On Interactive Machine Learning and the Potential of Cognitive Feedback

Chris J. Michael, Dina Acklin, and Jaelle Scheuerman
U.S. Naval Research Laboratory
1005 Balch Blvd, Code 7343
Stennis Space Center, Mississippi 39529. U.S.A.
chris.michael@nrlssc.navy.mil, dina.acklin@nrlssc.navy.mil, jaelle.scheuerman@nrlssc.navy.mil

Abstract

In order to increase productivity, capability, and data exploitation, numerous defense applications are experiencing an integration of state-of-the-art machine learning and AI into their architectures. Particular to defense applications, having a human analyst in the loop is of high interest due to quality control, accountability, and complex subject matter expertise not readily automated or replicated by AI. However, many applications are suffering from a very slow transition. This may be in large part due to lack of trust, usability, and productivity, especially when adapting to unforeseen classes and changes in mission context. Interactive machine learning is a newly emerging field in which machine learning implementations are trained, optimized, evaluated, and exploited through an intuitive human-computer interface. In this paper, we introduce interactive machine learning and explain its advantages and limitations within the context of defense applications. Furthermore, we address several of the shortcomings of interactive machine learning by discussing how cognitive feedback may inform features, data, and results in the state of the art. We define the three techniques by which cognitive feedback may be employed: self reporting, implicit cognitive feedback, and modeled cognitive feedback. The advantages and disadvantages of each technique are discussed.

The Emergence of Interactive Machine Learning

The vast majority of modern-day research in machine learning presents algorithms and implementations that do not consider human interaction. For example, the flourishing field of deep learning research is evaluated mainly by classification accuracy over large curated datasets and generative models. This approach, referred to as Automatic Machine Learning (AML) or sometimes conventional machine learning, forgoes the integration of dynamic human feedback into the system. Though undoubtedly useful for commercial big-data problems, there are many scenarios – especially in defense – where applying AML falls short in practice. For instance, applications at the tactical edge may suffer from smaller quantities of labeled examples for training. Moreover, classifiers may struggle to adapt to changes in data context quickly enough to be considered viable by an analyst, particularly in scenarios where the mission demands quick turn-around time. Many of these issues may be mitigated by emerging implementations of the *interactive machine learning* (IML) paradigm, which capitalizes on human input in order to improve machine learning implementations (Fails and Olsen Jr 2003). Unlike approaches that leverage AML, IML implementations allow for classifiers to very quickly train and apply newly discovered information with the help of a human subject-matter expert, which we refer to in this article as the analyst.

In general, IML may be described as a machine learning implementation where one or more analysts iteratively improve a model for automation by manipulating an interface that is tightly coupled to the desired task at hand. There are four main components to any IML implementation. The first component is the data associated with the task. Examples of such data include remotely sensed imagery, textual information such as reports, and spatiotemporal tracks of moving objects. The second component, referred to in this study as the *machine*, is the mathematical model that tries to estimate or automate the desired task. Ostensibly, this can be seen as a black-box, but we will discuss the properties of a successful IML classifier later in the article. The third component of IML is the Human-Computer Interface (HCI). The HCI may be as conventional as software receiving input through a keyboard and mouse, which is what we assume in this article, or as specialized as vehicle controls, immersive environments, and brain interfaces. The application is designed to allow immediate and intuitive presentation of the machine's classification on a manageable set of data. This data is then either confirmed or manipulated to be correct by the analyst, who is the last but most important component of an IML system. In this article, we discuss IML within the context of improving productivity and decision making for an analyst with a very specific task that requires subject-matter expertise. Though, as exemplified above, IML may be deployed in a wide variety of ways, we feel that deployment in this context

 In: Proceedings of AAAI Symposium on the 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applications: Potentials, Theories, Practices, Tools, and Risks, November 11-12, 2020, Virtual, published at http://ceur-ws.org

has the greatest potential for impact in defense applications. There are several studies that provide excellent perspectives of the current state of the art in IML outside of this scope (Dudley and Kristensson 2018; Wu, Weld, and Heer 2019; Robert et al. 2016).

A common architecture for IML implementations is shown in Figure 1. The data on which the analyst must perform a task may either be completely available in a database or sequentially available as a stream. *Active learning* may be used to pull the most effective data points from this database for labeling, as will be discussed in the next section. A machine for predicting the data is then used to present guesses for the task at hand to the analyst. The analyst must verify each of these guesses and correct any mistakes via the HCI. Once the verification step is completed for the current iteration, the machine will immediately learn from the corrections and/or confirmations. The process will then repeat by the machine gathering data examples and presenting guesses to the user once again. When the time comes for the analyst to leave their duty station, the machine model may optimize on the data that has been labeled in order to maximize its accuracy. This way, the most effective machine will be available once the analyst returns to duty. It is important to note that the machine may be deployed as a centralized general-purpose classifier that combines the work done by multiple analysts, or it may be deployed locally to be custimized towards the individual analyst.

The focus of this article is to introduce IML within the context of analyst-driven applications relevant to defense while highlighting research gaps, the most important of which involves incorporation of cognitive feedback. We choose not to discuss manual *model interactions* such as feature selection (Raghavan, Madani, and Jones 2006) or model selection (Talbot et al. 2009), which are processes whereby analysts directly optimize machine models. Rather, we choose to present implementations that can be used effectively by an analyst who is a subject matter expert for the task at hand and not knowledgeable in machine learning or statistical theory. Defense analysts hold invaluable subject-matter expertise for the mission, and it is unreasonable to assume that they must learn or worry about data-scientific concepts. Because intuitive HCIs may be designed to be congruent to their task, IML has great potential to leverage the power of modern-day ML while not burdening the analyst with parameter tuning, data curation, or any of the other burdens implicit to AML.

The next section will describe three examples of IML implementations that highlight the current state of the art. The section that follows will iterate through several advantages, shortcomings, and gaps in the state of the art. In the penultimate section, we specify the ways in which cognitive feedback may be used to address the shortcomings and gaps of IML with respect to defense applications. Finally, we conclude with commentary on prospects for future research.

Interactive Machine Learning in Action

In order to frame a more detailed discussion of IML, we now describe several IML implementations that have been presented in peer-reviewed literature. We specifically choose three application areas that are analyst-driven: region digitization, textual translation, and video annotation. These examples demonstrate the potential for IML to improve both the machine performance and the user experience with autonomy.

Geographic region digitization is a highly demanded yet arduous task whereby regions such as bodies of water and other land cover are digitized from remotely sensed images, usually within a Geographic Information System (Hossain and Chen 2019). Once digitized, regions may be represented in mapping products for geospatial situational awareness, climate-level studies, land surveys, and many other applications. Although numerous AML approaches to region digitization have been presented in the literature, they are not widely adopted in practice. This is most likely due to the all-or-nothing yield of AML approaches: If the machine incorrectly digitizes a region, it may be more burdensome for an analyst to correct than to start from scratch. Therefore, an analyst may prefer to digitize manually to circumvent frustration and presumably lower their workload. In order to address these shortcomings, an IML implementation for region digitization, named the Geospatial Region Application Interface Toolkit (GRAIT), is presented as a human-machine team application (Michael et al. 2019). The authors address the all-or-nothing approach to region digitization with an IML implementation where a region is digitized iteratively. In each iteration, the machine guesses the placement of a certain number of vertices of the contour and presents them to the analyst for verification. For each vertex presented, the analyst may either correct its placement by clicking and dragging it to an appropriate location or simply confirm its correct placement by not interacting with it. The analyst indicates via button press when all the vertices of the current iteration are corrected or confirmed. The machine will then train on the finalized vertex locations, and the process will continue until the region is completely digitized. In order to prevent inducing too high of a cognitive load on the analyst, an uncertainty model is used to estimate the probability of incorrect vertex placement and limit each iteration to around 2 incorrectly placed vertices. Results show that with no prior training data, the IML implementation accurately places 84% of vertices correctly in 4 separate image sets of 4 images each.

Another area where IML approaches show promise is that of textual language translation, commonly referred to as *machine translation*. While bodies of work in this field attempt to replace human translators with machine models, many of which are AML implementations (Koehn 2009), the current state of the art is far from perfect. As with region digitization, fully-automatic approaches may hinder rather than help the performance of a translator at times when too many mistranslated words may induce excessive cognitive load. Because of this, many approaches to machine translation are realized through a human-machine team. An IML approach to machine translation aims to remedy these issues by implementing iterative learning and modeling the informativeness of each machine translation at a fine-grained level (González-Rubio and Casacuberta 2014). In this approach, an initial guess of a sentence translation is given to the user

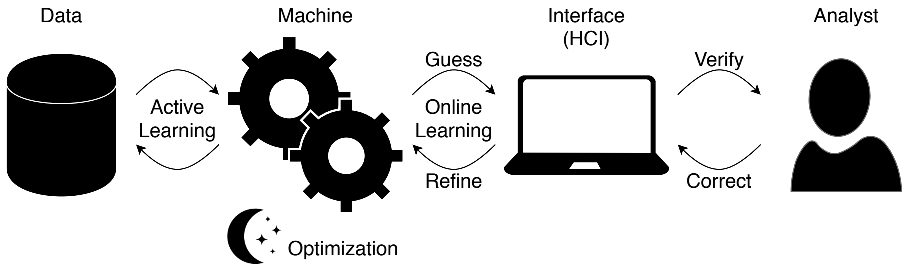

Figure 1: A common architecture for interactive machine learning implementations.

based on a metric of informativeness. The user will then make corrections to the guess by changing the first incorrect letter of the translation. The machine in turn suggests a new translation under this assumption. This process continues, with the machine immediately training on corrected data for future translations. Results show that employing this IML-based method produces twice the translation quality, a metric specific to machine translation, per user interaction over AML approaches.

Lastly, IML implementations have emerged for the difficult task of video annotations, where the amount of data generated per day has far surpassed the ability of analysts to inspect. When successful, annotated video allows for critical advantages such as the ability to search for events, quantify behavioral analytics, and study natural phenomena. Though many AML approaches to video analytics exist, they are typically tied to certain features of interest within some constrained context (Ananthanarayanan et al. 2017). In cases where context may change and the features of interest are unknown, AML implementations for automatic video annotation may be rendered incorrect or infeasible. This is especially true in cases where context has changed or features of interest are unknown beforehand. An IML implementation of video annotation named Janelia Automatic Animal Behavior Annotator (JAABA) demonstrates a semi-automatic approach to assess animal behavior (Kabra et al. 2013). JAABA allows for a user to annotate a video frame with an arbitrary label, for instance *jump*. Then, using trajectory information extracted from the video, the machine trains on the given label and presents classification results both at the level of the current video and a database of numerous animal videos. The machine also provides confidence levels for each classification to guide further labeling by the user. This process is repeated iteratively until an ideal classifier is attained. JAABA was used to create the first ML-driven behavior classifier over a diverse set of animals.

With these three examples in mind, a more detailed explanation of the advantages, limitations, and gaps of IML will follow.

Advantages, Shortcomings, and Gaps

Advantages

The advantages of IML approaches directly address many of the shortcomings that defense applications exhibit when utilizing ML. Numerous defense applications suffer from a shortage of labeled training examples due to a lack of crowd sourcing and the ever changing state of platform technologies among other reasons. As such, deep models relying on large amounts of labeled examples cannot be adequately trained. IML addresses the shortage of training data by providing an interface that allows for incorrect classifications to be immediately corrected and integrated into the machine model. In fact, several IML implementations may work well with no prior labeled data, which is usually referred to as the *cold start problem* (Lika, Kolomvatsos, and Hadjiefthymiades 2014). Additionally, the HCI allows for correction through an intuitive interface that potentially reduces the burden of data labeling. This allows an analyst to leverage their current subject-matter expertise – that of the application and data context – and circumvents the need to play the role of data scientist.

Defense problems must be very adaptable to context changes from one region of interest to the next. In order to accommodate this, any autonomy must immediately adapt to such changes at the pace of the analyst. Therefore, IML implementations typically apply active learning and online learning techniques in order to improve effectiveness. Active learning research entails the study of uncertainty or similarity metrics in order to develop a mathematical understanding of the likelihood that a machine will classify future data points correctly (Quionero-Candela et al. 2009). The field of online machine learning involves models that may train in stride to adapt to new situations quickly while optimizing exploration vs. exploitation (Bottou 1998).

Problems related to defense must sometimes be deployed at the tactical edge. In such situations, computational resources and downtime may be scarce. IML directly addresses this problem, since most IML implementations are meant to be deployed on desktop computers. In all three examples of IML presented in the previous section, online and active learning strategies are employed to iteratively build high-performance classifiers. Active learning is also used to gage the load of examples presented to the user, both by correlating uncertainty to the probability of an incorrect classification and by providing a priority for the analyst to manage their own work flow. Both GRAIT and JAABA support cold-start cases.

Shortcomings

Perhaps the most obvious shortcoming if IML is that the HCI and machine implementation must be tightly coupled to a specific application. This entails much more effort in the development of applications, since they must be built and studied uniquely towards an explicit work flow. This differs greatly from AML approaches, where for the most part implementations are general-purpose and specificity is implied through parameterization and classes for labeling. Studies define a general-purpose methodology for HCI, but this research is young and remains mostly theoretical in nature (Meza Martínez and Maedche 2019).

Deep models of machine learning exhibit very impressive results relating to throughput of data and classification times. IML implementations currently lag behind in these results. This is in part due to the nature of online machine learning; namely, the need to have tight classification and training cycles. However, research is trending more towards online and active learning problems, and IML-inspired classifiers with competent performance are emerging (Langford, Li, and Strehl 2007; Lu, Shi, and Jia 2013).

A further issue with IML is that overfitting may occur more frequently since data is generally labeled iteratively. Overfitting occurs when prior training data causes the model to correlate too tightly to features that do not justify the desired outcome. For example, one of the geographic sites in the GRAIT study is Johnson Lake, WA. The first three images show the shoreline in roughly the same location. The fourth image shows the lake with a receded shoreline. Though the shoreline may be spotted by an analyst clearly in the fourth image, the classifier overfit to spaital features and thus incorrectly identified the shoreline. This also caused the uncertainty calculations for the image to be undershot. AML approaches to overfitting typically require optimizing machine parameters or adding diversity to datasets, both of which typically require large amounts of computation and thus long turnaround times not conducive to successful IML implementations. Therefore, reinforcement meta-learning, whereby active learning implementations are informed by corrections via specialized ML implementations, may be employed to adapt quickly to situations where overfitting is inevitable (Bachman, Sordoni, and Trischler 2017).

The Cognitive Gap

Although frequently mentioned as a future direction of study, perhaps the largest identified gap in IML research is the lack of formalization and quantification of cognitive implications from the analyst. For instance, the IML machine translation study (González-Rubio and Casacuberta 2014) mentions specifically that the applied technique lessens the cognitive load of the translator by utilizing cost-sensitive metrics such as informativeness. However, the study does not perform any human-factors research to back support this claim, though it is mentioned as future work. As another example, the study presenting GRAIT uses mathematically modeled uncertainty calculations to meter the workload at each iteration. Though it is shown statistically that these uncertainty calculations correlate to the probability a vertex is placed correctly, results focus more on vertex placement accuracy and do not consider multiple load levels (e.g. the number of expected incorrect vertices is set to two for the entire study). Human factors research is also slated as future work. Both of these studies appreciate that there must be thresholds of cognitive load taken into account by the IML system for a successful implementation, but it is apparent that human-factors research is inevitable.

The Implications of Cognitive Feedback

Due to its interactive nature, IML most certainly is a human-in-the-loop endeavor. Several studies have highlighted difficulties that may arise from trust, safety, and quality (Dudley and Kristensson 2018; Groce et al. 2013; Gillies et al. 2016; Turchetta, Berkenkamp, and Krause 2019). This section is devoted to discussing the potential of researching and integrating models of human cognition as feedback for IML, which is not often mentioned in the state of the art. We also make the argument that cognitive feedback directly addresses the shortcomings of IML. The topic of cognitive feedback is especially useful for defense-related problems, where trust, safety, and quality of ML implementations is a prerequisite for adoption. Without analyst-driven cognitive feedback, an IML system can very quickly fall flat, which is illustrated in the following region digitization example.

Consider the analyst using GRAIT to digitize the fourth image of Johnson Lake as explained in the previous section. Recall that the machine is overfit, and thus its model for uncertainty is undershot. Because of this, the machine places 10 vertices, 8 of which are incorrectly placed. If the analyst continues, they will spend more time correcting the misplaced vertices than manually digitizing the lake without the help of the machine.

This example is simple, but it highlights one of the detrimental problems of IML implementations: Overfitting is inevitable, and it can induce, rather than relieve, cognitive load. As mentioned previously, reinforcement learning may be used to augment the uncertainty or similarity model based on the number of corrections the user has to make in any iteration. However, convergence of such a technique would involve the user making excessive corrections in order to inform the model in this example. Unlike AML, the uncertainty and workload involved with IML data must be somehow informed by the analyst.

Figure 2 shows several situations exemplifying various levels of cognitive load when an analyst uses GRAIT to annotate some region of interest. In the first example, the machine is very accurate but offers too few vertices for the analyst to verify. In this situation, the analyst is impeded by an overshot cognitive load. The analyst must work at the slow pace of the IML implementation, which not only reduces their productivity but may also reduce their attention and engagement. The second example shows the ideal situation where GRAIT correctly manages the cognitive load of the analyst. The analyst is expected to be engaged and productive. The last situation shows an example of the IML implementation undershooting the cognitive load. This causes the analyst to become overwhelmed and possibly confused, slowing their productivity and causing frustration.

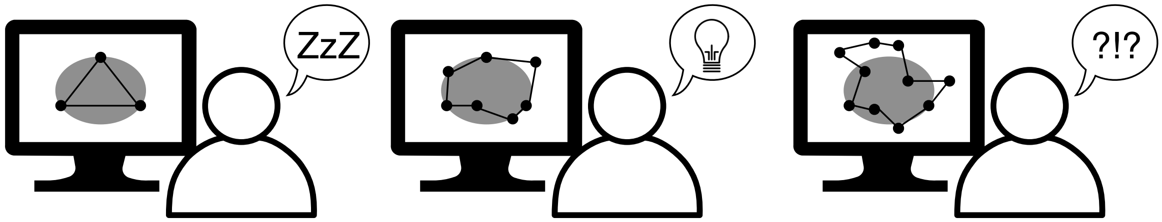

Figure 2: Various degrees of engagement with IML region digitization. In the first image, the machine has overshot cognitive load and thus the analyst's productivity is hampered. In the second image, the analyst is engaged in the task and the machine is helping their productivity. In the last image, the machine has undershot the cognitive load and thus the analyst is overwhelmed and will most likely abandon the IML implementation for the task.

Incorporation of cognitive load is necessary to avoid the pitfall of bad cognitive load estimation based on analysis of data alone. For instance, consider an augmentation to the third GRAIT example in the figure by providing the user with a survey at each iteration. The survey will occur before correction and simply ask, "Is this workload too little, too much, or fine?" In this particular situation, the analyst will inform the machine that the workload is too much to handle, and the machine may modify its uncertainty model accordingly (e.g. by adjusting weighting or performing best-fit optimization to prior iterations). This very simple solution illustrates how cognitive feedback may enable better IML for many applications, but this concept may be taken further. In order to promote discussion and research of the possibilities and implications of this concept, we now present a taxonomy for cognitive feedback to inform IML.

Self-reported cognitive feedback is gathered by surveys eliciting cognitive feedback from the user. An example of such a survey is the standard NASA-TLX, which allows a user to report on the general experienced workload of a particular task (Hart and Staveland 1988). This could be gathered offline during human factors evaluation or online through an interface for self reporting within the HCI. The main advantage of online self reporting cognitive load is the simplicity to collect feedback within the HCI. Implementation of simple interventions, such as providing buttons for when a workload is too heavy or too light, are trivial. However, this approach may be imprecise in complex user environments because sub-components of a task may differentially contribute to workload. In these situations, interventions may be too simplistic or induce load on an analyst.

Until now, we've discussed the implications of self reporting on cognitive load, but this technique may provide insight into more than just the analyst's ideal workload. The field of explainable artificial intelligence involves expressing the machine's decision making to a human user (Gunning and Aha 2019). If a model for explainability is feasible, then the user may communicate cognitive information relating to features as feedback to the model (Teso and Kersting 2019). Relating back to the example above, the machine may explain its decisions by stating "I believe that historic position of the shoreline is very important." The user may then augment the belief by stating "The historic position is not as important as color," and the machine may then optimize its classifier and uncertainty calculation based on this statement.

As opposed to surveying a user, *implicit cognitive feedback* may be collected in real time while analysts interact with the HCI during closed experimentation. Implicit cognitive feedback involves collecting physiological data in order to infer cognitive states in a manner that is continuous, objective, and occurs in real time. For example, because pupillary responses are reflective of nervous activity, pupil dilation may act as a proxy for measuring task-induced cognitive processes. As such, increases in pupil diameter may be indicative of high cognitive load, attentional processing, and decision making (Hess and Polt 1964; Kahneman 1973; Hahnemann and Beatty 1967) whereas decreases may reflect fatigue (Lowenstein, Feinberg, and Loewenfeld 1963). This data may then be correlated with self-reporting to define various states of cognitive load. Examples of such biofeedback include readings of skin conductance, heart rate, pupilometry, and electroencephalogram (EEG). Often, multiple physiological measures will be assessed to determine workload and inform adaptive algorithms, in essence creating user models that dynamically adjust to support user needs. For example, such physiological elements were examined to monitor the workload of operators while performing UAV piloting tasks of different levels (Wilson and Russell 2007). The physiological signals were used as features to train a neural network to classify workload. Another approach of implicit cognitive feedback is to incorporate cognitive cues as features in the machine learning algorithm (Rosenfeld et al. 2012). For example, in a recent choice competition, researchers incorporated cognitive features derived from behavior into a random forest algorithm. They found that this approach significantly outperformed other ML approaches that did not incorporate cognitive features (Plonsky et al. 2017). A recent study has explored how collecting and applying cognitive cues as features improves reinforcement learning algorithms for playing video games (Zhang et al. 2019). In summary, implicit cognitive feedback has the potential to improve IML implementations by gathering data in closed experimentation to inform cognitive load, uncertainty/similarity measurements, and inform the machine with features of interest related to a specific task.

Implicit cognitive feedback may provide invaluable in-

Table 1: Taxonomy of Cognitive Feedback for Interactive Machine Learning

Term	Definition	Examples
Self Reporting	Gathered by surveying the analyst.	Online: Buttons in HCI. Offline: Human-factors surveys.
Implicit	Collection and evaluation of biofeedback via closed experimentation.	Cognitive load of correction via HCI. Load as a function of correction count. Use of cognitive cues as ML features.
Modeled	Utilization of a cognitive model in the loop.	Feedback model of user interaction with HCI.

sight to IML implementations, but the disadvantage lies in the fact that closed experimentation is often necessary to collect biofeedback, control levels of tasking, and survey users of the HCI with respect to a particular application. Additionally, the cognitive state of the user may be more dynamic for some applications than others. In these situations, *modeled cognitive feedback* may provide cognitive feedback based on models of user interaction with the HCI. For example, simulating human behavior using a computational cognitive model is another potential method to provide feedback to an IML system. Models of cognition and decision making have been used to simulate human interactions with interfaces in military contexts (Blasch et al. 2011). Cognitive architectures represent a modeling paradigm that computationally defines the relationship between underlying biological and cognitive mechanisms to emerging behavior. Architectures, such as ACT-R (Anderson et al. 2004) and SOAR (Laird, Newell, and Rosenbloom 1987), have long been a part of HCI research to simulate users interacting with an interface. For example, ACT-R models are used for usability testing of menus (Byrne 2001), modeling how users detect phishing websites (Williams and Li 2017), and detecting situations with high cognitive load when using a smartphone (Wirzberger and Russwinkel 2015). Cognitive architectures have been used with physiological data, such as eye tracking information and fMRI, to map observed behavior the underlying mental states and brain regions (Tamborello and Byrne 2007; Borst and Anderson 2015). Cognitive models, combined with self-reported data from surveys and physiological data, can provide a starting point for IML systems to optimize their suggestions for the overall performance of a human-machine team.

These three different categories of cognitive feedback – self reporting, implicit cognitive feedback, and modeled cognitive feedback – delineate the possible ways in which IML implementations may be centered around the analyst. The categories are summarized in Table 1.

Once cognitive feedback has been integrated into IML, more conventional results such as classification accuracy and overall corrections may be used to evaluate approaches against their non-cognitive baseline. However, these results may lack true insight into the purpose of the human-machine team. Measuring the cognitive load on human subjects with more objective metrics of productivity would provide more insight into the effectiveness of IML implementations (Alves et al. 2016). Additionally, it is the analyst themselves who must also evaluate the effectiveness of an IML implementation, though this may take high levels of time and effort (Groce et al. 2013; Gillies et al. 2016).

A Future Driven by Cognitive Feedback

We have presented a summary of interactive machine learning along with several examples informing the state of the art. After discussing the advantages of IML, the major shortcomings and gaps were delineated. Finally, the implications of cognitive feedback for IML implementations were discussed to address the gaps. Though it may seem trivial to study cognitive feedback as it relates to data science for human-in-the-loop applications, there is a general lack of such studies in the literature, especially for defense applications. We hope this article will encourage research and development in more IML for defense applications and more research in how cognitive feedback may inform IML implementations.

References

Alves, F.; Szpak, K. S.; Gonçalves, J. L.; Sekino, K.; Aquino, M.; e Castro, R. A.; Koglin, A.; de Lima Fonseca, N. B.; and Mesa-Lao, B. 2016. Investigating cognitive effort in post-editing: A relevance-theoretical approach. *Eyetracking and Applied Linguistics* 2:109.

Ananthanarayanan, G.; Bahl, P.; Bodík, P.; Chintalapudi, K.; Philipose, M.; Ravindranath, L.; and Sinha, S. 2017. Real-time video analytics: The killer app for edge computing. *computer* 50(10):58–67.

Anderson, J. R.; Bothell, D.; Byrne, M. D.; Douglass, S.; Lebiere, C.; and Qin, Y. 2004. An integrated theory of the mind. *Psychological review* 111(4):1036–1060.

Bachman, P.; Sordoni, A.; and Trischler, A. 2017. Learning algorithms for active learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 301–310. JMLR. org.

Blasch, E. P.; Breton, R.; Valin, P.; and Bosse, E. 2011. User information fusion decision making analysis with the c-ooda model. In *14th International Conference on Information Fusion*, 1–8. IEEE.

Borst, J. P., and Anderson, J. R. 2015. Using the ACT-R Cognitive Architecture in Combination With fMRI Data. In Forstmann, B. U., and Wagenmakers, E.-J., eds., *An Introduction to Model-Based Cognitive Neuroscience*. New York, NY: Springer. 339–352.

Bottou, L. 1998. Online learning and stochastic approximations. *On-line learning in neural networks* 17(9):142.

Byrne, M. D. 2001. ACT-R/PM and menu selection: applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies* 55(1):41–84.

Dudley, J. J., and Kristensson, P. O. 2018. A review of user interface design for interactive machine learning. *ACM Trans. Interact. Intell. Syst.* 8(2):8:1–8:37.

Fails, J. A., and Olsen Jr, D. R. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, 39–45. ACM.

Gillies, M.; Fiebrink, R.; Tanaka, A.; Garcia, J.; Bevilacqua, F.; Heloir, A.; Nunnari, F.; Mackay, W.; Amershi, S.; Lee, B.; et al. 2016. Human-centred machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 3558–3565. ACM.

González-Rubio, J., and Casacuberta, F. 2014. Cost-sensitive active learning for computer-assisted translation. *Pattern Recognition Letters* 37:124 – 134. Partially Supervised Learning for Pattern Recognition.

Groce, A.; Kulesza, T.; Zhang, C.; Shamasunder, S.; Burnett, M.; Wong, W.-K.; Stumpf, S.; Das, S.; Shinsel, A.; Bice, F.; et al. 2013. You are the only possible oracle: Effective test selection for end users of interactive machine learning systems. *IEEE Transactions on Software Engineering* 40(3):307–323.

Gunning, D., and Aha, D. W. 2019. Darpa's explainable artificial intelligence program. *AI Magazine* 40(2):44–58.

Hahnemann, D., and Beatty, J. 1967. Pupillary responses in a pitch-discrimination task. *Perception & Psychophysics* 2(3):101–105.

Hart, S. G., and Staveland, L. E. 1988. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52. Elsevier. 139–183.

Hess, E. H., and Polt, J. M. 1964. Pupil size in relation to mental activity during simple problem-solving. *Science* 143(3611):1190–1192.

Hossain, M. D., and Chen, D. 2019. Segmentation for object-based image analysis (obia): A review of algorithms and challenges from remote sensing perspective. *ISPRS Journal of Photogrammetry and Remote Sensing* 150:115–134.

Kabra, M.; Robie, A. A.; Rivera-Alba, M.; Branson, S.; and Branson, K. 2013. Jaaba: interactive machine learning for automatic annotation of animal behavior. *Nature methods* 10(1):64.

Kahneman, D. 1973. *Attention and effort*, volume 1063. Citeseer.

Koehn, P. 2009. *Statistical machine translation*. Cambridge University Press.

Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. SOAR: An architecture for general intelligence. *Artificial Intelligence* 33(1):1–64.

Langford, J.; Li, L.; and Strehl, A. 2007. Vowpal wabbit online learning project.

Lika, B.; Kolomvatsos, K.; and Hadjiefthymiades, S. 2014. Facing the cold start problem in recommender systems. *Expert Syst. Appl.* 41(4):2065–2073.

Lowenstein, O.; Feinberg, R.; and Loewenfeld, I. E. 1963. Pupillary movements during acute and chronic fatigue: A new test for the objective evaluation of tiredness. *Investigative Ophthalmology & Visual Science* 2(2):138–157.

Lu, C.; Shi, J.; and Jia, J. 2013. Online robust dictionary learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 415–422.

Meza Martínez, Miguel Angel; Nadj, M., and Maedche, A. 2019. Towards an integrative theoretical framework of interactive machine learning systems. In *Proceedings of the 27th European Conference on Information Systems(ECIS)*.

Michael, C. J.; Dennis, S. M.; Maryan, C.; Irving, S.; and Palmsten, M. L. 2019. A general framework for human-machine digitization of geographic regions from remotely sensed imagery. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '19, 259–268. New York, NY, USA: ACM.

Plonsky, O.; Erev, I.; Hazan, T.; and Tennenholtz, M. 2017. Psychological Forest: Predicting Human Behavior. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*.

Quionero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. D. 2009. *Dataset shift in machine learning*. The MIT Press.

Raghavan, H.; Madani, O.; and Jones, R. 2006. Active learning with feedback on features and instances. *Journal of Machine Learning Research* 7(Aug):1655–1686.

Robert, S.; Büttner, S.; Röcker, C.; and Holzinger, A. 2016. Reasoning under uncertainty: Towards collaborative interactive machine learning. In *Machine learning for health informatics*. Springer. 357–376.

Rosenfeld, A.; Zuckerman, I.; Azaria, A.; and Kraus, S. 2012. Combining psychological models with machine learning to better predict people's decisions. *Synthese* 189(1):81–93.

Talbot, J.; Lee, B.; Kapoor, A.; and Tan, D. S. 2009. Ensemblematrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1283–1292. ACM.

Tamborello, F. P., and Byrne, M. D. 2007. Adaptive but non-optimal visual search behavior with highlighted displays. *Cognitive Systems Research* 8(3):182–191.

Teso, S., and Kersting, K. 2019. Explanatory interactive machine learning. In *http://www. aies-conference. com/accepted-papers/*. AAAI.

Turchetta, M.; Berkenkamp, F.; and Krause, A. 2019. Safe exploration for interactive machine learning. In *Advances in Neural Information Processing Systems*, 2887–2897.

Williams, N., and Li, S. 2017. Simulating Human Detection of Phishing Websites: An Investigation into the Applicability of the ACT-R Cognitive Behaviour Architecture Model. In *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*, 1–8. ISSN: null.

Wilson, G. F., and Russell, C. A. 2007. Performance enhancement in an uninhabited air vehicle task using psychophysiologically determined adaptive aiding. *Human factors* 49(6):1005–1018.

Wirzberger, M., and Russwinkel, N. 2015. Modeling Interruption and Resumption in a Smartphone Task: An ACT-R Approach. *Journal of Interactive Media* 14(2):147–154.

Wu, T.; Weld, D. S.; and Heer, J. 2019. Local decision pitfalls in interactive machine learning: An investigation into feature selection in sentiment analysis. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26(4):24.

Zhang, R.; Liu, Z.; Guan, L.; Zhang, L.; Hayhoe, M. M.; and Ballard, D. H. 2019. Atari-head: Atari human eye-tracking and demonstration dataset. *arXiv preprint arXiv:1903.06754*.