of success than failure 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 affects future expectations.

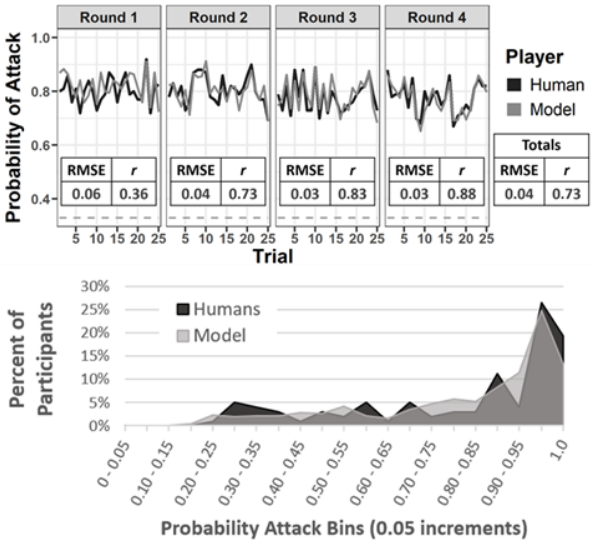

Figure 2. Model fits from the Insider Attack Game. The behavior generative model predicts average human performance with a high degree of reliability across blocks as well as individual differences. Figure adapted from Cranford et al. (2020a; 2021).

Another shortcoming of traditional security algorithms is that they are static and tailored to a population or average user. Individuals display vastly different behaviors in practice. As Cranford et al. (2021) 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 as a result of stochasticity in retrieval leading to different trajectories of experience, with some model runs exhibiting strong confirmation bias while others do not.

Given our model can account for the full range of human behavior, we leveraged model tracing to adapt a model run to a specific individual from Cranford et al. (2020b). Model tracing is a technique where a model is forced to respond with the same values as a human (or AI) agent, accruing the same experiences (Corbett & Anderson, 1995). As the adversary interacts with the system, it aligns the model's memory with the observed (and inferred) experiences of the human participant.

With more experience, the model more accurately predicted an attacker's decisions in real time, $r^2 = 0.95$.

To summarize, cognitive models exhibit adaptivity with limited data and also have the ability to explain the intentions of human operators and adversaries.

3. Cognitive Model of Intrusion Detection

The vast majority of state-of-the-art techniques for intrusion detection are ML-based, with neural network, SVM, Decision Tree, Bayesian, reinforcement learners, k-means, k-NN, and Fuzzy Logic being the most used across all IDS techniques reviewed by Hindy et al. (2020). ML techniques, especially neural-network based ones, frequently require large amounts of training 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 a multitude of classification tasks across many domains, the cybersecurity domain presents a relatively unique challenge in that the patterns of incoming data are constantly changing in real time (Hodo, Bellekens, Hamilton, Tachtatzis, & Atkinson, 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 because these models leverage regularities and constraints abstracted in the structure and mechanisms of cognitive architectures to require 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, Muna, Moustafa, & Sitnikova (2018) use a deep auto-encoder and deep feedforward neural network to learn with new incoming information. Similarly, Salo, Nassif, & 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 current ML techniques for IDS rely on datasets which lack real-life characteristics of recent network traffic and may 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.

Moustafa & 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. Their 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 cognitive modeling approach to intrusion detection.

3.1. Methodology

The cognitive model is a version of instance-based learning ACT-R models of categorization (Lebiere, 2005) and malware identification (Nunes et al., 2020). The model's input takes a row from the UNSW-NB15 training dataset and creates a chunk associating the set of features with its ground truth (one of the 9 attack categories or normal traffic). Since partial matching 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 due to the constrained range of values. For integer values a log-ratio similarity function was used with a base of 2 to scale across a large range of values. For symbolic values, the maximum dissimilarity value was left at the default value of -1.0, which is equivalent to maximally dissimilar (as would be the case with one-hot encoded values). All other parameters of the cognitive architecture were left at their default values: time decay d at 0.5, mismatch penalty MP at 2.5, activation noise ε at .25 and blending temperature t at 1.0.

Rather than use all the examples provided in the training set (over 175,000) we manipulated the number of instances of each attack type to determine how it impacts model accuracy. To prevent frequency effects from biasing results toward the most common categories, we sub-sampled and balanced the training instances by including the same number for each intrusion category as well as the same number of normal instances as the total number of intrusion instances.

To provide comparison, our model's performance was compared to several ML techniques including a decision tree, random forest, logistic regression, multi-layer perceptron, and convolutional neural network. To preprocess the data to be amenable with these techniques, the training and testing data were one-hot encoded and normalized via StandardScalar package to be as similar-to the cognitive model as possible, with no additional feature selection or pruning. Thus, these models will not have near-ceiling performance. We also investigated the comparative performance between the models' performing an 80/20 train-test split compared

to training on only 180 instances (comparable to the cognitive model results discussed below).

3.2. Results

We tested a range of number of instances per intrusion category, from 1 to 50. One instance for each of 9 intrusion categories resulted in 18 total instances, including 9 normal instances. This represents 1/10,000th of the total number instances. Even using 50 instances per category only represents half of one percent of the full training set. Figure 3 displays performance as a function of the number of training instances.

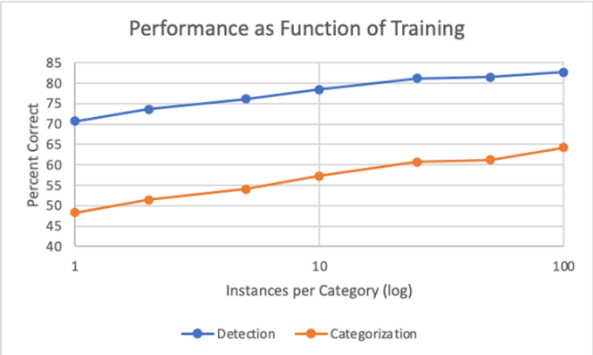

Figure 3. Percent correct as a function of number of training instances per intrusion category. Anomaly detection (binary classification) performance increased from 71-83% while multi-class intrusion detection ranged from 48-65% accuracy.

Of note is the 70% accuracy in detection using only a single instance from each attack category and nine instances of normal activity. For reference, with 10 samples per attack category and 90 normal samples, average binary classification rises to 73% while multi-class accuracy rises 50%. Similar to the decision tree and random forest model results described below, precision, recall, and F1 rates are all similar to accuracy, with the model exhibiting a binary classification sensitivity of .70 and specificity of .89.

To compare this performance with traditional ML techniques trained on the whole training dataset, both the decision tree and random forest models approach 86% multi-class categorization accuracy with binary classification performance near 96%, with similarly high values for precision, recall, and F1-score. Conversely, logistic regression only has 60.5% multi-class and 75% binary classification accuracy with reduced precision (47.5%) and F1-score (49%) for multi-class. Our multi-layer perception with (20,20) hidden layer states, *relu* activation and an *adam* solver only obtained multi-class 63.8% accuracy (56.6% F1-score) and 73.6% binary performance. A convolutional neural network with 3 dense layers (189 *relu*, 20 *relu*, 10 *softmax*), *sparse categorical crossentropy* loss function, and *adam* optimizer, performed comparably to the decision-tree and random forest models.

If the ML models were only trained on 180 instances, then interestingly the decision-tree and random forest maintained 75.3% multi-class and 83.2% multi-class, almost identical to the cognitive model. The logistic regression dropped to 44.9% accuracy (27.9% F1-score) for multi-class and 74.3% binary classification accuracy. The multi-layer perceptron and CNN were not always able to converge but exhibited interesting performance averaging 4.0% accuracy (43.9% precision, 3.7% F1-score) for multi-class and 36.4% performance for binary classification. Thus, while the cognitive model's performance was not as high overall as the best-performing ML techniques (e.g., random forest), they were comparable and higher than more data-hungry approaches, and maintained a high specificity.

The focus of this task was not to maximize performance but to determine whether the model would perform similarly to the malware identification model (Nunes et al., 2020) and how it may differ from traditional ML techniques. Table 1 displays a sample confusion matrix for the intrusion categorization task with multi-class categorization performance of 67.4% and binary classification performance of 83.7%.

Table 1. Sample confusion matrix between intrusion categories.

Categories	Generic	Worms	DoS	Exploits	Recon	Shell	Fuzzers	Analysis	Backdoor	Normal
Generic	17358	161	28	78	30	123	17	13	9	1054
Worms	0	32	0	2	0	3	0	1	0	6
DoS	16	22	458	329	154	158	70	836	731	1116
Exploits	49	1526	891	2451	718	1027	165	1180	679	2446
Recon	0	203	55	215	1003	195	1	77	89	1658
Shell	0	0	0	0	57	110	1	0	1	209
Fuzzers	28	123	133	121	602	726	634	768	134	2792
Analysis	14	0	38	3	35	40	2	380	75	90
Backdoor	14	5	17	7	59	17	2	354	42	66
Normal	11	269	39	344	682	559	1247	768	63	33018

Confusion probabilities between categories may vary substantially across runs depending on the noise parameter as well as which training instances in each category were selected. The Normal category is well recognized but also substantially intrudes upon various intrusion categories. Some categories such as Worms and Generic are also well recognized. Others such as Shellcode and Backdoor are very poorly recognized, while DOS, Exploits and Fuzzers substantially intrude upon each other. When compared against the decision tree and random forest trained on the complete set, the patterns of confusion are similar.

3.3 Cognitive Salience in Intrusion Detection

In an ongoing effort to further expand the model, we trained the model on the entire database and used Cognitive Salience (see Section 1.2) as a diagnostic tool to determine how the features affect binary intrusion detection classification (attack/no-attack). Figure 4 illustrates salience for the top 20 salient features (limited for illustration) for correct (blue) and incorrect (orange) 'attack' and 'no-attack' classifications. Data for this analysis was gathered by random sampling the UNSW-NB15 test set (5000 samples) and computing the cognitive salience of the classifications using the cognitive model described in section 3.1.

As illustrated in Figure 4, salience values can be either positive or negative, indicating whether a feature contributes to the classification or whether the feature detracts from the classification, with the degree of influence indicated by the length of the bar. There are a set of features which largely drive misclassification. For example, the magnitude of 'dload' is overweighted in misclassification of 'attack' (Panel A) and 'sttl' is underweighted. This means that removing 'dload' and heavily-weighting 'sttl' has the potential to increase accuracy in detecting attacks.

Similarly, random forest models select 'sttl' as the most important feature. Conversely, in Panel B we see that 'dload' is highly weighted in classifying correct normal traffic, so it is not something which should be removed. For reference, 'dload' is the 6th highest feature in the random forest model. Of interest, 'sbytes' is the 2nd highest feature in the random forest model but is not in the top-20 most influential features in the cognitive model. More research needs to be done to understand the differences in how aggregate salience and random forest's feature importance differ in their judgments given the similar patterns of confusion in multi-class categorization.

While aggregate salience gives a global indication of the influence of features on the classification, the Cognitive Salience technique is sensitive to particular cases of classification. Figure 5 illustrates the top 20 most influential features for an 'Attack' (Panel A) and 'No-Attack' (Panel B) categorization for two separate random probes in the test set. The salience is sensitive to the particular context of the example. In comparison to Figure 4, it can be determined that, although there is overlap with respect to the most salient features between the aggregate saliences (Figure 4) and the individual saliences (Figure 5), the magnitude and ranking of the saliences is context dependent, providing a second layer of diagnostics for classification and misclassification.

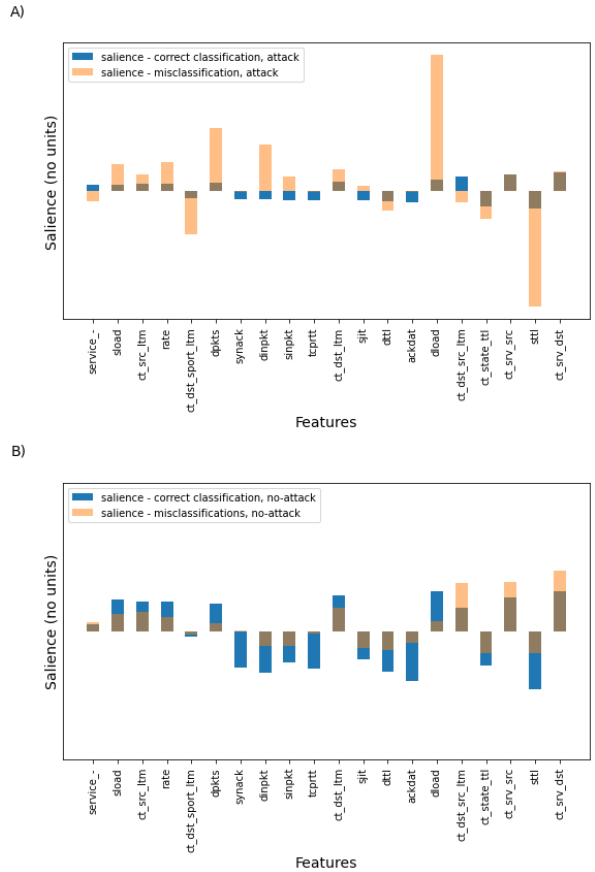

Figure 4: Aggregate Cognitive Salience for the top 20 most influential features in an intrusion classification. Panel A) compares the salience of the top 20 features for correct (blue) and incorrect (orange) classification of a positive intrusion class ('attack'). Panel B) compares the salience, for the same set of features, for classification of non-intrusion traffic ('no-attack'). Note that salience values are relative degrees of influence and have no interpretable units (hence we do not include units on the y-axis).

Cognitive Salience adds a facet of interpretability to the cognitive model, allowing the modeler to introspect upon individual decisions. The information gleaned from this method allows the modeler to understand the relative influence that features have on

individual decisions in a given context. In the future, we plan to investigate how salience can be used to improve classification through the selection of ideal training instances (e.g., Wu & Chang, 2003), or for adapting the model to detect specific classes of intrusions. Existing IDSs could benefit from using this approach by computing the derivative of the classification algorithm to evaluate the relative importance of features for individual classes and types of decisions. The approach not only allows for an overall evaluation of feature importance for dimensionality reduction but also a layer of explainability specific to decision contexts.

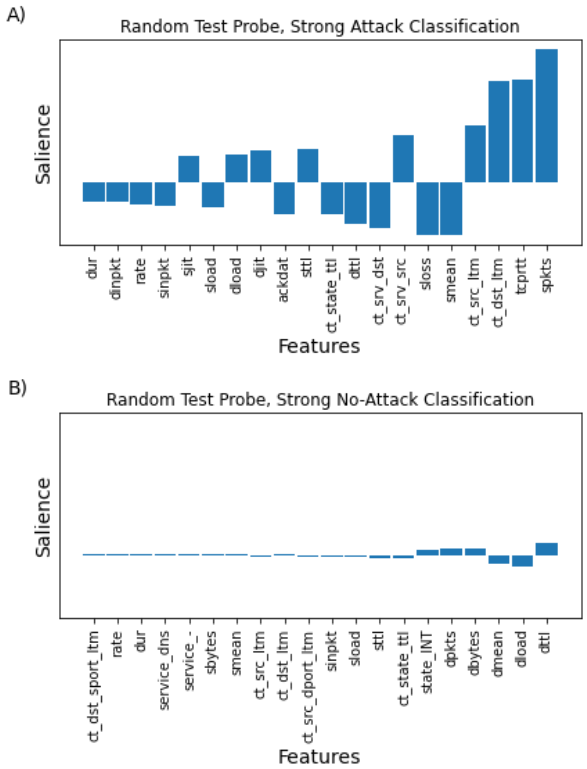

Figure 5. Cognitive Salience for the top 20 most influential features for random probes in 'Attack' (Panel A) and 'No-Attack' (Panel B) classifications. Each panel shows salience from a single probe.

4. Discussion and Conclusion

In this paper, we described a cognitive model capable of achieving 70% binary classification accuracy on the UNSW-NB15 intrusion detection dataset using only one instance per attack category and 9 instances of normal traffic for training. The model's performance linearly increases with additional training instances. The model also does not require one-hot encoding or normalization as the partial matching functionality is agnostic to the input data type. While this method does not outperform machine-learning techniques, it does achieve a high level of performance drawing from the few-shot learning properties of human cognition to scaffold performance. We also show a cognitive salience technique which can be used to introspect over individual decisions to determine which features were most influential in the model's decision-making.

While the present paper does not discuss the relative scalability of the cognitive model's algorithms, they generally increase linearly with the number of instances in memory. Prior efforts with cognitive models (e.g., Sanner et al., 2000, Lebiere et al., 2013) have used techniques to find the most salient instances and use them to limit the number of instances in memory, maintaining the model's adaptivity and overall performance while being able to operate in real-time.

4.1. Benefits of Cognitive Models

Cognitive models have been applied for both understanding and augmenting human analyst performance and as autonomous agents. They have leveraged human-inspired heuristic reasoning to process large amounts of information, well beyond human capacity and without the limitations of human cognitive biases (Thomson, et al., 2015). ACT-R accurately models human cognition in a variety of decision-making (Erev, et al., 2010; Lebiere et al., 2007) and general intelligence tasks (Lebiere et al., 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 knowledge is sparse, limited, or dissimilar to the current context. To scale to complex tasks involving substantial human expertise (Sanner, et al., 2000), models can abstract from the high-fidelity aspects of the task that cannot be constrained by data.

Furthermore, the structure of ACT-R's declarative memory allows for the coding of categorical, ordinal, and numerical data. This has a secondary benefit of maintaining some of the implicit structure of the packet within the instances in the model's memory and avoids the preprocessing requirement for many ML techniques that force inputs to numerical values using one-hot encoding (Potdar, Pardawala, & Pai, 2017) as well as balancing feature importance through normalization.

4.2. Further Model Development

We argue that cognitive 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 to compensate or exploit them. One advantage of cognitive techniques is their degree of flexibility. Specifically, declarative knowledge can be accessed and elaborated in different

ways and for different purposes. For instance, the same blending mechanism used to detect an intrusion and detect a categorization could be used to generate prototypes for each category. Those prototypes could be used to generate rule sets for intrusion detection systems. Another possibility is to use an inference engine that can elaborate instances by reasoning over a cybersecurity domain ontology (e.g., Oltramari et al. 2014a, b). That combination of cognitive mechanisms and ontological reasoning has proven effective in other domains such as robotics.

As we learned in the model of malware detection, simple categorization may not be the best way to achieve detection. That model leveraged ambiguity in assigning a specific family label by preserving a probability distribution over malware families and using that pattern in inferring malware intent, in many ways a more useful conclusion. One approach would be to recognize that intrusion categories may not form a single tight cluster but rather distinct sub-categories. In the domain of visual object recognition, we have developed a methodology for performing unsupervised categorization based on metacognitive signals of similarity (Vinokurov et al., 2012). When an instance is judged sufficiently dissimilar to other instances of that category, a new (sub)category is generated dynamically.

A variant to this approach would be to create a hierarchical categorization by breaking up an intrusion category when it leads to a categorization error for an instance situated between multiple instances of that category. 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), or utilize cognitive salience to prune the space more efficiently (Somers et al., 2019). It would be, in theory, possible to leverage the cognitive model to determine ideal training instances, which would improve model performance and in theory could be inverted to determine ideal data poisoning instances, allowing the model to act as a virtual red team member.

4.3. Integration with Human Analysts

Future research will investigate several techniques to present alerts to analysts with automated analyses tailored to the analyst. Some preliminary work seeking to integrate automation during initial triage of alerts has shown that participants interact differently with algorithms based on the degree of automation (i.e., levels of automation; Cassenti, Roy, Hawkins, & Thomson, 2022) with participants more likely to override an automation's recommendation when still needing to validate each decision (*human-in-the-loop*) when compared to being a position to override decisions (*human-on-the-loop*). We argue that cognitive models would be able to understand both the underlying risk tolerance of the analyst and be able to make timely interjections as required to optimize analyst workflow and workload. This could be abstracted as a decision-aid which mirrors the analysts' decisions and detects when biased decisions are made and presents the analyst with a salience-based explanation on which features are most likely to be more heavily weighted for an expected classification (e.g., suspected misclassification of an attack as benign traffic). This idea could be instantiated as a dashboard widget or visualization leveraging cognitive salience to introspect over the cognitive model's memory. More research needs to be completed in order to assess the viability of the model as a tool.

5. Acknowledgements

This research was sponsored by the Army Research Office and accomplished under MURI Grant Number W911NF-17-1-0370 and C5ISR agreement USMA23011. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

6. References

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought.* Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review* 111 (4), 1036-1060.

Cassenti, D., Roy, A., Hawkins, T., & Thomson, R. (2022) The Effect of Varying Levels of Automation During Initial Triage of Intrusion Detection. In *Artificial Intelligence and Social Computing*, 28, 59-66.

Corbett, A. T. & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling & User-Adapted Interaction*, 253-278.

Cranford, E. A., Gonzalez, C., Aggarwal, P., Cooney, S., Tambe, M., & Lebiere, C. (2020a). Toward personalized deceptive signaling for cyber defense using cognitive models. *TopiCS, 12*, 992–1011.

Cranford, E. A., Gonzalez, C., Aggarwal, P., Cooney, S., Tambe, M., & Lebiere, C. (2020b). Adaptive cyber deception: Cognitively-informed signaling for cyber defense. In *Proceedings of 53rd Hawaii International Conference on System Sciences* (1885-1894). Maui, HI.

Cranford, E. A., Gonzalez, C., Aggarwal, P., Tambe, M., Cooney, S., & Lebiere, C. (2021). Towards a cognitive theory of cyber deception. *Cognitive Science, 45*, 1–28.

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.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance based learning in dynamic decision making. *Cognitive Science, 27*(4), 591–635.

Hamed, T., Ernst, J. B., & Kremer, S. C. (2018). A survey and taxonomy of classifiers of intrusion detection systems. *In Computer and Network Security Essentials* (pp. 21-39).

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*.

Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., & Atkinson, R. (2017). Shallow and deep networks intrusion detection system: A taxonomy and survey. *arXiv preprint*. arXiv:1701.02145.

Lebiere, C. (1999). A blending process for aggregate retrievals. *Proceedings of the 6th ACT-R Workshop*.

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.

Lebiere, C., Gonzalez, C., & Martin, M. (2007) Instance-Based Decision-Making Model of Repeated Binary Choice. In *Proceedings of the International Conference on Cognitive Modeling*.

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.

Lebiere, C., Pirolli, P., Thomson, R., Paik, J., Rutledge-Taylor, M., Staszewski, J., & Anderson, J. (2013). A Functional Model of Sensemaking in a Neurocognitive Architecture. *Computnl Intelligence and Neuroscience*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.

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.

Mitsopoulos, K., Somers, S., Schooler, J., Lebiere, C., Pirolli, P., & Thomson, R. (2021). Toward a psychology of deep reinforcement learning agents using a cognitive architecture. *TopiCS, 14*(4), 756-779.

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.

Muna, A. H., Moustafa, N., & Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on deep learning models. *Journal of Information Security & Applications, 41*, 1-11.

Nunes, E., Buto, C., Shakarian, P., Lebiere, C., Bennati, S., & Thomson, R. (2020). Cognitively-inspired inference for malware task identification. *In Open Source Intelligence and Cyber Crime* (pp. 165-194).

Nunes, E., Buto, C., Shakarian, P., Lebiere, C., Bennati, S., Thomson, R., & Jaenisch, H. (2015). Malware task identification: A data driven approach. *In 2015 IEEE/ACM ASONAM* (pp. 978-985).

Oltramari, A., Cranor, L.F., Walls, R.J., & Mcdaniel, P. (2014a). Building an Ontology of Cyber Security. *Semantic Technologies for Intelligence, Defense, and Security*.

Oltramari, A., Vinokurov, Y., Lebiere, C., Oh, J., & Stentz, A. (2014b). Ontology-based Cognitive System for Contextual Reasoning in Robot Architectures. Presented at the AAAI Spring Symposium on Knowledge Representation and Reasoning in Robotics. *AAAI Spring Symposium Technical Report SS-14-04*. Menlo Park, CA: AAAI Press.

Potdar, K., Pardawala, T. S., & Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications, 175*(4), 7-9.

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

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science, 10*(3).

Sanner, S., Anderson, J., Lebiere, C., & Lovett, M. (2000). Achieving efficient and cognitively plausible learning in backgammon. *Proceedings of ICML, 823-830.*

Somers, S., Mitsopoulos, C., Lebiere, C., & Thomson, R. (2019). CogXAI: Cognitive-level salience for explainable artificial intelligence. In *Proceedings of the International Conference on Cognitive Modeling*.

Salo, F., Nassif, A., & Essex, A. (2019). Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Computer Networks, 148*, 164-175.

Thomson, R., Lebiere, C., Anderson, J. R., & Staszewski, J. (2015). A general instance-based learning framework for studying intuitive decision-making in a cognitive architecture. *Journal of Applied Research in Memory and Cognition, 4*(3), 180-190.

Vinokurov, Y., Lebiere, C., Wyatte, D., Herd, S., & O'Reilly, R. (2012). Unsupervised Learning in Hybrid Cognitive Architectures. In *AAAI-12 Workshop on Neural-Symbolic Learning and Reasoning*.

Wallach, D., & Lebiere, C. (2003). Implicit and explicit learning in a unified architecture of cognition. *Attention and implicit learning*, 215-250.

Wojtas, M., & Chen, K. (2020). Feature importance ranking for deep learning. *Advances in Neural Information Processing Systems, 33*, 5105-5114.

Wu, G., & Chang, E. Y. (2003, August). Class-boundary alignment for imbalanced dataset learning. In *ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC* (pp. 49-56).

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* (TIST), 10(2), 1-19.